
@inproceedings{Zhang:Xu:Kawarabayashi:Jegelka:Boyd-Graber-2019,  
  Author = {Mozhi Zhang and Keyulu Xu and Ken-ichi Kawarabayashi and Stefanie Jegelka and Jordan Boyd-Graber},  
  Title = {Are Girls Neko or Shōjo? Cross-Lingual Alignment of Non-Isomorphic Embeddings with Iterative Normalization},  
  Booktitle = {Association for Computational Linguistics},  
  Year = {2019},  
  Location = {Florence, Italy},  
  Url = {http://umiacs.umd.edu/~jbg/docs/2019_acl_clwe.pdf},  
}

Links:


Downloaded from [http://umiacs.umd.edu/~jbg/docs/2019_acl_clwe.pdf](http://umiacs.umd.edu/~jbg/docs/2019_acl_clwe.pdf)

Contact Jordan Boyd-Graber (jbg@boydgraber.org) for questions about this paper.
Are Girls Neko or Shōjo? Cross-Lingual Alignment of Non-Isomorphic Embeddings with Iterative Normalization

Mozhi Zhang¹ Keyulu Xu² Ken-ichi Kawarabayashi³
Stefanie Jegelka² Jordan Boyd-Graber¹

¹University of Maryland, College Park, Maryland, USA
²Massachusetts Institute of Technology, Cambridge, Massachusetts, USA
³National Institute of Informatics, Tokyo, Japan

{mozhi,jbg}@umiacs.umd.edu {keyulu,stefje}@mit.edu kkeniti@nii.ac.jp

Abstract

Cross-lingual word embeddings (CLWE) underlie many multilingual natural language processing systems, often through orthogonal transformations of pre-trained monolingual embeddings. However, orthogonal mapping only works on language pairs whose embeddings are naturally isomorphic. For non-isomorphic pairs, our method (Iterative Normalization) transforms monolingual embeddings to make orthogonal alignment easier by simultaneously enforcing that (1) individual word vectors are unit length, and (2) each language’s average vector is zero. Iterative Normalization consistently improves word translation accuracy of three CLWE methods, with the largest improvement observed on English-Japanese (from 2% to 44% test accuracy).

1 Orthogonal Cross-Lingual Mappings

Cross-lingual word embedding (CLWE) models map words from multiple languages to a shared vector space, where words with similar meanings are close, regardless of language. CLWE is widely used in multilingual natural language processing (Klementiev et al., 2012; Guo et al., 2015; Zhang et al., 2016). Recent CLWE methods (Ruder et al., 2017; Glavas et al., 2019) independently train two monolingual embeddings on large monolingual corpora and then align them with a linear transformation. Previous work argues that these transformations should be orthogonal (Xing et al., 2015; Smith et al., 2017; Artetxe et al., 2016): for any two words, the dot product of their representations is the same as the dot product with the transformation. This preserves similarities and substructure of the original monolingual word embedding but enriches the embeddings with multilingual connections between languages.

Thus, many state-of-the-art mapping-based CLWE methods impose an orthogonal constraint (Artetxe et al., 2017; Conneau et al., 2018; Alvarez-Melis and Jaakkola, 2018; Artetxe et al., 2018; Ruder et al., 2018; Alvarez-Melis et al., 2019). The success of orthogonal methods relies on the assumption that embedding spaces are isomorphic; i.e., they have the same inner-product structures across languages, but this does not hold for all languages (Søgaard et al., 2018; Fujinuma et al., 2019). For example, English and Japanese fastText vectors (Bojanowski et al., 2017) have different substructures around “girl” (Figure 1). As a result, orthogonal mapping fails on some languages—when Hoshen and Wolf (2018) align fastText embeddings with orthogonal mappings, they report 81% English–Spanish word translation accuracy but only 2% for the more distant English–Japanese.

While recent work challenges the orthogonal assumption (Doval et al., 2018; Joulin et al., 2018; Jawanpuria et al., 2019), we focus on whether simple preprocessing techniques can improve the suitability of orthogonal models. Our iterative method normalizes monolingual embeddings to make their structures more similar (Figure 1), which improves subsequent alignment.

