Word vector translation: a survey with experiments

P. A. I. Forsyth
Department of Applied Mathematics
University of Waterloo
Waterloo, ON, N2L 3G1
pa2forsy@uwaterloo.ca

Abstract

After introducing word vectors, I survey word vector translation schemes. I emphasize the hubness problem and the question of whether a seed dictionary is necessary. To facilitate comparison, I implement several schemes and apply them to a common dataset.

1 Word vectors and the distributional hypothesis

The distributional hypothesis states that a word's meaning is encoded in the frequencies with which other words occur near it in natural text. These frequencies are called the word's environment. To elucidate the distributional hypothesis, Harris [1, p. 156] notes, "If we consider *oculist* and *eye-doctor* we find that, as our corpus of actually-occurring utterances grows, these two occur in almost the same environment." That *oculist* and *eye-doctor* are synonyms is revealed by their being found near the same words. Erk [2, p. 17:5] observes "the direct object of *eat* is usually a concrete object and edible." Even disregarding syntax, words frequently found near *eat* usually pertain to eating. Finally, Landauer and Dumais [3, p. 211, 222, 226] present evidence suggesting that after they have learned the tens of thousands of words the average person uses in everyday speech, children acquire the rest of their vocabulary (most words are rarely spoken) by repeatedly encountering words in context while reading, a process to which distribution is key. Clearly, a word's environment encodes a great deal of semantic information.

Many researchers (e.g. [4], [5], [6]) have devised methods for automatically extracting and storing this semantic information. I present one such method described by Mikolov et al. [6]. Let the sequence of words $(w_i)_{i=1}^{r}$ be a naturally-occurring text, called a corpus. Let V be the vocabulary of unique words occurring in the corpus[1]. For each unique word, we aim to find a vector in $\mathbb{R}^d$ that summarizes its environment and hence reveals its semantics. Thus our algorithm outputs $f : V \to \mathbb{R}^d$, the mapping between words and their associated vectors, which are called word vectors.

Together with the word vectors, our algorithm learns an auxiliary function g that maps $\mathbb{R}^d$ to the $|V|$-dimensional probability simplex, and which given a word's vector, produces an estimate of the probability of finding each other word in V near that word in the corpus. Force the kth component of g to have the form

$$(g(x))_k := \frac{\exp(\langle q_k, x\rangle)}{\sum_{j=1}^{|V|} \exp(\langle q_j, x\rangle} \tag{1}$$

where $k \in \{1, \ldots, |V|\}$ and where the vectors $q_1, \ldots, q_{|V|} \in \mathbb{R}^d$ are the parameters we determine when learning g.

[1]For a corpus to be useful, we must have $r \gg |V|$.

Submitted to Introduction to Machine Learning (CS698, 2017 Fall). Do not distribute.

To generate training data, we draw words — called predictor words — randomly from the corpus. For each predictor word, we draw a word — called a predicted word— nearby[2]. We train f and g by stochastic gradient descent so that each predicted word is probable given its predictor word[3].

