# Improving Bilingual Lexicon Induction for Low Frequency Words

**Jiaji Huang** **Xingyu Cai** **Kenneth Church**
Baidu Research, 1195 Bordeaux Dr, Sunnyvale, CA, USA, 94089
{huangjiaji, xingyucai, kennethchurch}@baidu.com

## Abstract

This paper designs a *Monolingual Lexicon Induction* task and observes that two factors accompany the degraded accuracy of bilingual lexicon induction for rare words. First, a diminishing *margin* between similarities in low frequency regime, and secondly, exacerbated *hubness* at low frequency. Based on the observation, we further propose two methods to address these two factors, respectively. The larger issue is hubness. Addressing that improves induction accuracy significantly, especially for low-frequency words.

## 1 Introduction

Bilingual Lexicon Induction (BLI) studies how to generate word-level translations from non-parallel corpora in two languages. Recently, Irvine and Callison-Burch (2017) observe that rarer words are harder to translate than frequent ones. But their BLI method is based on various "hand-crafted" features. We show that the same phenomenon occurs as well in BLI methods that are based on word embeddings. This type of methods have become especially popular in recent years (Mikolov et al., 2013; Faruqui and Dyer, 2014; Artetxe et al., 2016, 2018) and achieved state-of-art accuracies (Conneau et al., 2018).

We briefly review BLI methods that are based on word embeddings. Without loss of generality, in this paper, we focus on "supervised" BLI, which assumes that a seeding dictionary is available. Unsupervised BLI (Artetxe et al., 2018; Conneau et al., 2018) often alternates between inducing a seeding dictionary and using that to refine generated translations. Therefore, to some extent, "supervised" BLI is a key step in its "unsupervised" counterpart, and a more basic prototype to study.

Let the source space be $\mathbf{X} \triangleq [\mathbf{x}_1, \dots, \mathbf{x}_m]$, where $\mathbf{x}_i \in \mathbb{R}^d$ is the embedding vector for the $i$-th source word. Similarly, let the target space be $\mathbf{Y} \triangleq [\mathbf{y}_1, \dots, \mathbf{y}_n]$ where $\mathbf{y}_j$ is the embedding vector for the $j$-th target word. Here $m$ and $n$ are the vocabulary sizes for the two spaces. The seeding dictionary is made up of subsets of $\mathbf{X}$ and $\mathbf{Y}$, denoted as $\mathbf{X}^s = [\mathbf{x}_1^s, \dots, \mathbf{x}_S^s]$ and $\mathbf{Y}^s = [\mathbf{y}_1^s, \dots, \mathbf{y}_S^s]$ respectively, where $\mathbf{x}_k^s$ and $\mathbf{y}_k^s$ are the word embeddings of a pair of translations. $S$ is the size of seeding dictionary.

The typical supervised BLI works by first learning a transformation $\mathbf{W}$ that minimizes the discrepancy between $\mathbf{X}^s$ and $\mathbf{Y}^s$,

$$\mathbf{W} = \operatorname*{arg\,min}_{\mathbf{W} \in \mathcal{W}} \|\mathbf{W}\mathbf{X}^s - \mathbf{Y}^s\|_F^2, \tag{1}$$

where $\|\cdot\|_F$ is Frobenius norm, and $\mathcal{W}$ is a constraint set of $\mathbf{W}$. The easiest choice of $\mathcal{W}$ may be $\mathbb{R}^{d\times d}$, seen in (Mikolov et al., 2013). On the other hand, Xing et al. (2015) has observed substantial gain by letting $\mathcal{W} = \mathcal{O}(d)$, the set of orthogonal matrices. In this case, (1) is also called a *Procrustes problem*.

Once the transformation $\mathbf{W}$ is learned, translation can be induced for a word $\mathbf{x}_i$, by retrieving the Nearest Neighbor (NN) of $\mathbf{W}\mathbf{x}_i$ in $\mathbf{Y}$. Cosine distance is often adopted in the retrieval. Solving the Procrustes problem (Eq. (1)), followed by NN search is a representative framework. We use it as a baseline of this paper.

Despite the existing success in word embedding based BLI, understanding of its performance against word frequency is still lacking. This paper observes that BLI's accuracy significantly degrades for low-frequency words. Then, two factors are identified that may explain the observation. Motivated by them, we propose two methods that address each of the two factors, both improving BLI's performance in low-frequency regime.

*Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 1310–1314,
November 16–20, 2020. ©2020 Association for Computational Linguistics

## 2 Lexicon Induction at Low Frequency

We study how induction accuracies vary for words of different frequencies. Before we start, it should be emphasized that the frequency ranking of a source word and its translation(s) can differ a lot in their respective language. We term this fact as *frequency mismatch*, and the extent of mismatch also depends on the language pair. To simplify the problem, we design a "Monolingual Lexicon Induction" (MLI) task.

### 2.1 Monolingual Lexicon Induction (MLI)

MLI works with two sets of word embeddings for a single language. Given a word to be translated, the induction is supposed to retrieve the same word. The embeddings are trained respectively from two pieces of monolingual corpora (in the same language). While frequency mismatch still exists due to the differences in the two corpora, it is however reduced significantly. Compared with BLI, the induction task is also much simplified as the ground-truth is an one-to-one mapping.

We take the fasttext wiki and crawl[1] embeddings, and build a shared vocabulary of 500K words. The words are sorted from the most to least frequent according to the crawl corpora, and the order is more or less preserved in the wiki corpora. We split the 500K words into 50 frequency bins. That is, the first bin includes the 10K most frequent words. The second bin includes the next 10K most frequent words, and so on.

In each frequency bin, we randomly hold out 4K words as test words. The rest 6K are used to build seeding dictionary. To see how the size of seeds may impact induction accuracy, we vary the number of seeds sampled from the 6K words. In particular, we sample 0.02K, 0.2K and all the 6K in each frequency bin, resulting in seeding dictionaries of size 1K, 10K and 300K respectively. We ensure that any smaller seeding set is a subset of a bigger one.

An orthogonal transformation is learned using the seeds. Then, for the transformed source embeddings, nearest neighbors are retrieved in the target space. Figure 1a shows the accuracies of retrievals in each frequency bin. The accuracies drop significantly at low frequency. One may wonder if adding more seeds can help. It helps but is not very effective, as the gain diminishes quickly. Indeed, the improvement is tiny from a seeding size of 10K to 300K. In the next two subsections, we look into two statistics as diagnostics of the observation.

### 2.2 Cosine Similarities and Margin

Consider a source word $\mathbf{x}_i$, when we apply NN retrieval, it is supposed that its true translation $\text{trans}(\mathbf{x}_i)$ is the closest to $\mathbf{W}\mathbf{x}_i$, among all the candidates in $\mathbf{Y}$. In other words, we want

$$\cos(\mathbf{W}\mathbf{x}_i, \text{trans}(\mathbf{x}_i)) \geq \cos(\mathbf{W}\mathbf{x}_i, \mathbf{y}_j),$$

for $\mathbf{y}_j \neq \text{trans}(\mathbf{x}_i)$. Further define the difference between $\cos(\mathbf{W}\mathbf{x}_i, \text{trans}(\mathbf{x}_i))$ and $\max_j \cos(\mathbf{W}\mathbf{x}_i, \mathbf{y}_j)$ as a *margin* associated with $\mathbf{x}_i$, *i.e.*,

$$\begin{aligned} M(\mathbf{x}_i) &\triangleq \cos(\mathbf{W}\mathbf{x}_i, \text{trans}(\mathbf{x}_i)) \\ &\quad - \max_j \cos(\mathbf{W}\mathbf{x}_i, \mathbf{y}_j), \;\; \mathbf{y}_j \neq \text{trans}(\mathbf{x}_i). \end{aligned} \tag{2}$$

When $M(\mathbf{x}_i) < 0$, a translation error occurs. Figure 1b plots the (averaged) $M(\mathbf{x})$ values within each frequency bin. We observe that the margin decreases in low frequency regime, leading to the degraded accuracies. Again, when the seeding size increases, the margin can be enlarged, but it also saturates quickly.

### 2.3 Hubness and Tail of $k$-occurrence

Hubness is a tendency in high dimensional space that some data points, called hubs, appear to be suspiciously close to many others (Radovanovic et al., 2010). Hubness is detrimental as NN search may retrieve these hubs more often than should be.

A variable to measure the degree of hubness is $k$-occurrence. $k$-occurrence, $N_k$, is defined for any one item in the target space. It is the number of times the item being retrieved as a $k$-nearest neighbor against a query set $\mathcal{Q}$. Formally,

$$N_k(\mathbf{y}; \mathcal{Q}) = |\{\mathbf{x} \in \mathcal{Q} : \mathbf{y} \in k\text{-NN}(\mathbf{x})\}|.$$

To see the degree of hubness, one makes a histogram of $N_k(\mathbf{y}; \mathcal{Q})$ for all the $\mathbf{y}$'s in the target space. A long tail of the histogram is an indication of strong hubness. The "*tailness*" can be measured by the number of times the $N_k$ values being bigger than a threshold $n$, formally defined as

$$T_n(N_k) \triangleq |\mathbf{y} : \{N_k(\mathbf{y}; \mathcal{Q}) > n\}|. \tag{3}$$

A bigger $T_n(N_k)$ indicates longer tail of the distribution of $N_k$, hence more hubness.

