Knowledge Distillation for Bilingual Dictionary Induction

Ndapandula Nakashole and **Raphael Flauger**
Computer Science and Engineering
University of California, San Diego
La Jolla, CA 92093
nnakashole@eng.ucsd.edu

Abstract

Leveraging zero-shot learning to learn mapping functions between vector spaces of different languages is a promising approach to bilingual dictionary induction. However, methods using this approach have not yet achieved high accuracy on the task. In this paper, we propose a bridging approach, where our main contribution is a knowledge distillation training objective. As teachers, rich resource translation paths are exploited in this role. And as learners, translation paths involving low resource languages learn from the teachers. Our training objective allows seamless addition of teacher translation paths for any given low resource pair. Since our approach relies on the quality of monolingual word embeddings, we also propose to enhance vector representations of both the source and target language with linguistic information. Our experiments on various languages show large performance gains from our distillation training objective, obtaining as high as 17% accuracy improvements.

1 Introduction

In traditional supervised learning, a classifier is trained on a labeled dataset of the form $(\mathbf{X}, \mathbf{Y})$. Each $x_i \in X$ is a feature vector representing a single training instance and $y_i \in Y$ is the label associated with x_i. In zero-shot learning (Mitchell et al., 2008), at test time we can encounter a test instance x_j whose corresponding label was not seen at training time. This setting occurs in domains where $\mathbf{Y}$ can take on many values, and obtaining labeled examples for all possible Y values is expensive. Computer vision is one such domain, where there are thousands of objects a system needs to recognize yet at training time we may only see examples of some of the objects. In zero-shot learning, instead of learning parameters associated with each possible label in Y, the learning task is cast as a problem of learning a single mapping function from the vector space of input instances to the vector space of the output labels. The resulting induced function can then be applied to test instances x_j whose labels may not have been seen at training time, producing a projected vector, $\hat{y}_j$, in the label space. The nearest neighbor of the mapped vector in the label space is then considered to be the label of x_j.

In this paper, we study zero-shot learning in the context of bilingual dictionary induction, which is the problem of mapping words from a source language to equivalent words in a target language. The label space is the full vocabulary of the target language which can be on the order of millions of tokens. First, word embeddings are learned separately for each language, and second, using a given seed dictionary, we train a mapping function to connect the two monolingual vector spaces, thereby facilitating bilingual dictionary induction. The advantage of zero-shot learning is that it can help reduce the amount of labeled data for applications with many possible labels, such as the application we study in this paper, bilingual dictionary induction. However, the state-of-the-art accuracy on zero-shot bilingual dictionary induction is still low. On the task of English to Italian $(en \rightarrow it)$, top-1 and top-10 accuracies are around 40% and 60%, respectively (Lazaridou et al., 2015; Dinu et al., 2014).

An important aspect of zero-shot learning for bilingual dictionary induction is that, it relies on availability of a large seed dictionary[1]. Such large

[1] 5000 seed pairs for the $(en \rightarrow it)$ dataset.

Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 2497–2506
Copenhagen, Denmark, September 7–11, 2017. ©2017 Association for Computational Linguistics

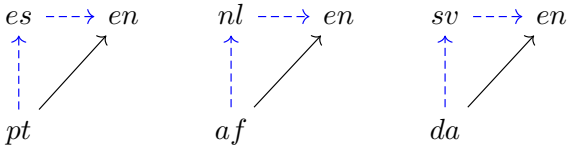

Figure 1: Trilingual paths for Portuguese(pt) to English(en) via Spanish (es), Afrikaans(af) to (en) via Dutch (nl), and Danish(da) to (en) via Swedish(sv).

training dictionaries might not be available for all languages. However, for a given language with only a small seed dictionary, there could be a highly related language with a much larger seed dictionary. For example, we might have a small seed dictionary for translating Portuguese to English ($pt \rightarrow en$), but a large seed dictionary for translating Spanish to English language ($es \rightarrow en$). At training time, we can train the ($pt \rightarrow en$) mapping function not only using the small seed dictionary, but also make use of the trilingual path going through Spanish, ($pt \rightarrow es \rightarrow en$). Since pt and es are highly related, a small amount of data may be sufficient to learn the projection ($pt \rightarrow es$). This is the idea of using a bridge or pivot language in machine translation (Utiyama and Isahara, 2007). Our contribution is a knowledge distillation training objective function that encourages the mapping function ($pt \rightarrow en$) to predict the true English target words as well as to match the predictions of the trilingual path ($pt \rightarrow es \rightarrow en$) within a margin. This is approach allows seamless Example trilingual paths are shown in Figure 1.

By setting up our objective function in this way, we are distilling knowledge (Bucilu et al., 2006; Hinton et al., 2015) from the trilingual paths to train a single mapping function for ($pt \rightarrow en$). In our experiments, we show performance gains for several language pairs, 17% for top-10 precision for ($pt \rightarrow en$). We also show that, for a given language pair, our objective seamlessly allows us to distill from several related languages. Moreover, we learn weights for each of the distillation paths, thereby automatically learning indicative weights of how useful each distillation path is. Finally, we show that even when we only use unlabeled data to distill knowledge from trilingual paths, we still obtain performance gains over a model trained on a small seed dictionary.