The word vectors produced by this training procedure capture striking semantic information. Firstly, words with similar word vectors have similar meanings. This is a consequence of the continuity of g: if $\|f(w_1) - f(w_2)\|$ is small, then $\|g(f(w_1)) - g(f(w_2))\|$ is small, so w_1 and w_2 have similar environments, so by the distributional hypothesis they have similar meanings[4]. Secondly, algebraic relationships between word vectors correspond to semantic relationships between words. For example, it is typical to find that $f(\text{France}) \approx f(\text{England}) + f(\text{Paris}) - f(\text{London})$, which corresponds to the analogies "France is to Paris as England is to London", and "France is to England as Paris is to London". This property can be understood via the following informal analysis. Let $x_P := f(\text{Paris})$, $x_L := f(\text{London})$, $x_E := f(\text{England})$, $v := x_E + x_P - x_L$, and $1 \leq k_1, k_2 \leq |V|$. Consider $\frac{g(v)_{k_1}}{g(v)_{k_2}} = \frac{g(x_E)_{k_1}}{g(x_E)_{k_2}} \frac{g(x_P)_{k_1}}{g(x_P)_{k_2}} \frac{g(x_L)_{k_2}}{g(x_L)_{k_1}}$. The words *London* and *Paris* have similar environments, so usually $\frac{g(x_P)_{k_1}}{g(x_P)_{k_2}} \frac{g(x_L)_{k_2}}{g(x_L)_{k_1}} \approx 1$ and $\frac{g(v)_{k_1}}{g(v)_{k_2}} \approx \frac{g(x_E)_{k_1}}{g(x_E)_{k_2}}$. However, if k_1 is more associated with *Paris* and less associated with *London* than k_2, then $\frac{g(v)_{k_1}}{g(v)_{k_2}} > \frac{g(x_E)_{k_1}}{g(x_E)_{k_2}}$. Also, the reverse holds. So for most word indices k, including words pertaining to being a Western European nation, it is likely that $g(v)_k \approx g(x_E)_k$[5]. However, for words associated with Paris and Frenchness we expect $g(v)_k > g(x_E)_k$ and for words associated with London and Englishness we expect $g(v)_k < g(x_E)_k$. Thus it is reasonable to expect $g(v)$ to approximate $g(f(\text{France}))$ and so it is also reasonable (though not logically necessary) to expect $v \approx f(\text{France})$. That word vectors can algebraically express analogies suggests that they align vector space structure with semantic structure[6].

2 Translation using word vectors

If a group of words have a certain semantic relationship, their translations have the same semantic relationship. By the arguments of the previous section, this semantic consistency between languages implies an algebraic consistency between the corresponding word vectors. For example, if f maps English words to their word vectors and g maps French words to their word vectors, then just as $f(\text{king}) - f(\text{man}) + f(\text{woman}) \approx f(\text{queen})$ so $g(\text{roi}) - g(\text{homme}) + g(\text{femme}) \approx g(\text{reine})$. Hence if W maps the word vectors of one language to the word vectors of their translations, we should expect W to be linear[7].

Mikolov et al. [8] use this observation to devise a method for expanding small bilingual lexicons, which we describe next. Assume we have separately found word vectors for two languages, called the source and target. Let the columns of $U \in \mathbb{R}^{d_1 \times n_1}$ be the source word vectors, and let the columns of $Z \in \mathbb{R}^{d_2 \times n_2}$ be the target word vectors. Use $D \in \{0, 1\}^{n_1 \times n_2}$ to represent the bilingual lexicon, setting $D_{i,j} = 1$ if word i of our source language translates to word j in our target language. Note that since our lexicon is small, most columns and rows of D will be zero. Also note that we do not assume translations are one-to-one. We seek $W \in \mathbb{R}^{n_1 \times n_2}$ solving

$$W \in \underset{W \in \mathbb{R}^{n_1 \times n_2}}{\operatorname{argmin}} \sum_{i,j} D_{i,j} \|WU_{:,i} - Z_{:,j}\|^2. \tag{2}$$

[2]Usually one picks a window size about five, meaning that an instance of word B is near an instance of word A if it occurs less than five words before or less than five words after it.

[3]Thus letting near(i) denote the indices in V of the words occurring near the ith word in the corpus, the global objective function to be minimized is $-\sum_{i=1}^{r} \sum_{k \in \text{near}(i)} \log g(f(w_i))_k$.

[4]Unfortunately the reverse is not guaranteed. Suppose we are in $\mathbb{R}^2$ with $|V| = 2$ and $q_1 = (0, 1)$, $q_2 = (1, 0)$. Then as $\alpha \to \infty$, the frequencies associated with the word vectors $(\alpha, \frac{\alpha}{2})$ and $(\alpha, 0)$ become identical, even though the vectors differ.

[5]The jump from statements about ratios to statements about raw probabilities requires our assumption that $\frac{g(x_P)_{k_1}}{g(x_P)_{k_2}} \frac{g(x_L)_{k_2}}{g(x_L)_{k_1}}$ is usually 1 and is never enormous or tiny.