Our method is motivated by two desired properties of monolingual embeddings that support orthogonal alignment:

1. Every word vector has the same length.
2. Each language’s mean has the same length.

Standard preprocessing such as dimension-wise mean centering and length normalization (Artetxe et al., 2016) do not meet the two requirements at the same time. Our analysis leads to Iterative Normalization, an alternating projection algorithm that normalizes any word embedding to provably satisfy both conditions. After normalizing the monolingual embeddings, we then apply mapping-based CLWE algorithms on the transformed embeddings.
Figure 1: The most similar Japanese words for 少女 (shōjo “girl”) and English words for “girl”, measured by cosine similarity on Wikipedia fastText vectors, before (left) and after (right) Iterative Normalization. In the original embedding spaces, “boy” is the nearest neighbor for both languages but with a very different cosine similarity, and “cat” in English is not close to “girl”: both violate the isomorphism assumed by an orthogonal transformation for cross-lingual representations. Iterative Normalization replaces 猫 (neko “cat”) with the more relevant 美少女 (bishōjo “pretty girl”) and brings cosine similarities closer.

We empirically validate our theory by combining Iterative Normalization with three mapping-based CLWE methods. Iterative Normalization improves word translation accuracy on a dictionary induction benchmark across thirty-nine language pairs.

2 Learning Orthogonal Mappings

This section reviews learning orthogonal cross-lingual mapping between word embeddings and, along the way, introduces our notation.

We start with pre-trained word embeddings in a source language and a target language. We assume\(^1\) all embeddings are \(d\)-dimensional, and the two languages have the same vocabulary size \(n\). Let \(X \in \mathbb{R}^{d \times n}\) be the word embedding matrix for the source language, where each column \(x_i \in \mathbb{R}^d\) is the representation of the \(i\)-th word from the source language, and let \(Z \in \mathbb{R}^{d \times n}\) be the word embedding matrix for the target language. Our goal is to learn a transformation matrix \(W \in \mathbb{R}^{d \times d}\) that maps the source language vectors to the target language space. While our experiments focus on the supervised case with a seed dictionary \(D\) with translation pairs \((i,j)\), the analysis also applies to unsupervised projection.

One straightforward way to learn \(W\) is by minimizing Euclidean distances between translation pairs \((i,j)\). Formally, we solve:

\[
\min_W \sum_{(i,j) \in D} \|Wx_i - z_j\|_2^2.
\]

\(^1\)Word translation benchmarks use the same assumptions.
3 When Orthogonal Mappings Work

When are two embedding spaces easily aligned? A good orthogonal mapping is more likely if word vectors have two properties: length-invariance and center-invariance.

Length-Invariance. First, all word vectors should have the same, constant length. Length-invariance resolves inconsistencies between monolingual word embedding and cross-lingual mapping objectives (Xing et al., 2015). During training, popular word embedding algorithms (Mikolov et al., 2013b; Pennington et al., 2014; Bojanowski et al., 2017) maximize dot products between similar words, but evaluate on cosine similarity. To make things worse, the transformation matrix minimizes a third metric, Euclidean distance (Equation 1). This inconsistency is naturally resolved when the lengths of word vectors are fixed. Suppose $u \in \mathbb{R}^d$ and $v \in \mathbb{R}^d$ have the same length, then

$$u^\top v \propto \cos(u, v) = 1 - \frac{1}{2} \|u - v\|^2_2.$$ 

Minimizing Euclidean distance is equivalent to maximizing both dot product and cosine similarity with constant word vector lengths, thus making objectives consistent.

Length-invariance also satisfies a prerequisite for bilingual orthogonal alignment: the embeddings of translation pairs should have the same length. If a source word vector $x_i$ can be aligned to its target language translation $z_j = Wx_i$ with an orthogonal matrix $W$, then

$$\|z_j\|_2 = \|Wx_i\|_2 = \|x_i\|_2,$$  \hspace{1cm} (2)

where the second equality follows from the orthogonality of $W$. Equation (2) is trivially satisfied if all vectors have the same length. In summary, length-invariance not only promotes consistency between monolingual word embedding and cross-lingual mapping objective but also simplifies translation pair alignment.