[1] https://fasttext.cc/docs/en/english-vectors.html

In our MLI case, the ground-truth translation is one-to-one. Therefore if a target word $\mathbf{y}$ has a big value of $N_1(\mathbf{y})$, it is a "hub" that is incorrectly retrieved for at least $N_1(\mathbf{y}_j) - 1$ times. Note that $N_k$ and $T_n(N_k)$ both depend on the query set $\mathcal{Q}$. We vary the $\mathcal{Q}$ from the most frequent to the least words, and plot $T_2(N_1)$ values in figure 1c. For all seeding sizes, hubness becomes more prominent for low-frequency words, implying that some "hubby" target words are being retrieved more often than should be.

Summarizing this section, we have identified two statistics that may explain the inferior accuracy for low-frequency words. Moreover, adding more seeds is not very effective for improving the accuracy.

## 3 Two Methods

Motivated by the two diagnostics in the last section, we introduce two methods, each individually improving the accuracy in low frequency regime.

### 3.1 Hinge Loss for Learning Transformation

We first design a learning objective that enlarges the *margin*, as follows,

$$\begin{array}{r} \min_{\mathbf{W}\in\mathcal{O}(d)} \sum_i \sum_{j:\mathbf{y}_j\neq\text{trans}(\mathbf{x}_i^s)} \max\{0, \\ \gamma - \cos(\mathbf{W}\mathbf{x}_i^s, \mathbf{y}_i^s) + \cos(\mathbf{W}\mathbf{x}_i^s, \mathbf{y}_j)\} \end{array}, \tag{4}$$

where $\gamma > 0$ is a threshold. The objective encourages the margin $\cos(\mathbf{W}\mathbf{x}_i^s, \mathbf{y}_i^s) - \cos(\mathbf{W}\mathbf{x}_i^s, \mathbf{y}_j)$ to be bigger than $\gamma$.

It should be noticed that using hinge loss to learn the transformation is not a new idea. Examples are seen not only for BLI (Lazaridou et al., 2015), but also zero-shot image classification (Frome et al., 2013). Our difference with (Lazaridou et al., 2015) is that $\mathcal{W}$ is set to $\mathcal{O}(d)$ instead of $\mathbb{R}^{d\times d}$, as empirically we observe some gain. This is consistent with the discovery in (Xing et al., 2015), although they experiment with the Procrustes loss (Eq. (1)) instead.

We apply the hinge loss to train the orthogonal transformation, using a seeding dictionary of 10K. Accuracy is reported as the green line in figure 2a. A notable gain is observed over the Procrustes loss (blue line), especially in low frequency regime. Figure 2b validates that the margins (for low-frequency words) are indeed enlarged by adopting a hinge loss.

### 3.2 Hubless Nearest Neighbor (HNN) Search

To motivate, let us first consider a case where the translation is an one-to-one mapping. We should be able to take advantage of this strong prior, so that each target word is retrieved exactly only once. To this end, we introduce an assignment matrix $\mathbf{P} \in [0,1]^{m\times n}$ such that $P_{i,j}$ is the probability of assigning $\mathbf{y}_j$ as a translation of $\mathbf{x}_i$. By this definition,

$$\sum_j P_{i,j} = 1.$$

On the other hand, $\sum_i P_{i,j}$ measures how each $\mathbf{y}_j$ is likely to be retrieved. We want them to be equally preferred, so we constrain $\sum_i P_{i,j}$ to be uniform over all $j$. In other words,

$$\sum_i P_{i,j} = m/n.$$

Observe that $m$ does not necessarily equal $n$, so in fact we do not constrain the mapping to be one-to-one. The $\mathbf{P}$ is such that $\sum_{i,j} P_{i,j} \cos(\mathbf{W}\mathbf{x}_i, \mathbf{y}_j)$ is minimized, which can be considered as the "cost" of translation. In summary, we want to solve the following optimization problem,