[6]See section 3 of Pennington et al. [7] for a word vector algorithm derived using this kind of argument.

[7]We have actually only argued that W needs to preserve vector addition or subtraction. By repeated vector addition one can argue it must preserve multiplication by integers, and hence multiplication by rationals. If we also assume W is continuous, then it follows it must preserve multiplication by real numbers, and so must be linear.

Table 1: Results from my implementations of word vector translation schemes. S is seed dictionary size. A_1, A_5, A_{10} are top-one, top-five, and top-ten accuracy. H_{20}^{10} is the average 20-neighbour hubness score of the top 10 hubs in the target language when the test set is mapped from the source language. The dataset is the English-Italian Europarl data released by Dinu et al. [9]. In all CSLS examples, $k = 10$. See text for method details.

Row	Method	S	A_1	A_5	A_{10}	H_{20}^{10}
1.1	Mikolov et al. [8] + norm	5000	0.3380	0.4833	0.5393	19.8
1.2	Mikolov et al. [8]	5000	0.3493	0.4907	0.5453	17.2
1.3	Procrustes [10]	5000	0.3673	0.5280	0.5860	13.2
1.4	Procrustes +norm [11]	5000	0.3687	0.5273	0.6340	13
1.5	GC [9]	5000	0.3553	0.5280	0.5840	3.1
1.6	Artetxe et al. [12]	25	0.3787	0.5360	0.5913	17
1.7	GC [9] + 2000 pivots	5000	0.3800	0.5620	0.6240	-
1.8	Procrustes + GC	5000	0.3833	0.5400	0.5913	3.4
1.9	Procrustes + GC + 2000 pivots	5000	0.3927	0.5633	0.6260	-
1.10	Procrustes+ centering [12]	5000	0.3927	0.5633	0.6173	12.4
1.11	Procrustes + CSLS [13]	5000	0.4540	0.6160	0.6607	6.1

Let $((i_1, j_1), (i_2, j_2), \ldots, (i_m, j_m))$ be the indices of all the nonzero entries of D. Let $X \in \mathbb{R}^{d_1 \times m}$ be the matrix whose kth column is $U_{:,i_k}$, and let $Y \in \mathbb{R}^{d_2 \times m}$ be the matrix whose kth column is $Z_{:,j_k}$. Then (2) is equivalent to [8]

$$W \in \underset{W \in \mathbb{R}^{n_1 \times n_2}}{\operatorname{argmin}} \|WX - Y\|^2, \tag{3}$$

which we solve by least-squares[9]

$$WXX^* = YX^*. \tag{4}$$

Given a vector $u_i \in \mathbb{R}^{n_1}$ representing a word in the source language not present in the seed dictionary D (i.e. $D_{i,:} = 0$), we find k candidate translations by taking the words corresponding to the k columns of Z closest to Wu_i according to cosine similarity (for $x, y \neq 0$, $\operatorname{cossim}(x, y) = \langle x, y \rangle / (\|x\| \|y\|)$). Denote these k candidates by $\mathrm{NN}_k(Wu_i, Z)$. To test the translation method, pick a test set of words from the source vocabulary whose translations are known and compute $\mathrm{NN}_k(Wu_i, Z)$ for each u_i in the test set. The translation of u_i is a success if $\mathrm{NN}_k(Wu_i, Z)$ contains the word vector of the correct translation.

I implemented this method and tested it on the dataset published by Dinu et al. [9][10], which consists of 200000 300-dimensional word vectors derived by the method of Mikolov et al. [6] from the European parliament corpus for both English and Italian. The dataset also includes a training and test set, of which I make use. The training set consists of 5000 high frequency word pairs, while the test set consists of 1500 word pairs drawn from 5 frequency bins. The results are in row 1.2 of Table 1. The top-one accuracy of about 35 percent is impressive, and empirically supports the theoretical arguments by which the method was derived.