Center-Invariance. Our second condition is that the mean vector of different languages should have the same length, which we prove is a pre-requisite for orthogonal alignment. Suppose two embedding matrices $X$ and $Z$ can be aligned with an orthogonal matrix $W$ such that $Z = WX$. Let $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$ and $\bar{z} = \frac{1}{n} \sum_{i=1}^n z_i$ be the mean vectors. Then $\bar{z} = W \bar{x}$. Since $W$ is orthogonal,

$$\|\bar{z}\|_2 = \|W \bar{x}\|_2 = \|\bar{x}\|_2.$$ 

In other words, orthogonal mappings can only align embedding spaces with equal-magnitude centers. A stronger version of center-invariance is zero-mean, where the mean vector of each language is zero. Artetxe et al. (2016) find that centering improves dictionary induction; our analysis provides an explanation.

4 Iterative Normalization

We now develop Iterative Normalization, which transforms monolingual word embeddings to satisfy both length-invariance and center-invariance. Specifically, we normalize word embeddings to simultaneously have unit-length and zero-mean. Formally, we produce embedding matrix $X$ such that

$$\|x_i\|_2 = 1 \quad \text{for all } i,$$  \hspace{1cm} (3)

and

$$\sum_{i=1}^n x_i = 0.$$  \hspace{1cm} (4)

Iterative Normalization transforms the embeddings to make them satisfy both constraints at the same time. Let $x_i^{(0)}$ be the initial embedding for word $i$. We assume that all word embeddings are non-zero. For every word $i$, we iteratively transform each word vector $x_i$ by first making the vectors unit length,

$$y_i^{(k)} = x_i^{(k-1)} / \|x_i^{(k-1)}\|_2.$$  \hspace{1cm} (5)

and then making them mean zero,

$$x_i^{(k)} = y_i^{(k)} - \frac{1}{n} \sum_{i=1}^n y_i^{(k)}.$$  \hspace{1cm} (6)

Equation (5) and (6) project the embedding matrix $X$ to the set of embeddings that satisfy Equation (3) and (4). Therefore, our method is a form of alternating projection (Bauschke and Borwein, 1996), an algorithm to find a point in the intersection of two closed sets by alternatively projecting onto one of the two sets. Alternating projection guarantees convergence in the intersection of two convex sets at a linear rate (Gubin et al., 1967; Bauschke and Borwein, 1993). Unfortunately, the unit-length constraint is non-convex, ruling out the classic convergence proof. Nonetheless, we use recent results on alternating non-convex projections (Zhu and Li, 2018) to prove Iterative Normalization’s convergence (details in Appendix A).
Table 1: Word translation accuracy aligning English embeddings to seven languages. We combine three normalizations—no normalization (None), mean centering and length normalization (C+L), and Iterative Normalization (IN) for five rounds—with three CLWEs: Procrustes, Procrustes with refinement (Conneau et al., 2018), and RCSLS (Joulin et al., 2018). Procrustes with C+L is equivalent to Artetxe et al. (2016). The best result for each CLWE in each column in bold. Iterative Normalization has the best accuracy of the three normalization techniques.