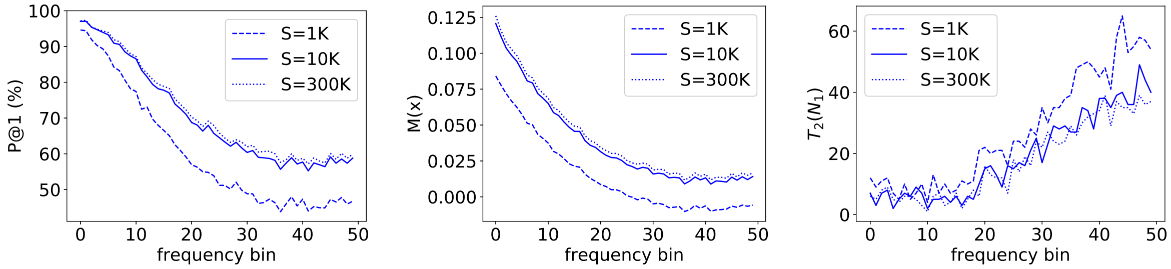
$$\begin{array}{l} \min_{\mathbf{P}\in[0,1]^{m\times n}} \sum_{i,j} P_{i,j} \cos(\mathbf{W}\mathbf{x}_i, \mathbf{y}_j) \\ s.t. \sum_j P_{i,j} = 1, \sum_i P_{i,j} = m/n \end{array}. \tag{5}$$

(5) is a linear assignment problem. It can be solved by Hungarian algorithm (Jonker and Volgenant, 1987) with cubic complexity. Recently, a more efficient solver is rediscovered in (Cuturi, 2013) by regularizing the entropy of $\mathbf{P}$,

$$\begin{array}{l} \min_{\mathbf{P}\in[0,1]^{m\times n}} \sum_{i,j} P_{i,j} \cos(\mathbf{W}\mathbf{x}_i, \mathbf{y}_j) - \epsilon H(\mathbf{P}) \\ s.t. \sum_j P_{i,j} = 1, \sum_i P_{i,j} = m/n \end{array}, \tag{6}$$

where $H(\mathbf{P}) = -\sum_{i,j} P_{i,j} \log P_{i,j}$ is entropy of $\mathbf{P}$. It is known that as $\epsilon \to 0$, the solution of problem (6) converges to that of problem (5).

Now a key challenge in solving (6) is its expensive computational cost, when $m$ and $n$ are huge. One way to circumvent that is by solving a dual problem (Genevay et al., 2016) of (6) instead. Since the math is lengthy and less relevant, we refer interested readers to a sister paper, (Huang et al., 2019), for all details. Our implementation

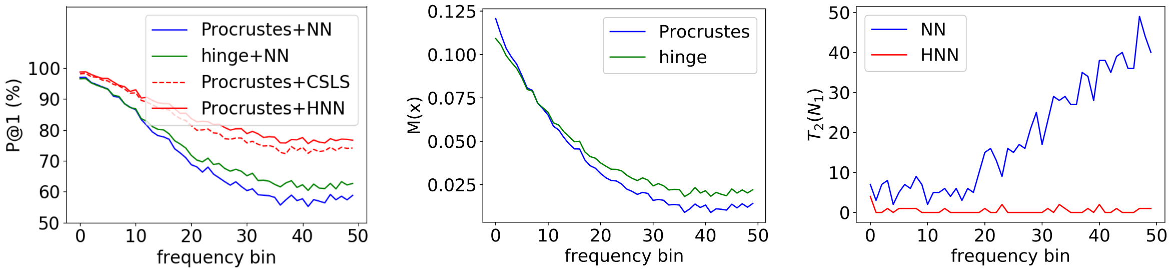


(a) Induction accuracies

(b) margins

(c) tailness of $N_1$

Figure 1: Applying the Procrustes + NN pipeline for MLI: In each plot, the x-axis are frequency bins of words, from the most (left) to least (right) frequent. All statistics are averaged within each bin. (a) Accuracy is inferior in low-frequency regime. The accuracies saturate though more seeds are used. (b) Margin decays for low-frequency words, resulting in lower accuracy. (c) *Tailness* of $N_1$ values. Hubness exacerbates in low frequency regime.