As rest of this essay is devoted to schemes that, like the one above, make use of word vectors for word translation. it seems appropriate to mention the uses of such schemes. The most obvious use is the automatic generation of large bilingual dictionaries between language pairs for which such data is scarce. A second use is in the transference of a model learned on word vectors of one language to word vectors of another for which less data is available. As noted by Artetxe et al. [10], examples of models that lend themselves to this kind of transfer include parsing, document classification, and part-of-speech tagging. Lastly, a word translation scheme can be used as a baseline against which to compare translation schemes that operate on larger units of text.

[8] In this essay, $\|\cdot\|$ or $\langle \cdot, \cdot \rangle$ applied to matrices always denote the Frobenius norm or inner product.

[9] This always has a solution since $\mathbb{R}^d XX^* = \mathbb{R}^m X^*$ since $\{y \in \mathbb{R}^m : yX^* = 0\} \perp \mathbb{R}^d X$.

[10] See http://clic.cimec.unitn.it/~georgiana.dinu/down/

3 Learning an isometric map

As (1) reveals, the semantic information captured by word vectors is encoded in their inner-products. Thus if we seek to learn a linear translation map W between the word vectors of two languages, it is reasonable to force our map to preserve these inner products. This amounts to forcing W to be an isometry (i.e. an orthogonal matrix), in which case we can directly solve (2) by noting that

$$\sum_{i,j} D_{i,j} \|WU_{:,i} - Z_{:,j}\|^2 = \sum_{i,j} D_{i,j} \left(\|U_{:,i}\|^2 + \|Z_{:,j}\|^2 - 2\langle WU_{:,i}, Z_{:,j}\rangle\right) \tag{5}$$

implies that $W \in \mathbb{R}^{d\times d}$ minimizes $\|WU_{:,i} - Z_{:,j}\|^2$ if and only if it maximizes $\sum_{i,j} D_{i,j}\langle WU_{:,i}, Z_{:,j}\rangle = \langle W, ZD^*U^*\rangle$. Let $ZD^*U^* = \sum_{i=1}^{d} \sigma_i a_i b_i^T$ be the singular value decomposition. Then by Cauchy-Schwarz,

$$\langle W, ZD^*U^*\rangle = \sum_i \sigma_i \langle Wb_i, a_i\rangle \leq \sum_i \sigma_i \|Wb_i\|\|a_i\| = \sum_i \sigma_i. \tag{6}$$

By the orthonormality of $\{a_i\}_{i=1}^d$ and $\{b_i\}_{i=1}^d$ we can achieve this bound by setting $W = \sum_{i=1}^{n} a_i b_i^T$. Thus we have found an optimal translation map W[11].

Some authors consider normalizing word vectors, either during word vector training [11] or afterward [10], to force the Euclidian distance, by which W is learned, to agree with cosine similarity, according to which nearest-neighbors are found. I tried learning an isometric map with and without normalization. My results in rows 1.3 and 1.4 of Table 1, which agree with those of Artetxe et al. [10, p. 2292], show that while forcing W to be an isometry yields an accuracy increase, normalization has minimal effect. Indeed, when W is not forced to be isometric, normalization decreases accuracy, as row 1.1 of Table 1 shows. It may be that normalization imposed during training would be more beneficial.

4 Hubness

We next focus on the nearest-neighbor strategy by which transformed source-language vectors are matched with target-language words. Let $Q \subset \mathbb{R}^d$. For any $x \in \mathbb{R}^d$ and for any positive integer k, let $\mathrm{NN}_k(x, Q)$ denote the k points in Q closest to x (breaking ties arbitrarily). Furthermore, let $Q' \subset \mathbb{R}^d$ and define for any $y \in Q$,

$$\mathrm{H}_k(y, Q', Q) := |\{x \in Q' : y \in \mathrm{NN}_k(x, Q)\}|, \tag{7}$$