<table>
<thead>
<tr>
<th>Method</th>
<th>Normalization</th>
<th>JA</th>
<th>ZH</th>
<th>HI</th>
<th>TR</th>
<th>DA</th>
<th>DE</th>
<th>ES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Procrustes</td>
<td>None</td>
<td>1.7</td>
<td>32.5</td>
<td>33.3</td>
<td>44.9</td>
<td>54.0</td>
<td>73.5</td>
<td>81.4</td>
</tr>
<tr>
<td></td>
<td>C+L</td>
<td>12.3</td>
<td>41.1</td>
<td>34.0</td>
<td>46.5</td>
<td>54.9</td>
<td>74.6</td>
<td>81.3</td>
</tr>
<tr>
<td></td>
<td>IN</td>
<td>44.3</td>
<td>44.2</td>
<td>36.7</td>
<td>48.7</td>
<td>58.4</td>
<td>75.5</td>
<td>81.5</td>
</tr>
<tr>
<td>Procrustes + refine</td>
<td>None</td>
<td>1.7</td>
<td>32.5</td>
<td>33.6</td>
<td>46.3</td>
<td>56.8</td>
<td>74.3</td>
<td>81.9</td>
</tr>
<tr>
<td></td>
<td>C+L</td>
<td>13.1</td>
<td>42.3</td>
<td>34.9</td>
<td>48.7</td>
<td>59.3</td>
<td>75.2</td>
<td>82.4</td>
</tr>
<tr>
<td></td>
<td>IN</td>
<td>44.3</td>
<td>44.2</td>
<td>37.7</td>
<td>51.7</td>
<td>60.9</td>
<td>76.0</td>
<td>82.5</td>
</tr>
<tr>
<td>RCSLS</td>
<td>None</td>
<td>14.6</td>
<td>17.1</td>
<td>5.0</td>
<td>18.3</td>
<td>19.2</td>
<td>43.6</td>
<td>50.5</td>
</tr>
<tr>
<td></td>
<td>C+L</td>
<td>16.1</td>
<td>45.1</td>
<td>36.2</td>
<td>50.7</td>
<td>58.3</td>
<td>77.5</td>
<td>83.6</td>
</tr>
<tr>
<td></td>
<td>IN</td>
<td>56.3</td>
<td>48.6</td>
<td>38.0</td>
<td>52.4</td>
<td>60.5</td>
<td>78.1</td>
<td>83.9</td>
</tr>
</tbody>
</table>

Theorem 1. If the embeddings are non-zero after each iteration; i.e., $x_i^{(k)} \neq 0$ for all $i$ and $k$, then the sequence $\{X^{(k)}\}$ produced by Iterative Normalization is convergent.

All embeddings in our experiments satisfy the non-zero assumption; it is violated only when all words have the same embedding. In degenerate cases, the algorithm might converge to a solution that does not meet the two requirements. Empirically, our method always satisfy both constraints.

Previous approach and differences. Artetxe et al. (2016) also study he unit-length and zero-mean constraints, but our work differs in two aspects. First, they motivate the zero-mean condition based on the heuristic argument that two randomly selected word types should not be semantically similar (or dissimilar) in expectation. While this statement is attractive at first blush, some word types have more synonyms than others, so we argue that word types might not be evenly distributed in the semantic space. We instead show that zero-mean is helpful because it satisfies center-invariance, a necessary condition for orthogonal mappings. Second, Artetxe et al. (2016) attempt to enforce the two constraints by a single round of dimension-wise mean centering and length normalization. Unfortunately, this often fails to meet the constraints at the same time—that length normalization can change the mean, and mean centering can change vector length. In contrast, Iterative Normalization simultaneously meets both constraints and is empirically better (Table 1) on dictionary induction.

5 Dictionary Induction Experiments

On a dictionary induction benchmark, we combine Iterative Normalization with three CLWE methods and show improvement in word translation accuracy across languages.

5.1 Dataset and Methods

We train and evaluate CLWE on MUSE dictionaries (Conneau et al., 2018) with default split. We align English embeddings to thirty-nine target language embeddings, pre-trained on Wikipedia with fastText (Bojanowski et al., 2017). The alignment matrices are trained from dictionaries of 5,000 source words. We report top-1 word translation accuracy for 1,500 source words, using cross-domain similarity local scaling (Conneau et al., 2018, CSLS). We experiment with the following CLWE methods:

- **Procrustes Analysis.** Our first algorithm uses Procrustes analysis (Schönemann, 1966) to find the orthogonal transformation that minimizes Equation 1, the total distance between translation pairs.

- **Post-hoc Refinement.** Orthogonal mappings can be improved with refinement steps (Artetxe et al., 2017; Conneau et al., 2018). After learning an initial mapping $W_0$ from the seed dictionary $D$, we build a synthetic dictionary $D_1$ by translating each word with $W_0$. We then use the new dictionary $D_1$ to learn a new mapping $W_1$ and repeat the process.