Since our approach relies on the quality of monolingual word embeddings, we also propose to enhance vector representations of both the source and target language with linguistic information. In particular, we augment word vectors with additional dimensions capturing corpus statistics of part-of-speech tags of words. Second, we model sub-word information in the vector representations of words.

2 Related Work

Cross Vector Space Mapping with Seed Dictionaries. Our work is most related to models that do zero-shot learning for bilingual dictionary induction, using maps between vector spaces with seed dictionaries as training data. Examples include the models of (Mikolov et al., 2013; Dinu et al., 2014; Lazaridou et al., 2015; Vulic and Korhonen, 2016). Like these approaches, we first learn word embeddings for each language, then use a seed dictionary to train a mapping function between the two vector spaces. In a departure from these prior methods, we propose to distill knowledge from trilingual paths of nearby languages for languages with small seed dictionaries using a distillation training objective. Additionally, we model linguistic information in the vector space of the source and target languages. Another line of research in this vein is the work of (Vulic and Korhonen, 2016), who analyze how properties of the seed dictionary affect bilingual dictionary induction across different dimensions (i.e., lexicon source, lexicon size, translation method, translation pair reliability). However, methodologically, their approach is based on prior work (Mikolov et al., 2013; Dinu et al., 2014).

Bilingual word embeddings. There is a rich body of work on bilingual embeddings. Bilingual word embedding learning methods produce a shared bilingual word embedding space where words from two languages are represented in the new space so that similar words, which may be in different languages, have similar representations. Such bilingual word embeddings have been used in a number of tasks including semantic word similarity (Faruqui and Dyer, 2014; Ammar et al., 2016) learning bilingual word lexicons (Mikolov et al., 2013; Gouws et al., 2015; Vulic and Korhonen, 2016), parsing (Guo et al., 2015; Täckström et al., 2012), information retrieval (Vulic and Moens, 2015), and cross-lingual document classification (Klementiev et al., 2012; Kočiskỳ et al., 2014).

Some bilingual word embedding methods such as (Blunsom and Hermann, 2014; Gouws et al., 2015) require sentence or word aligned data, which our approach does not require. We compare our approach to the bilingual embeddings produced by the recent method of (Ammar et al., 2016). Like our approach, this work does not require availability of parallel corpora but only a seed dictionary.

On the aspect of enriching word embeddings with linguistic knowledge for the purpose of machine translation, Sennrich and Barry (Sennrich and Haddow, 2016) introduce linguistic features in sequence to sequence neural machine translation. Like our work, they also represent such features in the embedding layer. In addition to part-of-speech tags and morphological features, they also use syntactic dependency labels which are not applicable to our model since we work at the word level while their model is at the sentence level.

Knowledge Distillation. Knowledge distillation was introduced for model compression to learn small models from larger models (Bucilu et al., 2006; Hinton et al., 2015). For example, from a large neural network model a smaller model can be distilled such that it generalizes in the same way as the large model (Romero et al., 2014). Knowledge distillation was also used by (Hu et al., 2016) to distill knowledge from logical rules in the tasks of named entity recognition and sentiment analysis, thereby enforcing constraints on the trained model. Our approach is different from this prior work on knowledge distillation in that we distill knowledge from mapping functions of related languages into mapping functions of languages with only small seed dictionaries.

Domain adaptation, for which there is a long history, is also related to our work (Ben-David et al., 2007; Daumé III, 2007; Pan et al., 2010; Long and Wang, 2015). (Daumé III, 2007) proposed feature augmentation, suggesting that a model should have features that are general across domains, as well as features that are domain-specific. Thus the model learns from all domains while preserving domain-specific information. These kinds of models have to be retrained when a new domain is added. Our work however only has to train mapping functions that involve a new language, all others can be distilled without retraining them.

3 Embedding Linguistic Information

Since our approach relies on the quality of monolingual word embeddings, we would like to work with high quality word embeddings. We therefore, first seek to enhance the vector representations of words in the source and target languages so that they can capture useful linguistic information. The intuition is that such information can help narrow down the words in the target language that are considered valid translations for a given source language word. To that end, we model both part of speech (POS) tag distributions of words and subword information in the vector representations.

3.1 Part of Speech Distributions

The idea behind modeling POS tags is that words should have the same part of speech tag in different languages. For example, if we are translating the noun *Katze* from German to English, in English we expect the singular noun *cat* and not the plural *cats*. While this information may be monolithically represented in word vectors generated by embedding methods such as Skip-gram and CBOW, here we seek to explicitly model POS tags. Since each word can have multiple POS tags, we model a word's part of speech information as a distribution over all the possible POS tags that it can take on. We learn POS tag statistics by first tagging a large corpus of each language, we then use tag counts to generate distributions. For example, if the English word, *bark* appears tagged as a verb 30 times in our corpus, and tagged as a noun 10 times, we generate a vector which puts 2/3 in the verb direction, and 1/3 in the noun direction, and 0 in the directions of all other POS tags. While these statistics can be noisy, we hope they can still provide useful signals. We use the universal POS tags, there are 12 tags in the universal POS tags (Petrov et al., 2011).