that is $H_k(y, Q', Q)$ denotes the number of points $x \in Q'$ such that y is on the k nearest neighbor list of x. A point $y \in Q$ whose $\mathrm{H}_k(y, Q', Q)$ is much larger than that of most points in Q is called a hub. Radovanic et al. [14] and other researchers have observed empirically that high dimensional datasets often have hubs, and that hubs can impede algorithms relying on neighbor retrieval. While the definitive theoretical treatment of hubness has yet to be written, Theorem 3 in Newman et al. [15] and Theorem 1 in Radovanovic et al. [16], suggest that hubness may be a fundamental property of many distributions in high dimensional spaces. These theorems do not apply directly to our case, because we do not know the distribution of our data, and because cosine similarity does not satisfy the hypothesis required of the distance function in the theorems[12]. Nevertheless, Dinu et al. [9] observed that hubness is often a problem in the automatic translation methods we have discussed: certain words in the target language are inappropriately chosen as the translation for many source words. These hubs are often low-frequency specialized words. For example, when I applied Mikolov et al.'s [8] method to Italian to English translation using Dinu et al.'s [9] Europarl data, I found that the rare English words *Harsnet*, *Jalilabad*, and *Soviet-backed* were on the 10-nearest neighbor lists 70, 36, and 27 mapped test set words.

[11] The problem of finding an orthogonal matrix that best maps one list of vectors to another is called the Procrustes problem, an allusion to a mythical Greek torturer.

[12] If we normalize our data so that cosine similarity is equivalent to Euclidian distance, then our distance function becomes admissible, but the distribution of our word vectors (on the surface of the unit sphere) becomes inadmisable.

4.1 Hubness mitigation

The nearest neighbor relation used in our translation scheme is asymmetric, in that while a point can only have one nearest neighbor (ignoring ties), it can be nearest neighbor to many points. One could correct this asymmetry by looking for translation pairs in which the target word vector and the mapped source word vector are mutual nearest neighbors. This would eliminate the hubness problem, but, since not every vector is the nearest neighbor of its nearest neighbor, it would prevent us from translating many words. Next, I discuss two methods for addressing the hubness problem which attempt to approximate the notion of mutual nearest neighbors without sacrificing the ability to translate an arbitrary source word.

Dinu et al. [9] suggests an approach called Global Correction (GC), which I will describe. Let $Q', Q \subset \mathbb{R}^d$ let $z \in Q', y \in Q$. Define

$$\mathrm{order}(z, y, Q') := \min\{k \in \mathbb{N} : z \in \mathrm{NN}_k(y, Q')\}, \tag{8}$$

that is, $\mathrm{order}(z, y, Q')$ is the rank of z on y's nearest neighbor list. Define for $z \in Q'$

$$\mathrm{gcscore}(z, y, Q') := \mathrm{order}(z, y, Q') - \mathrm{cossim}(z, y). \tag{9}$$

Now let W be a linear map derived by one of the above methods, let $Q := \{y_1, \ldots, y_{n_2}\}$ be the target vocabulary, and let $Q' := \{Wx_1, \ldots, Wx_m\}$ be the mapped test set. Then to find k candidate translations for a word vector x, we take the words corresponding to the k points y in Q with smallest $\mathrm{gcscore}(Wx, y, Q')$. The GC scheme approximates the notion of mutual nearest neighbors by, for a given mapped point Wx, finding the point nearest to it among those points to which it is nearest (note that order yields an integer while $-1 \leq \mathrm{cossim} \leq 1$). We should expect GC to perform better when the test set Q' is larger, since in this case $\mathrm{order}(x, y, Q')$ is more informative. As Dinu et al. [9] note, one way to achieve this with a fixed test set is to simply add extra mapped words called pivots to Q' for the purposes of computing $\mathrm{order}(x, y, Q')$. I implemented this scheme. The results with 0 pivots and 2000 pivots are in rows 1.5 and 1.7 of Table 1.