3We only report accuracy for one run, because these CLWE methods are deterministic.
Relaxed CSLS Loss (RCSLS). Joulin et al. (2018) optimize CSLS scores between translation pairs instead of Equation (1). RCSLS has state-of-the-art supervised word translation accuracies on MUSE (Glavas et al., 2019). For the ease of optimization, RCSLS does not enforce the orthogonal constraint. Nevertheless, Iterative Normalization also improves its accuracy (Table 1), showing it can help linear non-orthogonal mappings too.

5.2 Training Details

We use the implementation from MUSE for Procrustes analysis and refinement (Conneau et al., 2018). We use five refinement steps. For RCSLS, we use the same hyperparameter selection strategy as Joulin et al. (2018)—we choose learning rate from \( \{1, 10, 25, 50\} \) and number of epochs from \( \{10, 20\} \) by validation.

5.3 Translation Accuracy

For each method, we compare three normalization strategies: (1) no normalization, (2) dimension-wise mean centering followed by length normalization (Artetxe et al., 2016), and (3) five rounds of Iterative Normalization. Table 1 shows word translation accuracies on seven selected target languages. Results on other languages are in Appendix B.

As our theory predicts, Iterative Normalization increases translation accuracy for Procrustes analysis (with and without refinement) across languages. While centering and length-normalization also helps, the improvement is smaller, confirming that one round of normalization is insufficient. The largest margin is on English-Japanese, where Iterative Normalization increases test accuracy by more than 40%. Figure 1 shows an example of how Iterative Normalization makes the substructure of an English-Japanese translation pair more similar.

Surprisingly, normalization is even more important for RCSLS, a CLWE method without orthogonal constraint. RCSLS combined with Iterative Normalization has state-of-the-art accuracy, but RCSLS is much worse than Procrustes analysis on unnormalized embedded mappings, suggesting that length-invariance and center-invariance are also helpful for learning linear non-orthogonal mappings.

5.4 Monolingual Word Similarity

Many trivial solutions satisfy both length-invariance and center-invariance; e.g., we can map half of words to \( e \) and the rest to \(-e\), where \( e \) is any unit-length vector. A meaningful transformation should also preserve useful structure in the original embeddings. We confirm Iterative Normalization does not hurt scores on English word similarity benchmarks (Table 2), showing that Iterative Normalization produces meaningful representations.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Before</th>
<th>After</th>
</tr>
</thead>
<tbody>
<tr>
<td>WS-353</td>
<td>73.9</td>
<td>73.7</td>
</tr>
<tr>
<td>MC</td>
<td>81.2</td>
<td>83.9</td>
</tr>
<tr>
<td>RG</td>
<td>79.7</td>
<td>80.0</td>
</tr>
<tr>
<td>YP-130</td>
<td>53.3</td>
<td>57.6</td>
</tr>
</tbody>
</table>

Table 2: Correlations before and after applying Iterative Normalization on four English word similarity benchmarks: WS-353 (Finkelstein et al., 2002), MC (Miller and Charles, 1991), RG (Rubenstein and Goodenough, 1965), and YP-130 (Yang and Powers, 2006). The scores are similar, which shows that Iterative Normalization retains useful structures from the original embeddings.

6 Conclusion

We identify two conditions that make cross-lingual orthogonal mapping easier: length-invariance and center-invariance, and provide a simple algorithm that transforms monolingual embeddings to satisfy both conditions. Our method improves word translation accuracy of different mapping-based CLWE algorithms across languages. In the future, we will investigate whether our method helps other downstream tasks.

Acknowledgments

We thank the anonymous reviewers for comments. Boyd-Graber and Zhang are supported by DARPA award HR0011-15-C-0113 under subcontract to Raytheon BBN Technologies. Jegelka and Xu are supported by NSF CAREER award 1553284. Xu is also supported by a Chevron-MIT Energy Fellowship. Kawarabayashi is supported by JST ERATO JPMJER1201 and JSPS Kakenhi JP18H05291. Any opinions, findings, conclusions, or recommendations expressed here are those of the authors and do not necessarily reflect the view of the sponsors.
References


Herbert Rubenstein and John B Goodenough. 1965. 