For a given word w, we compute a vector representation $w_i \in \mathbb{R}^d$ using a word embedding method. For now, let's assume we use the Skip-gram model. In the next section, we describe an enhanced word embedding method. We compute a POS corpus statistics vector $v_i \in \mathbb{R}^{12}$ for the word using the 12 universal POS tags. With this new information, the representation for word i is given by

$$x_i = (w_i, v_i) \in \mathbb{R}^{d+12}\,. \qquad (1)$$

3.2 Word Internal Structure

Morphology carries information that is useful for capturing the identity of a word. It represents information such as tense. When doing cross-lingual zero-shot projection of a word in a source language, we wish to translate to words that have the same linguistic properties. For example, the German word *gewinnen* should be translated to the present tense win, not the past tense won which in German is *gewonnen*. We approximate morphology by incorporating sub-word information into the vector representations. There are several ways of doing this, one approach is to work on the level of characters. We go for the middle-ground, in which a word is represented as a combination of a vector for the word itself with vectors of sub-word units that comprise it. In particular, for a given a word we learn a vector representation for the word itself, and also for each n-gram of $>= 3$ and < 6 in the word (Bojanowski et al., 2017). Each word is thus represented by the sum of the vector representations of its n-grams, including the word itself. This representation is then used to replace w_i in equation 1.

4 Training Objective

A common objective function used in prior work (Mikolov et al., 2013; Dinu et al., 2014) for learning cross vector space mapping functions is the regularized least squares error:

$$\hat{\mathbf{W}} = \underset{\mathbf{W} \in \mathbb{R}^{s \times t}}{\arg\min} ||\mathbf{XW} - \mathbf{Y}||_F + \lambda ||\mathbf{W}|| \quad (2)$$

where matrix $\hat{\mathbf{W}}$ is the learned mapping function, $\mathbf{X}$ and $\mathbf{Y}$ represent the matrices containing the vectors for the source language words and vectors for the target language words, respectively. Instead of the least squares loss shown in equation 2, we use a ranking loss, as in (Lazaridou et al., 2015), which aims to rank correct training data pairs (x_i, y_i) higher than incorrect pairs (x_i, y_j) with a margin of at least γ. The margin γ is a hyper-parameter which is application specific, and the incorrect labels, y_j can be selected randomly such that $j \neq i$ or in a more application specific manner[2].

[2] In our experiments, we explored several application specific approaches for choosing negative examples, including one that picks negative examples among words whose part of speech class is different from the positive example. However, these approaches did produce significant improvement, and we resorted back to randomly selected negative examples.

Given a seed dictionary training data of the form $D^{tr} = \{x_i, y_i\}_{i=1}^{m}$, the margin-based ranking loss is defined as:

$$J_{\text{single}} = \sum_{i=1}^{m} \sum_{j \neq i}^{k} \max\Big(0, \gamma + d(y_i, \hat{y_i}) - d(y_j, \hat{y_i})\Big) \quad (3)$$

where $\hat{y_i} = \mathbf{W}x_i$ is the prediction, k is the number of incorrect examples per training instance, and $d(x, y) = (x - y)^2$ is the distance measure.

For a given correct pair and incorrect pair, substituting $\hat{y_i} = \mathbf{W}x_i$. The loss is given by:

$$\max\Big(0, \gamma + (y_i - \hat{y_i})^2 - (y_j - \hat{y_i})^2\Big) : j \neq i. \quad (4)$$

To evaluate the derivative analytically, we can write:

$$\begin{aligned} &\max\left(0, \gamma + (y_i - \hat{y_i})^2 - (y_j - \hat{y_i})^2\right) \\ &= \theta\left(\gamma + (y_i - \hat{y_i})^2 - (y_j - \hat{y_i})^2\right) \times \\ &\quad \left[\gamma + (y_i - \hat{y_i})^2 - (y_j - \hat{y_i})^2\right] \end{aligned} \quad (5)$$

where $\theta(x)$ denotes the Heaviside θ-function. The derivative with respect to the elements of the matrix $\mathbf{W}$ is then approximated by, after neglecting a term that would only contribute if the difference $(y_j - \hat{y_i})^2 - (y_i - \hat{y_i})^2$ were exactly γ

$$\begin{aligned} &\frac{\partial}{\partial W_{ab}}(\theta\left(\gamma + (y_i - \hat{y_i})^2 - (y_j - \hat{y_i})^2\right) \times \\ &\quad \left[\gamma + (y_i - \hat{y_i})^2 - (y_j - \hat{y_i})^2\right] \\ &\simeq 2\theta\left(\gamma + (y_i - \hat{y_i})^2 - (y_j - \hat{y_i})^2\right) \times \\ &\quad x_{i\,b}\left(y_{j\,a} - y_{i\,a}\right) \end{aligned} \quad (6)$$

5 Model Distillation

In zero-shot learning for bilingual dictionary induction a large seed dictionary is used to train a mapping function. Such large training dictionaries might not be available for all languages. However, for a given language with only a small seed dictionary, there could be a highly related language with a much larger seed dictionary. We propose a method for leveraging mapping functions of nearby languages to train mapping functions for languages where large seed dictionaries may not be available. Our method is related to notion of having a bridge or pivot language as done in sentence level translation (Utiyama and Isahara, 2007). We develop a distillation training objective that allows us to seamlessly leverage several bridge languages for word level translation.