Conneau et al. [13] introduce another approach to hubness reduction using a new similarity function called *Cross-domain similarity local scaling* (CSLS). To define it, fix a positive integer k, and assume we have normalized word vectors and an isometric transform W. Let $P := \{x_1, \ldots, x_{n_1}\}$ denote the source vocabulary, and let $Q := \{1, \ldots, y_{n_2}\}$ denote the target vocabulary. Define the functions $r_P : Q \to \mathbb{R}$ and $r_Q : P \to \mathbb{R}$ by

$$r_P(y) := \frac{1}{k} \sum_{x \in \mathrm{NN}_k(W^* y, P)} \mathrm{cossim}(x, W^* y), \quad r_Q(x) := \frac{1}{k} \sum_{y \in \mathrm{NN}_k(Wx, Q)} \mathrm{cossim}(Wx, y). \tag{10}$$

r_P and r_Q measure the average cosine similarity of a point in one domain to its neighborhood in the other domain. We should generally expect r_P and r_Q to be large for hubs and small for isolated points. Finally, define $\mathrm{CSLS}_W : P \times Q \to \mathbb{R}$ by

$$\mathrm{CSLS}_W(x, y) = 2\,\mathrm{cossim}(Wx, y) - r_Q(x) - r_P(y). \tag{11}$$

To translate a word with word vector x, we compute the isometric map W from the seed dictionary as before, and then find x's nearest neighbor according to CSLS_W. Note that we need not compute $r_Q(x)$ since this term will be the same for every y whose similarity with x we measure. I implemented the CSLS algorithm, and the results are shown in row 1.11 of Table 1 ($K = 10$). Since points tend to be similar according to the CSLS measure if their cosine similarity to each other exceeds their cosine similarity to their neighborhoods, CSLS, like the GC scheme, approximates the notion of mutual nearest neighbors.

While the CSLS scheme depends on the map W being an isometry, the GC scheme does not constrain W. To clarify the comparison, I modified the GC scheme to force an isometric W. The resulting scheme has improved performance (see rows 1.8 and 1.9 of Table 1), but is still inferior to the CSLS scheme. A possible explanation is that the GC scheme is rigid, in that no matter how close a target word vector is to a mapped source word vector, the target word vector cannot be its nearest neighbor if there is another target word vector assigning it a lower order. In contrast, the CSLS scheme is flexible, trading off hubness information against distance information.

To further investigate, I computed the statistic H_{20}^{10}, the average hubness of the top 10 hubs, for various methods (Table 1). As expected, the methods with hubness reduction have lower H_{20}^{10} than the methods without. Interestingly, GC has lower H_{20}^{10} than CSLS even though CSLS is more accurate. This may support my earlier analysis: the GC method prioritizes hubness at the cost of accuracy.

5 Overcoming the need for a seed lexicon

The word vectors of a given language form a highly complex configuration of points in a high-dimensional Euclidian space. The problem of word translation is to, as best as possible, align one such configuration with another. So far we have used information from seed dictionaries to facilitate alignment, but one could also align using the shapes of the two configurations themselves. In so doing, one might reduce dependence on the seed dictionary, or eliminate it entirely.

Artetxe et al. [12] approach the problem of word translation with a small seed dictionary by viewing (3) as a sub-problem of a larger problem. To be more precise, let

$$\mathcal{D} := \{D \in \{0,1\}^{n_1 \times n_2} : \text{for all } i \in \{1,\dots,n_1\} \text{ there is a unique } j \in \{1,\dots,n_2\} \text{ such that } D_{i,j} = 1\} \tag{12}$$

be the set of valid dictionaries. Note that in (12) we are assuming that valid dictionaries map each source word to exactly one target world. We aim to solve

$$\operatorname*{argmin}_{D\in\mathcal{D}} \min_{W\in O(d)} \sum_{i_{i=1}}^{n_1} \sum_{j=1}^{n_2} D_{i,j} \|WU_{:,i} - Z_{:,j}\|^2 \tag{13}$$

in which we optimize over the set of valid bilingual dictionaries $\mathcal{D}$ and for each such dictionary optimize over the orthogonal matrices $O(d)$, attempting to find the one that best realizes the dictionary.