Chao Xing, Dong Wang, Chao Liu, and Yiye Lin. 2015. Normalized word embedding and orthogonal transform for bilingual word translation. In Conference of the North American Chapter of the Association for Computational Linguistics.


A Proof for Theorem 1

Our convergence analysis is based on a recent result on alternating non-convex projections. Theorem 1 in the work of Zhu and Li (2018) states that the convergence of alternating projection holds even if the constraint sets are non-convex, as long as the two constraint sets satisfy the following assumption:

**Assumption 1.** Let \( \mathbb{X} \) and \( \mathbb{Y} \) be any two closed semi-algebraic sets, and let \( \{(x_k, y_k)\} \) be the sequence of iterates generated by the alternating projection method (e.g., Iterative Normalization). Assume the sequence \( \{(x_k, y_k)\} \) is bounded and the sets \( \mathbb{X} \) and \( \mathbb{Y} \) obey the following properties:

(i) **three-point property of \( \mathbb{Y} \):** there exists a non-negative function \( \delta_\alpha : \mathbb{Y} \times \mathbb{Y} \to \mathbb{R} \) with \( \alpha > 0 \) such that for any \( k \geq 1 \), we have

\[
\delta_\alpha(y_k, y_{k-1}) \geq \alpha \|y_k - y_{k-1}\|_2
\]

and

\[
\delta_\alpha(y_{k-1}, y_k) + \|x_k - y_k\|_2 \leq \|x_k - y_{k-1}\|_2
\]

(ii) **local contraction property of \( \mathbb{X} \):** there exist \( \epsilon > 0 \) and \( \beta > 0 \) such that when \( \|y_k - y_{k-1}\|_2 \leq \epsilon \), we have

\[
\|x_{k+1} - x_k\| = \|P_{\mathbb{X}}(y_k) - P_{\mathbb{X}}(y_{k-1})\|_2 \leq \beta \|y_k - y_{k-1}\|_2
\]

where \( P_{\mathbb{X}} \) is the projection onto \( \mathbb{X} \).

Zhu and Li (2018) only consider sets of vectors, but our constraint are sets of matrices. For ease of exposition, we treat every embedding matrix \( X \in \mathbb{R}^{d \times n} \) as a vector by concatenating the column vectors: \( X = [x_1, x_2, \ldots, x_n] \). The \( l^2 \)-norm of the concatenated vector \( \|X\|_2 \) is equivalent to the Frobenius norm of the original matrix \( \|X\|_F \).

The two operations in Iterative Normalization, Equation (5) and (6), are projections onto two constraint sets, unit-length set \( \mathbb{Y} = \{X \in \mathbb{R}^{d \times n} : \forall i, \|x_i\|_2 = 1\} \) and zero-mean set \( \mathbb{X} = \{X \in \mathbb{R}^{d \times n} : \sum_{i=1}^n x_i = 0\} \). To prove convergence of Iterative Normalization, we show that \( \mathbb{Y} \) satisfies the three-point property, and \( \mathbb{X} \) satisfies the local contraction property.

**Three-point property of \( \mathbb{Y} \).** For any \( Y' \in \mathbb{Y} \) and \( X \in \mathbb{R}^{n \times d} \), let \( Y \) be the projection of \( X \) onto the constraint set \( \mathbb{Y} \) with Equation (5). The columns of \( Y \) and \( Y' \) have the same length, so we have

\[
\|X - Y'\|_2^2 - \|X - Y\|_2^2 = \sum_{i=1}^n \|x_i - y_i'\|^2 - \|x_i - y_i\|^2
\]

\[
= \sum_{i=1}^n 2x_i^\top y_i - 2x_i^\top y_i'. \tag{7}
\]

Since \( Y \) is the projection of \( X \) onto the unit-length set with Equation (5); i.e., \( y_i = x_i/\|x_i\|_2 \), we can rewrite Equation (7).

\[
\|X - Y'\|_2^2 - \|X - Y\|_2^2 = \sum_{i=1}^n \|x_i\|_2(2y_i^\top y_i - 2y_i^\top y_i'). \tag{8}
\]