5.1 Trilingual Paths for distillation

Let us consider the problem of translating from a given source language to English. As a running example, we use Portuguese(pt) as the source language. We wish to learn a mapping function from word vectors in Portuguese to word vectors in English. We can set up a learning task, using a training dataset $D = \{x_i, y_i\}_{i=1}^{m}$ and the loss defined in Equation 3. This gives us the projection function in the form of a matrix: $\mathbf{W}^{(pt \to en)}$. We can thus translate Portuguese words to English as follows:

$$\hat{y_i}^{(en) \leftarrow (pt)} = \mathbf{W}^{(pt \to en)} x_i^{(pt)} \tag{7}$$

If the seed dictionary for Portuguese to English is small, $\mathbf{W}^{(pt \to en)}$ might generalize poorly, producing many wrong translations when using Equation 7. Suppose, a related language, for example, Spanish has a lot of training data available, and we have independently trained its mapping function, which can make predictions from Spanish to English as follows:

$$\hat{y_i}^{(en) \leftarrow (es)} = \mathbf{W}^{(es \to en)} x_i^{(es)} \tag{8}$$

Since $\mathbf{W}^{(es \to en)}$ is trained with a lot of data, we expect it to generalize better and make more accurate predictions than $\mathbf{W}^{(pt \to en)}$. One insight here is that since the languages es and pt are highly related, we need much less data to train an accurate mapping matrix $\mathbf{W}^{(pt \to es)}$ than we to need to learn an accurate $\mathbf{W}^{(pt \to en)}$. Therefore we train a mapping function from Portuguese to Spanish, which makes predictions as follows.

$$\hat{y_i}^{(es) \leftarrow (pt)} = \mathbf{W}^{(pt \to es)} x_i^{(pt)} \tag{9}$$

We now have a second path that goes from Portuguese to English much like Equation 7 but this path goes via Spanish as follows:

$$\begin{aligned} \hat{y_i}^{(es) \leftarrow (pt)} &= \mathbf{W}^{(pt \to es)} x_i^{(pt)} \\ \hat{y_i}^{(en) \leftarrow (es) \leftarrow (pt)} &= \mathbf{W}^{(es \to en)} \hat{y_i}^{(es) \leftarrow (pt)} \end{aligned} \tag{10}$$

Figure 2 illustrates the two paths from Portuguese to English. Our main insight is to use knowledge distillation, to improve the accuracy of the mapping matrix $\mathbf{W}^{(pt \to en)}$ through $\hat{y_i}^{(en) \leftarrow (es) \leftarrow (pt)}$. This distillation is done by modifying our learning objective.

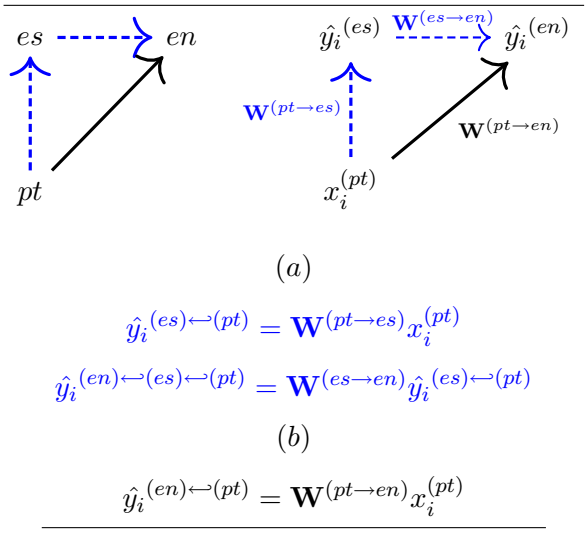

Figure 2: Translating with both a trilingual path (dotted lines, and equation (a)) , and a bilingual path (solid line, and equation (b))

5.2 Distillation Objective

For a given Portuguese word $x_i^{(pt)}$, Equation 7 makes the prediction $\hat{y_i}^{(en) \leftarrow (pt)}$ and Equation 10 makes the trilingual prediction $\hat{y_i}^{(en) \leftarrow (es) \leftarrow (pt)}$ which involves three languages. We would like to improve predictions made by Equation 7 by improving the mapping matrix $\mathbf{W}^{(en \to pt)}$. Therefore when training using the Portuguese to English training data, we want our objective to both minimize the loss defined in Equation 3 and simultaneously to let $\mathbf{W}^{(en \to pt)}$ mimic predictions made through the path $\hat{y_i}^{(en) \leftarrow (es) \leftarrow (pt)}$ as "soft targets" within a margin. The distillation objective is as follows:

$$J_d = \sum_{i=1}^{m} \max\Big(0, \Big(\hat{y_i}^{(en) \leftarrow (pt)} - \hat{y_i}^{(en) \leftarrow (es) \leftarrow (pt)}\Big)^2 - \phi\Big), \tag{11}$$

where ϕ is the margin. We combine J_{single} and J_d through a weighted average of the two different objective functions. Notice that J_d can be computed without having labeled training data. In our experiments, we show that even in this case of unlabeled data, which gets rid of J_{single} since it requires labeled data, J_d outperforms models trained using only J_{single} when the training data is small.