The authors propose the alternating minimization scheme Algorithm 1, which is similar to algorithms that have been used for 3D point cloud alignment in engineering problems [17]. The idea is to alternately update W to best realize D, and then update D so that each word vector translates to its nearest neighbor under the mapping W. I implemented algorithm 1 using 25 random words from

while *improvement in* $\operatorname{tr} YD^*X^*T^*$ *greater than threshold* **do**
- $W \leftarrow \operatorname{argmin}_{W\in O(d)} \sum_{i_{i=1}}^{n_1} \sum_{j=1}^{n_2} D_{i,j} \|WX_{:,i} - Y_{:,j}\|^2$;
- $D_{i,j} = 0_{n_1 \times n_2}$;
- **for** $i = 1 \dots n_1$ **do**
  - $j = \operatorname{argmax}_j \operatorname{cossim}(Wx_i, y_j)$;
  - $D_{i,j} = 1$;

Algorithm 1: Alternating minimization scheme for bilingual dictionary construction with a small seed dictionary.

the 5000-word English-Italian Europarl training dictionary as the seed. I found that the convergence of the algorithm was dependent on a preprocessing step: Artetxe et al. center the vectors in each language so that their mean is zero before applying their algorithm[13]. Without mean centering, the algorithm produces a dictionary with 0 test accuracy on every iteration after the first, and appears to converge to such a dictionary[14]. With mean centering, it converges to a high-quality dictionary (see row 1.6 of Table 1). This suggests multiple local minima are present, and that mean-centering directs the algorithm to the correct one.

To further investigate the effect of mean centering, I applied it to the standard Procrustes translation procedure without an iterative component, reproducing the results of Artetxe et al. [10, p. 2292] (see row 1.10 of Table 1). As Artetxe et al. observed, it yields a significant boost to accuracy. Artetxe et al. [10] explain mean centering as a means of ensuring that the expected inner product of any two vectors in the same language is zero. It is possible that this improves the quality of the learned isometric mapping W by ensuring that for each vector x_i in a language, there are only a small number of vectors x_j in the same language whose images Wx_j severely restrict the value of Wx_i[15].

For comparison, I tried the Mikolov et al. method with a random 25 word dictionary and got 0 accuracy, confirming results from Artexe et al. [12, p. 456]. Interestingly, the alternating minimization scheme produces an accurate dictionary but nevertheless has a high H_{20}^{10} score. This may indicate

[13] Note that after this operation, word vectors are no longer normalized.

[14] It could also be converging to a better dictionary very slowly.

[15] Matlab experiments reveal that the inner product of a vector in a given language with a random vector in the same language has a distribution concentrated around its mean.

that the algorithm is achieving accuracy independently of hubness reduction. If so, one might use this insight to design a high quality word vector translation algorithm by attempting combine both types of information and reduce hubness while matching word vector distributions.

5.1 Generative adversarial net

Observing that generative adversarial nets specialize in aligning distributions, Conneau et al. [13] apply one to the problem of word vector translation without a seed dictionary. They achieve accuracy comparable to that of the best methods requiring a seed dictionary. Key to their approach is a CSLS-based measure of the similarity of two distributions, which they use to adjust their gradient-descent step size. I implemented their algorithm and applied it to several toy problems, but did not have time to tune and run the GAN on linguistic data. Some observations: in toy problems I constructed, the algorithm failed when the distributions did not initially overlap. This behavior was identical when I modified the algorithm to use a Wasserstein GAN instead of standard GAN. It may be that when the distributions do not sufficiently overlap, the discriminator becomes extremely effective quickly, preventing the generator from learning.

6 Conclusions and future work

In this essay I surveyed word vector translation, attempting to offer insight based on experiments. Of the methods using full seed dictionaries, the Procrustes CSLS methods was the best by far, balancing similarity and hubness information. Of the methods I implemented, only that of Artetxe et al. [12] could handle small seed dictionaries, but the results of Conneau et al. [13, p. 7]) indicate their GAN can beat it at this task. I finish with some suggestions for future work. One conclusion I can draw from Table 1 is that hubness is responsible for a significant portion of the differences in accuracy between methods (though it is not the only factor: see row 1.6). Hubness, however, is poorly understood. There is an opportunity for a cunning theoretician to give it a firmer foundation, and provide rigorous justification for the performance of neighbor-retrieval-based algorithms.