All columns of \( Y \) and \( Y' \) are unit-length. Therefore, we can further rewrite Equation (8).

\[
\|X - Y'\|_2^2 - \|X - Y\|_2^2 = \sum_{i=1}^n \|x_i\|_2(2y_i^\top y_i - 2y_i^\top y_i').
\]

Let \( l = \min_i \{\|x_i\|_2\} \) be the minimum length of the columns in \( X \). We have the following inequality:

\[
\|X - Y'\|_2^2 - \|X - Y\|_2^2 \geq \sum_{i=1}^n l\|y_i - y_i'\|_2^2 = l\|Y - Y'\|_2^2.
\]

From our non-zero assumption, the minimum column length \( l \) is always positive. Let \( l_k \) be the minimum column length of the embedding matrix \( X^{(k)} \) after the \( k \)-th iteration. It follows that \( Y \) satisfies the three-point property with \( \alpha = \min_k \{l_k\} \) and \( \delta_\alpha(Y, Y') = \alpha \|Y - Y'\|_2^2 \).

**Local contraction property of \( \mathbb{X} \).** The zero-mean constraint set \( \mathbb{X} \) is convex and closed: if two matrices \( X \) and \( Y \) both have zero-mean, their linear interpolation \( \lambda X + (1 - \lambda) Y \) must also have zero-mean for any \( 0 < \lambda < 1 \). Projections onto convex
sets in a Hilbert space are contractive (Browder, 1967), and therefore $X$ satisfies the local contraction property with any positive $\epsilon$ and $\beta = 1$.

In summary, the two constraint sets that Iterative Normalization projects onto satisfy Assumption 1. Therefore, Iterative Normalization converges following the analysis of Zhu and Li (2018).

B Results on All Languages

Table 3 shows word translation accuracies on all target languages. Iterative Normalization improves accuracy on all languages.
# Table 3: Word translation accuracy aligning English embeddings to thirty-nine languages.

We combine three normalizations—no normalization (None), mean centering and length normalization (C+L), and Iterative Normalization (IN) for five rounds—with three CLWEs: Procrustes, Procrustes with refinement (Conneau et al., 2018), and RCSLS (Joulin et al., 2018). Procrustes with C+L is equivalent to Artetxe et al. (2016). The best result for each CLWE in each column in bold. Iterative Normalization has the best accuracy of the three normalization techniques.