5.3 Multiple Trilingual Paths

We are not restricted to distilling Portuguese through Spanish only. Our model can, in addition,

for example distill through German, French, and other languages. We can modify the distillation loss as follows:

$$J_{d-multi} = \sum_{i=1}^{m} \sum_{j=1}^{n} \psi_j \max \Big(0, \big(\hat{y_i}^{(en)\leftrightarrow(pt)} - \hat{y_i}^{(en)\leftrightarrow(j)\leftrightarrow(pt)}\big)^2 - \phi\Big), \quad (12)$$

where j labels the distillation language. With the objective $J_{d-multi}$ combined with J_{single}, we are training a mapping function which mimics the behavior of many trilingual paths, as "soft targets" within a margin ψ. We keep ϕ the same in our experiments across all trilingual paths. The ψ_i are weights that reflect how much we penalize our model if it diverges from the predictions of a particular trilingual path. Intuitively, if a language is similar to our source language, (pt) in this case, its corresponding ψ value should be high. For example, if Spanish is considered more related to Portuguese than any other language in the trilingual paths in Equation 12, than we expect $\forall i \neq 1, \psi_1 > \phi_i$. This is assuming that the second parts of the trilingual paths have similar accuracies, ie. $\mathbf{W}^{(es\rightarrow en)}$, $\mathbf{W}^{(fr\rightarrow en)}$, and $\mathbf{W}^{(...\rightarrow en)}$ have similar projection accuracies. The most similar language is expected to be the easiest to project into from Portuguese. For example we might expect $\mathbf{W}^{(pt\rightarrow es)}$ to be more accurate than $\mathbf{W}^{(pt\rightarrow de)}$, if we have similar amounts of training data for learning both of these. We next present how we learn the ψ_i values for the multiple paths.

5.4 Weighted Trilingual Paths

Going back to the example, we first learn the weights using the Portuguese to English training set, $D = \{x_i^{pt}, y_i^{en}\}_{i=1}^{m}$, and then input the weights into the model before training with $J_{d-multi}$ and J_{single}. Suppose we want to compute ψ_1 which corresponds to Spanish in Equation 12. For a given Portuguese word $x_i^{pt} \in D$, whose English translation is y_i^{en}, we can compute:

$$\psi_{1i}^{dot} = (y_i^{en})^T \hat{y_i}^{(en)\leftrightarrow(es)\leftrightarrow(pt)} \quad (13)$$

We also experimented with a bilinear term:

$$\psi_{1i}^{bilinear} = (y_i^{en})^T H \hat{y_i}^{(en)\leftrightarrow(es)\leftrightarrow(pt)} \quad (14)$$

We found a better performing approach to be:

$$V = \Big(\hat{y_i}^{(en)\leftrightarrow(pt)} - \hat{y_i}^{(en)\leftrightarrow(es)\leftrightarrow(pt)}\Big)^2$$
$$\psi_{1i}^{euclid} = \exp(\frac{1}{V}). \quad (15)$$

	P@1	**P@5**	**P@10**
	Italian ($en \rightarrow it$)		
THIS	51.0	66.6	72.4
THIS w/pos	**51.6**	**68.5**	**73.4**
Ridge	29.7	44.2	49.1
Lazaridou et. al	40.2	54.2	60.4
MultiCluster	2.40	7.30	11.0
MultiCCA	0	0.1	0.3

Table 1: Translation accuracy on the English to Italian dataset of (Dinu et al., 2014).

6 Experimental Evaluation

In this section, we study the following questions:

- What is the effect of modeling linguistic information in the vector representations of the source and target languages on accuracy of bilingual dictionary induction?
- Can our knowledge distillation objective from trilingual paths involving related languages improve accuracy of mapping functions of languages with small seed dictionaries?

6.1 Data and Experimental Setup

In most of our experiments, we use the training data that was used to train the multi-lingual embeddings in (Ammar et al., 2016). We indicate when this is not the training data used. This data was obtained automatically by using Google Translate. For test data, we use manual translations either from prior work or from searching the Web, including genealogical word lists [3].

For word vector representations, we use Wikipedia to train 300 dimensional vectors for all languages we evaluate on. Based on a validation set, we set the margin γ in Equation 3 through Equation 6 to be $\gamma = 0.4$, ϕ in Equations 11, 12, and 15 to be $\phi = 0.01$. We estimate model parameters using stochastic gradient descent.

6.2 Methods Under Comparison

In our experiments, we compare performance of the following methods.

- The method **THIS** refers to our model which uses a max-margin loss function as defined

[3] For example: https://familysearch.org/wiki/en/Afrikaans_Word_List

in Equation 3. It uses sub-word information in the vector representations of words. One variation of our method is, **THIS w/pos**, which includes POS tag statistics as additional dimensions. The distilled variations of our method explicitly indicate the languages involved, for example $(pt \rightarrow es \rightarrow en)$.