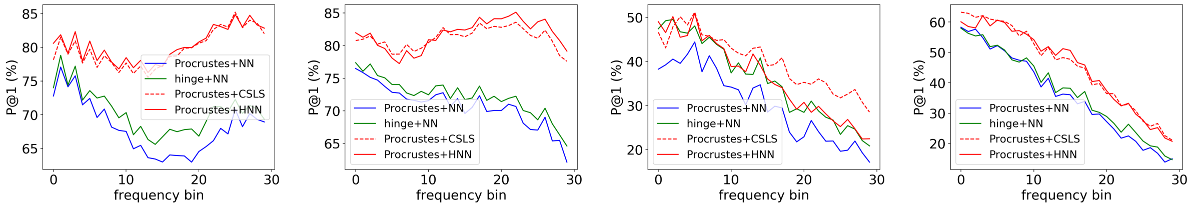


(a) Induction accuracies

(b) margins

(c) tailness of $N_1$

Figure 2: Using a seeding dictionary of 10K: (a) improved accuracy for low-frequency words by hinge loss (green line) and HNN (red line) (b) Margin increases (mostly in low-frequency regime) by using hinge loss to learn the transformation; (c) Hubness decreases significantly by using HNN.



(a) en-fr

(b) fr-en

(c) en-fi

(d) fi-en

Figure 3: BLI experiments on (a)(b) English and French; (c)(d) English and Finnish.

can be found at github[2]. Once we obtain the $\mathbf{P}$ by solving (6), translation of the $\mathbf{x}_i$ is induced as the $\mathbf{y}_j$ where $j = \arg\max_j P_{i,j}$.

We learn the transformation by solving the Procrustes problem (with 10K seeds) but replace NN with HNN search. The accuracies across all frequency bins are reported as the red line in figure 2a. It significantly outperforms Procrustes + NN, especially in the low-frequency regime. The reduced hubness is validated by the smaller $T_2(N_1)$ values in figure 2c. While HNN is effective, it is not the only method to reduce hubness. For example, CSLS (Conneau et al., 2018) is a recent state-of-art. Fig. 2a also compares our HNN (solid red line) against CSLS (dashed red line) across all frequency bins, and HNN outperforms CSLS for rare words.

### 3.3 Bilingual Experiments

We experiment with two language pairs, an easier pair (English, French) and a harder pair (En-

[2] code at https://github.com/baidu-research/HNN

glish, Finnish). The embeddings and ground-truth dictionaries are downloaded from MUSE repo[3]. We use a vocabulary of size 200K for both source and target languages. Following the same setup as in section 2.1, we create 30 frequency bins and a seeding dictionary of size 10K by uniformly sampling from each bin. The remaining words are used for test. Figure 3 shows accuracy as a function of frequency rank.

In all cases, the proposed two methods both improve upon the baseline (blue curve), and HNN shows more gain over hinge loss. However, compared with MLI (figure 2a), now the improvement seems to be more evenly distributed over all frequencies, especially on the harder language pair.

Moreover, HNN is on-par with or slightly better than CSLS for closer language pair. In contrast, en-fi (fig. 3c) is a case where CSLS works better than HNN notably. We think it is due to a strong morphology in Finnish.

## 4 Conclusion

Accuracy of bilingual lexicon induction decays for low-frequency words, as indicated by two factors: (1) diminishing *margin* between cosine similarities, and (2) exacerbated *hubness*. Two methods are proposed to address each factor. Experimental results validate their effectiveness.

## References

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2016. Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. In *the 2016 Conference on Empirical Methods in Natural Language Processing*.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *the 56th Annual Meeting of the Association for Computational Linguistics*.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word translation without parallel data. In *International Conference on Learning Representations*.

M. Cuturi. 2013. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in neural information processing systems*.

Manaal Faruqui and Chris Dyer. 2014. Improving vector space word representations using multilingual correlation. In *the 14th Conference of the European Chapter of the Association for Computational Linguistics*.

Andrea Frome, Greg Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. 2013. Devise: A deep visual-semnatic embedding model. In *Advances in Neural Information Processing Systems*.

Aude Genevay, Marco Cuturi, Gabriel Peyré, and Francis Bach. 2016. Stochastic optimization for large-scale optimal transport. In *Advances in neural information processing systems*.

Jiaji Huang, Qiang Qiu, and Kenneth Church. 2019. Hubless nearest neighbor search for bilingual lexicon induction. In *57th Annual Meeting of the Association for Computational Linguistics*.

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

Roy Jonker and Anton Volgenant. 1987. A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing*, 38(4):325–340.

Angeliki Lazaridou, Georgiana Dinu, and Marco Baroni. 2015. Hubness and pollution: Delving into cross-space mapping for zero-shot learning. In *53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*.

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

Milos Radovanovic, Alexandros Nanopoulos, and Mirjana Ivanovic. 2010. Hubs in space: Popular nearest neighbors in high-dimensional data. *Journal of Machine Learning Research*.

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

[3] https://github.com/facebookresearch/MUSE