<table>
<thead>
<tr>
<th>Target</th>
<th>Procrustes</th>
<th>Procrustes + refine</th>
<th>RCSLS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>None</td>
<td>C+L</td>
<td>IN</td>
</tr>
<tr>
<td>AF</td>
<td>26.3</td>
<td>28.3</td>
<td><strong>29.7</strong></td>
</tr>
<tr>
<td>AR</td>
<td>36.5</td>
<td>37.1</td>
<td><strong>37.9</strong></td>
</tr>
<tr>
<td>BS</td>
<td>22.3</td>
<td>23.5</td>
<td><strong>24.4</strong></td>
</tr>
<tr>
<td>CA</td>
<td>65.9</td>
<td>67.6</td>
<td><strong>68.9</strong></td>
</tr>
<tr>
<td>CS</td>
<td>54.0</td>
<td>54.7</td>
<td><strong>55.3</strong></td>
</tr>
<tr>
<td>DA</td>
<td>54.0</td>
<td>54.9</td>
<td><strong>58.4</strong></td>
</tr>
<tr>
<td>DE</td>
<td>73.5</td>
<td>74.6</td>
<td><strong>75.5</strong></td>
</tr>
<tr>
<td>EL</td>
<td>44.0</td>
<td>44.9</td>
<td><strong>47.5</strong></td>
</tr>
<tr>
<td>ES</td>
<td>81.4</td>
<td>81.3</td>
<td><strong>81.5</strong></td>
</tr>
<tr>
<td>ET</td>
<td>31.9</td>
<td>34.5</td>
<td><strong>36.1</strong></td>
</tr>
<tr>
<td>FA</td>
<td>33.1</td>
<td>33.7</td>
<td><strong>37.3</strong></td>
</tr>
<tr>
<td>FI</td>
<td>47.6</td>
<td>48.5</td>
<td><strong>50.9</strong></td>
</tr>
<tr>
<td>FR</td>
<td>81.1</td>
<td>81.3</td>
<td><strong>81.7</strong></td>
</tr>
<tr>
<td>HE</td>
<td>40.2</td>
<td>43.1</td>
<td><strong>43.7</strong></td>
</tr>
<tr>
<td>HI</td>
<td>33.3</td>
<td>34.0</td>
<td><strong>36.7</strong></td>
</tr>
<tr>
<td>HR</td>
<td>37.0</td>
<td>37.8</td>
<td><strong>40.2</strong></td>
</tr>
<tr>
<td>HU</td>
<td>51.8</td>
<td>54.1</td>
<td><strong>55.5</strong></td>
</tr>
<tr>
<td>ID</td>
<td>65.6</td>
<td>65.7</td>
<td><strong>67.9</strong></td>
</tr>
<tr>
<td>IT</td>
<td>76.2</td>
<td>76.6</td>
<td><strong>76.6</strong></td>
</tr>
<tr>
<td>JA</td>
<td>1.7</td>
<td>13.1</td>
<td><strong>44.3</strong></td>
</tr>
<tr>
<td>KO</td>
<td>31.5</td>
<td>32.1</td>
<td><strong>33.9</strong></td>
</tr>
<tr>
<td>LT</td>
<td>22.5</td>
<td>22.8</td>
<td><strong>23.2</strong></td>
</tr>
<tr>
<td>LV</td>
<td>23.6</td>
<td>24.9</td>
<td><strong>26.1</strong></td>
</tr>
<tr>
<td>MS</td>
<td>44.0</td>
<td>45.4</td>
<td><strong>48.9</strong></td>
</tr>
<tr>
<td>NL</td>
<td>72.8</td>
<td>73.7</td>
<td><strong>74.1</strong></td>
</tr>
<tr>
<td>PL</td>
<td>58.2</td>
<td>60.2</td>
<td><strong>60.1</strong></td>
</tr>
<tr>
<td>PT</td>
<td>79.5</td>
<td>79.7</td>
<td><strong>79.9</strong></td>
</tr>
<tr>
<td>RO</td>
<td>58.1</td>
<td>60.5</td>
<td><strong>61.8</strong></td>
</tr>
<tr>
<td>RU</td>
<td>51.7</td>
<td>52.1</td>
<td><strong>52.1</strong></td>
</tr>
<tr>
<td>SK</td>
<td>38.0</td>
<td>39.3</td>
<td><strong>40.4</strong></td>
</tr>
<tr>
<td>SL</td>
<td>32.5</td>
<td>34.3</td>
<td><strong>36.7</strong></td>
</tr>
<tr>
<td>SQ</td>
<td>23.5</td>
<td>25.1</td>
<td><strong>27.3</strong></td>
</tr>
<tr>
<td>SV</td>
<td>58.7</td>
<td>59.6</td>
<td><strong>60.7</strong></td>
</tr>
<tr>
<td>TA</td>
<td>15.1</td>
<td>15.5</td>
<td><strong>16.8</strong></td>
</tr>
<tr>
<td>TH</td>
<td>22.5</td>
<td>23.3</td>
<td><strong>23.3</strong></td>
</tr>
<tr>
<td>TR</td>
<td>44.9</td>
<td>46.5</td>
<td><strong>48.7</strong></td>
</tr>
<tr>
<td>UK</td>
<td>34.8</td>
<td>35.9</td>
<td><strong>36.3</strong></td>
</tr>
<tr>
<td>VI</td>
<td>41.3</td>
<td>42.1</td>
<td><strong>43.7</strong></td>
</tr>
<tr>
<td>ZH</td>
<td>32.5</td>
<td>42.3</td>
<td><strong>44.2</strong></td>
</tr>
</tbody>
</table>

Average: 44.7, 46.3, **48.4**, 45.3, 47.0, **49.1**, 21.8, 49.0, **50.9**