- The method **Ridge** is used in a number of prior work (Mikolov et al., 2013; Dinu et al., 2014; Vulic and Korhonen, 2016). These approaches use an L2-regularized least-squares error objective as shown in Equation 2.

- The method **Lazaridou et al.** was proposed by (Lazaridou et al., 2015). It uses a max-margin ranking function and introduces a way of picking negative examples in computing the loss.

- The methods **MultiCluster** and **MultiCCA** refer to the multilingual word embeddings introduced by (Ammar et al., 2016). They extend canonical correlation analysis (CCA) based methods (Haghighi et al., 2008; Faruqui and Dyer, 2014) to a multi-lingual setting where they treat English as the common vector space. For these methods, we use their pre-trained word embeddings.

6.3 Linguistic Information Evaluation

To address the first of our evaluation questions, we performed experiments on the dataset introduced by (Dinu et al., 2014), where the state-of-the art is the work of (Lazaridou et al., 2015). This is an Italian to English dataset, which consists of $5K$ translation pairs as training data, and $1.5K$ pairs as test data. In both (Dinu et al., 2014) and (Lazaridou et al., 2015), the embeddings were trained on Wikipedia and additional corpora, we only train on Wikipedia.

The results for this experiment are shown in Table 1. Our method, THIS, performs well above the previous state of the art (Lazaridou et al., 2015). For top-1 precision, as can been seen in Table 1, we obtained an 11% gain. From Table 1, we can also see that the POS statistics are only marginally helpful. The word embeddings generated with MultiCluster and MultiCCA perform poorly, with MultiCluster doing better than MultiCCA.

We additionally carried out experiments on 8 other language pairs, further showing our method outperforming prior work. The results are shown in Table 3, and the corresponding data is shown in Table 2. For these language pairs, we do not show results for our method, THIS w/pos, since POS taggers are not available for some of the languages. We also do not show (Lazaridou et al., 2015), as they did not do experiments on these data sets, and we did not have an implementation of their approach. Additionally, (Ammar et al., 2016) did not have trained embeddings for Dutch (nl).

	$de \downarrow en$	$es \downarrow en$	$fr \downarrow en$	$it \downarrow en$	$nl \downarrow en$	$sv \downarrow en$
Train	400k*	400k*	100k*	5k	1,392	110k*
Test	1,180	1,109	810	2,148	296	471

Table 2: Training and test sets for various language pairs. The training datasets marked with (*) are from (Ammar et al., 2016) obtained through Google Translate. Italian to English is from (Dinu et al., 2014). The Dutch to English training dataset is introduced in this paper. With the exception of Italian to English, all test datasets are introduced in this paper.

	$de \downarrow en$	$es \downarrow en$	$fr \downarrow en$	$it \downarrow en$	$nl \downarrow en$	$sv \downarrow en$
	P@10					
THIS	**57.8**	**59.5**	**67.4**	**70.0**	**60.8**	**54.6**
Ridge	32.8	54.2	59.9	66.4	58.8	44.6
MultiCluster	12.2	8.1	4.6	6.9	-	9.0
MultiCCA	6.7	4.3	2.9	5.6	-	10.1

Table 3: Top-10 precision for eight languages translated to English. The high accuracy on Italian can be explained by the fact that, unlike other language pairs, for Italian we do not use Google Translate training data, but the data of (Dinu et al., 2014), as shown in Table 2.

6.4 Trilingual Paths for Distillation

To address our second evaluation question, we carried out experiments with languages for which we only had small seed dictionaries. The training and test datasets for this setup are shown in Table 4. We gathered these datasets by searching for manually created datasets. In the cases were we could not find any, we used Google Translate, which, however produces some noisy translations. This is partly due to the fact that the translations are done out of context.

We begin with thorough experiments on the

	pt $\downarrow$ en	pt $\downarrow$ es	pt $\downarrow$ fr	pt $\downarrow$ de	da $\downarrow$ en	da $\downarrow$ sv	af $\downarrow$ en	af $\downarrow$ nl
Train	573	701	1,808	465	3,000	1,980	3,744	2,000
Test	296	0	0	0	262	0	459	0

Table 4: Training and test datasets used in the trilingual path distillation experiments. We evaluated sub-parts of trilingual paths such as $pt \rightarrow es$, and $pt \rightarrow fr$ using cross validation hence the test sets for those languages are zero.

		P@10
Portuguese ($pt \rightarrow en$)		
1	THIS ($pt \rightarrow en$)	65.2
2	($pt \rightarrow en$) +($pt \rightarrow es \rightarrow en$) [unlabeled data]	74.0
3	($pt \rightarrow en$) + ($pt \rightarrow es \rightarrow en$)	**82.1**
4	($pt \rightarrow en$) + ($pt \rightarrow \begin{pmatrix} de \\ es \\ fr \end{pmatrix} \rightarrow en$) [Weighted]	81.8
5	($pt \rightarrow en$) + ($pt \rightarrow \begin{pmatrix} de \\ es \\ fr \end{pmatrix} \rightarrow en$)[Unweighted]	78.4
6	Ridge	60.8

Table 5: Trilingual path distillation results for Portuguese to English.