As another observation, I note that the algorithms surveyed here can be divided into two classes: those that find the linear transformation W using only a seed dictionary (Mikolov et al. [8], Procrustes, and their variants), and those that find W by directly attempting to align the two word vector configurations, using the seed dictionary only for initialization (Artetxe et al. [12]) or not at all (Conneau et al. [13]). There is room for another class of algorithm, which would attempt to align the word vector configurations, but which would never forget the seed dictionary. Such an algorithm would involve the optimization of the sum of two terms: one measuring the degree of alignment of the two word vector configurations, and one measuring faithfulness to the original seed dictionary. To go further, one could observe that besides Conneau et al.'s GAN [13] and the alternating minimization algorithm of Artetxe et al. [12], all algorithms discussed here have two stages. In the first, they learn a linear transformation between the Euclidian spaces of the two languages. In the second, they match the mapped source vectors to the target vectors using some measure of proximity. One could combine the two stages and directly learn a mapping between source vectors and target vectors, perhaps minimizing an objective measuring the sum of hubness, the degree to which the mapping differs from an isometry, and unfaithfulness to a seed dictionary. Unfortunately, such an algorithm would likely be a combinatorial nightmare, but perhaps a relaxation could be found.

I did not have time to apply my implementation of Conneau et al.'s GAN [13] to language data, but had I been able to do so, I would have liked to have measured how GAN training interacts with hubness. Artexe et al.'s algorithm [10], the GAN's main competitor, achieves good accuracy despite significant hubness. It would be interesting to see to what extent this is also true of the GAN, especially given that the GAN algorithm includes refinement steps based on the CSLS hubness reduction scheme.

Finally, consider the analogy between word vector translation and the image registration problem. In this analogy, the Mikolov et al. [8] and Procrustes word-translation methods correspond to anchor-point based image registration, while the alternating minimization scheme of Artetxe et al. [12] resembles an iterative closest point (ICP) type registration algorithm. The literature on image registration is vast (see [18]), with algorithms ranging from those based on physical processes to those justified by statistical consideration. It is probable that some of these algorithms are ripe for exportation to other domains.

References

[1] Zellig S Harris. Distributional structure. *Word*, 10(2-3):146–162, 1954.

[2] Katrin Erk. What do you know about an alligator when you know the company it keeps? *Semantics and Pragmatics*, 9:17–1, 2016.

[3] Thomas K Landauer and Susan T Dumais. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, 104(2):211, 1997.

[4] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155, 2003.

[5] Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391, 1990.

[6] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

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

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

[9] Georgiana Dinu, Angeliki Lazaridou, and Marco Baroni. Improving zero-shot learning by mitigating the hubness problem. *arXiv preprint arXiv:1412.6568*, 2014.

[10] Mikel Artetxe, Gorka Labaka, and Eneko Agirre. Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. In *EMNLP*, pages 2289–2294, 2016.

[11] Chao Xing, Dong Wang, Chao Liu, and Yiye Lin. Normalized word embedding and orthogonal transform for bilingual word translation. In *HLT-NAACL*, pages 1006–1011, 2015.

[12] Mikel Artetxe, Gorka Labaka, and Eneko Agirre. Learning bilingual word embeddings with (almost) no bilingual data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 451–462, 2017.

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

[14] Milos Radovanović, Alexandros Nanopoulos, and Mirjana Ivanović. On the existence of obstinate results in vector space models. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 186–193. ACM, 2010.

[15] Charles M Newman and Yosef Rinott. Nearest neighbors and voronoi volumes in high-dimensional point processes with various distance functions. *Advances in Applied Probability*, 17(4):794–809, 1985.

[16] Miloš Radovanović, Alexandros Nanopoulos, and Mirjana Ivanović. Hubs in space: Popular nearest neighbors in high-dimensional data. *Journal of Machine Learning Research*, 11(Sep): 2487–2531, 2010.

[17] Anatoliy Kats and Mark McCartin-Lim. Point cloud alignment, 2009.

[18] Aristeidis Sotiras, Christos Davatzikos, and Nikos Paragios. Deformable medical image registration: A survey. *IEEE transactions on medical imaging*, 32(7):1153–1190, 2013.