Portuguese-English language pair. The results are shown in Table 5. First, we see that if we distill through the Spanish trilingual path ($pt \rightarrow es \rightarrow en$), without using any labeled data from $pt \rightarrow en$, we already obtain a 9% gain in accuracy, line 2 in Table 5. If, in addition to distilling through Spanish, we use the available training data $pt \rightarrow en$, 573 translation pairs, line 3 in Table 5, we obtain a 17% gain in accuracy. We see however that adding the distillation paths via French, and German did not improve performance, line 4 in Table 5. This can be attributed to the fact that with multiple distillation paths, the model has to optimize a more difficult function. On the other hand, we see that our trilingual weighting mechanism is effective. Without path weights, top-10 accuracy is 78.4% vs 81.8% with weights, lines 4 and 5 in Table 5. The learned weights for the three languages involved in the trilingual paths for Portuguese are shown in Figure 3. Spanish is the highest weighted, followed by French, and German has the lowest weight. By definition, the learned weights add up to 1. In Figure 4, we show accuracy while varying the size of the seed dictionary. We can see that, given the small size of the training data, distillation provides a strong advantage.

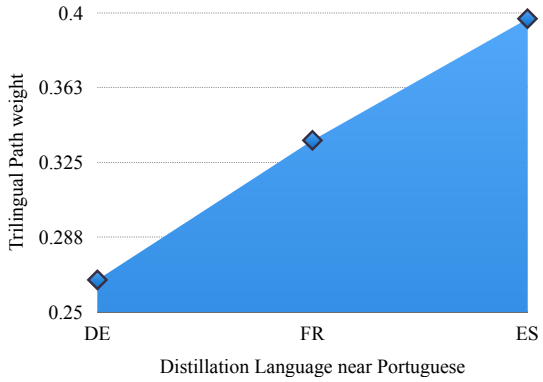

Figure 3: Learned weights for languages involved in trilingual paths for translating Portuguese to English. Spanish is the highest weighted and German is the lowest.

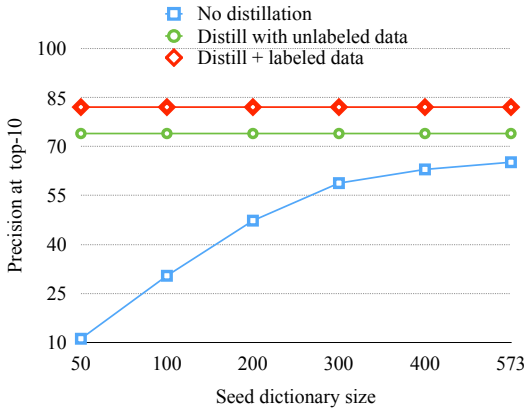

Figure 4: Varying the size of the seed dictionary for ($pt \rightarrow en$).

Finally, we applied our distillation method to Afrikaans and Danish. Afrikaans distills from Dutch, and Danish distills from Swedish. As shown in Table 6, in both cases, we obtained performance gains. However, in both of these cases, performance gains are modest. Unlike Portuguese to English, the seed dictionaries involved in training these language pairs were obtained automatically using Google Translate and contain noisy translations.

7 Conclusion

We have presented a knowledge distillation training objective that leverages trilingual paths of related languages to improve mapping functions of languages with small seed dictionaries. The model produces substantial gains in accuracy for several language pairs.

There are several future directions. First, due

	P@10
Afrikaans ($af \rightarrow en$)	
THIS ($af \rightarrow en$)	46.4
($af \rightarrow en$)+ ($af \rightarrow nl \rightarrow en$) [unlabeled data]	49.9
($af \rightarrow en$)+($af \rightarrow nl \rightarrow en$)	51.0
Ridge	38.6
Danish ($da \rightarrow en$)	
THIS ($da \rightarrow en$)	44.4
($da \rightarrow en$) + ($da \rightarrow sv \rightarrow en$) [unlabeled data]	45.2
($da \rightarrow en$) + ($da \rightarrow sv \rightarrow en$)	47.2
Ridge	37.1

Table 6: Trilingual path distillation results for Afrikaans and Danish.

to advances in methods for extracting general purpose knowledge (Mitchell et al., 2015; Nakashole et al., 2013; Wijaya et al., 2014), the use of semantic knowledge has been explored for several natural language tasks (Nakashole and Mitchell, 2015; Yang and Mitchell, 2017). However, for bilingual dictionary induction, and more generally, machine translation, the role of semantic knowledge has not been fully explored. We consider this to be a promising line of future work. Second, although we focus on bilingual dictionary induction, our knowledge distillation training objective that enables seamless use of paths of rich resource languages as teachers of low resource languages is general and can be applied to problems such as multilingual tagging and parsing.

References

Waleed Ammar, George Mulcaire, Yulia Tsvetkov, Guillaume Lample, Chris Dyer, and Noah A. Smith. 2016. Massively multilingual word embeddings. *CoRR*, abs/1602.01925.

Shai Ben-David, John Blitzer, Koby Crammer, Fernando Pereira, et al. 2007. Analysis of representations for domain adaptation. *Advances in neural information processing systems*, 19:137.

Phil Blunsom and Karl Moritz Hermann. 2014. Multilingual distributed representations without word alignment.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *TACL*.

Cristian Bucilu, Rich Caruana, and Alexandru Niculescu-Mizil. 2006. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 535–541. ACM.

Hal Daumé III. 2007. Frustratingly easy domain adaptation. *ACL 2007*, page 256.

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

Manaal Faruqui and Chris Dyer. 2014. Improving vector space word representations using multilingual correlation. In *EACL*, pages 462–471.

Stephan Gouws, Yoshua Bengio, and Greg Corrado. 2015. Bilbowa: Fast bilingual distributed representations without word alignments. In *IICML*, pages 748–756.

Jiang Guo, Wanxiang Che, David Yarowsky, Haifeng Wang, and Ting Liu. 2015. Cross-lingual dependency parsing based on distributed representations. In *ACL*, pages 1234–1244.

Aria Haghighi, Percy Liang, Taylor Berg-Kirkpatrick, and Dan Klein. 2008. Learning bilingual lexicons from monolingual corpora. In *ACL*, pages 771–779.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Zhiting Hu, Xuezhe Ma, Zhengzhong Liu, Eduard Hovy, and Eric Xing. 2016. Harnessing deep neural networks with logic rules.

Alexandre Klementiev, Ivan Titov, and Binod Bhattarai. 2012. Inducing crosslingual distributed representations of words. In *COLING*, pages 1459–1474.

Tomáš Kočiskỳ, Karl Moritz Hermann, and Phil Blunsom. 2014. Learning bilingual word representations by marginalizing alignments. *arXiv preprint arXiv:1405.0947*.

Angeliki Lazaridou, Georgiana Dinu, and Marco Baroni. 2015. Hubness and pollution: Delving into cross-space mapping for zero-shot learning. In *ACL*, pages 270–280.

Mingsheng Long and Jianmin Wang. 2015. Learning multiple tasks with deep relationship networks. *arXiv preprint arXiv:1506.02117*.

Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. 2013. Exploiting similarities among languages for machine translation. *CoRR*, abs/1309.4168.

Tom M. Mitchell, William W. Cohen, Estevam R. Hruschka Jr., Partha Pratim Talukdar, Justin Betteridge, Andrew Carlson, Bhavana Dalvi Mishra, Matthew Gardner, Bryan Kisiel, Jayant Krishnamurthy, Ni Lao, Kathryn Mazaitis, Thahir Mohamed, Ndapandula Nakashole, Emmanouil Antonios Platanios, Alan Ritter, Mehdi Samadi, Burr Settles, Richard C. Wang, Derry Tanti Wijaya, Abhinav Gupta, Xinlei Chen, Abulhair Saparov, Malcolm Greaves, and Joel Welling. 2015. Never-ending

learning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, (AAAI)*, pages 2302–2310.

Tom M Mitchell, Svetlana V Shinkareva, Andrew Carlson, Kai-Min Chang, Vicente L Malave, Robert A Mason, and Marcel Adam Just. 2008. Predicting human brain activity associated with the meanings of nouns. *science*, 320(5880):1191–1195.

Ndapandula Nakashole and Tom M. Mitchell. 2015. A knowledge-intensive model for prepositional phrase attachment. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics, (ACL)*, pages 365–375.

Ndapandula Nakashole, Gerhard Weikum, and Fabian Suchanek. 2013. Discovering semantic relations from the web and organizing them with patty. *ACM SIGMOD Record*, 42(2):29–34.

Sinno Jialin Pan, Xiaochuan Ni, Jian-Tao Sun, Qiang Yang, and Zheng Chen. 2010. Cross-domain sentiment classification via spectral feature alignment. In *Proceedings of the 19th international conference on World wide web*, pages 751–760. ACM.

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2011. A universal part-of-speech tagset. *arXiv preprint arXiv:1104.2086*.

Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. 2014. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*.

Rico Sennrich and Barry Haddow. 2016. Linguistic input features improve neural machine translation. In *Proceedings of the First Conference on Machine Translation*, pages 83–91, Berlin, Germany. Association for Computational Linguistics.

Oscar Täckström, Ryan T. McDonald, and Jakob Uszkoreit. 2012. Cross-lingual word clusters for direct transfer of linguistic structure. In *NAACL*, pages 477–487.

Masao Utiyama and Hitoshi Isahara. 2007. A comparison of pivot methods for phrase-based statistical machine translation. In *HLT-NAACL*, pages 484–491.

Ivan Vulic and Anna Korhonen. 2016. On the role of seed lexicons in learning bilingual word embeddings. ACL.

Ivan Vulic and Marie-Francine Moens. 2015. Bilingual word embeddings from non-parallel document-aligned data applied to bilingual lexicon induction. In *ACL*, pages 719–725.

Derry Tanti Wijaya, Ndapandula Nakashole, and Tom M Mitchell. 2014. Ctps: Contextual temporal profiles for time scoping facts using state change detection. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Bishan Yang and Tom M. Mitchell. 2017. Leveraging knowledge bases in lstms for improving machine reading. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, (ACL)*.