BILBOWA: Fast Bilingual Distributed Representations without Word Alignments

Stephan Gouws[*]
Stellenbosch University

Yoshua Bengio
CIFAR Fellow
University of Montréal

Greg Corrado
Google Inc.
Mountain View, CA.

Abstract

We introduce BilBOWA ("Bilingual Bag-of-Words without Alignments"), a simple and computationally-efficient model for learning bilingual distributed representations of words which can scale to large datasets and does not require word-aligned training data. Instead it trains directly on monolingual data and extracts a bilingual signal from a smaller set of raw text sentence-aligned data. This is achieved using a novel sampled bag-of-words cross-lingual objective, which is used to regularize two noise-contrastive language models for efficient cross-lingual feature learning. We show that bilingual embeddings learned using the proposed model outperforms state-of-the-art methods on a cross-lingual document classification task as well as a lexical translation task on the WMT11 data. Our code will be made available as part of the open-source `word2vec` toolkit.

1 Introduction

Raw text data is freely available in many languages, yet labeled data – e.g. text marked up with parts-of-speech or named-entities – is expensive and mostly available for English. Although several techniques exist that can learn to map hand-crafted features from one domain to another [4, 7, 18], it is in general non-trivial to come up with good features which generalize well across tasks, and even harder across different languages. It is therefore very desirable to have unsupervised techniques which can learn useful syntactic and semantic features that are invariant to the tasks or languages that we are interested in. Unsupervised *distributed* representations of words capture important syntactic and semantic information about languages and these techniques have been succesfully applied to a wide range of tasks [6, 19], across many different languages [1]. Inducing these representations usually involves training a neural network language model (NLM) [2], where words are represented as learned, real-valued feature vectors referred to as *word embeddings*. These models have the property that similar embedding vectors are learned for similar words during training. This improves generalization when the embedding vectors are used as features on word- and sentence-level prediction tasks.

Distributed representations can also be induced *over different language-pairs* and can serve as an effective way of learning linguistic regularities which generalize across languages, in that words with similar syntactic and semantic properties are represented using similar vectorial representations (i.e. embed nearby in the embedded space). This is especially useful for transferring the limited label information from high-resource to low-resource languages, and has been demonstrated to be effective for document classification [12], outperforming a strong machine-translation baseline; as well as named-entity recognition and machine translation [21, 15]. However, one significant drawback of these methods is the **high computational cost** of inducing the embeddings. As a first step, current methods [12, 21] rely on performing a word-alignment step over sentence-aligned data to obtain the word translation-pair co-occurrence frequencies *prior to training* the models. In a second step,

[*]Part of this research was conducted while the first author was an intern at Google in Mountain View, CA.

two NLMs are trained jointly, with an added regularization term which penalizes large distances between the learned embeddings of high-frequency translation pairs. Inducing word alignments is a research field of its own, and in general a costly step. Moreover, monolingual NLMs are traditionally criticized for their slow training speeds, with each training update scaling linearly in the vocabulary-length, which can easily range in the tens or hundreds of thousands. Although faster methods exist for training NLMs [3, 6, 16], evaluating the cross-lingual regularization term (which is based on noisy word-alignments and in the worst case considers the $O(V^2)$ possible word translation-pairs) is still very expensive. Hence, *previous work report training times on the order of weeks* [12, 21].

We are motivated by the good results obtained using these methods, but we believe that the prohibitively expensive training times limit the large-scale application of current approaches, and hence in this paper we attack these high training costs head-on. We introduce **BilBOWA** (*Bilingual Bag-of-Words without Word Alignments*), a family of simple, efficient algorithms for inducing bilingual word embeddings (with a trivial extension to multilingual embeddings). BilBOWA **does not require word-level alignments, but instead extracts a bilingual signal directly from a limited sample of raw-text, sentence-aligned, parallel data** (e.g. Europarl) which it uses to align embeddings as they are learned over monolingual training data. The only requirement is monolingual and parallel training data in raw text form. Our contributions are:

- We introduce a novel, computationally-efficient **sampled cross-lingual objective** which only considers sampled bag-of-words sentence-aligned data at each training step, thereby avoiding the need for estimating word alignments;
- we experimentally evaluate the induced cross-lingual embeddings on a document-classification and lexical translation task, where we show that our method outperforms state-of-the-art methods, with training time reduced to minutes or hours compared to several days for prior approaches;
- finally, we make available our efficient C-implementation as part of the open-source `word2vec` toolkit[1] to hopefully stimulate further research on cross-lingual distributed feature learning.

2 Approaches to Learning Cross-lingual Word Embeddings

In the cross-lingual learning setup, the goal is to learn features which generalize well *across different languages*. For instance, in the bilingual setup, we have two domains X_1 (e.g. English) and X_2 (e.g. French). Our goal is to learn features (embeddings) θ_i for each word or phrase or document $x_i^{(j)}$ in X_j, such that *similar units in each language* are assigned similar embeddings (the **monolingual objectives**), but additionally we also want *similar units across languages* to have similar representations (the **cross-lingual objective**). The latter property allows us to use the learned embeddings as features for training a classifier to predict labels in one language (e.g. topics, parts-of-speech, or named-entities) where we have labelled data, and then **directly transfer** it to a language for which we do not have much labelled data.

From an optimization perspective, there are several approaches to how one can optimize these two objectives. The simplest approach is to optimize each monolingual objective separately (i.e. train embeddings on each language separately), and then enforce the cross-lingual constraints as a separate, disjoint, 'alignment' step. The alignment step consists of learning a function for projecting the embeddings of words onto the embeddings of their translation pairs, obtained from a dictionary. This was shown to be a viable approach by Mikolov et al. [15] who learned a linear projection from one embedding space to the other. It was extended by Faruqui et al. [9], who simultaneously projected source and target language embeddings into a joint space using canonical correlation analysis. The advantage of this approach is that it is very fast to learn the embedding alignments. The main conceptual criticism of this approach is that it is not clear that a single transformation (whether linear or nonlinear) can capture the relationships between all words in the source and target languages. Further, more practical, disadvantages are that it requires an accurate dictionary for the language-pair and considers only one translation per word, ignoring the rich multi-sense polysemy of natural languages. Since in this approach one is aligning the two embedding spaces as an independent, disjoint step, we refer to this approach as **offline alignment**.

[1] `https://code.google.com/p/word2vec/`

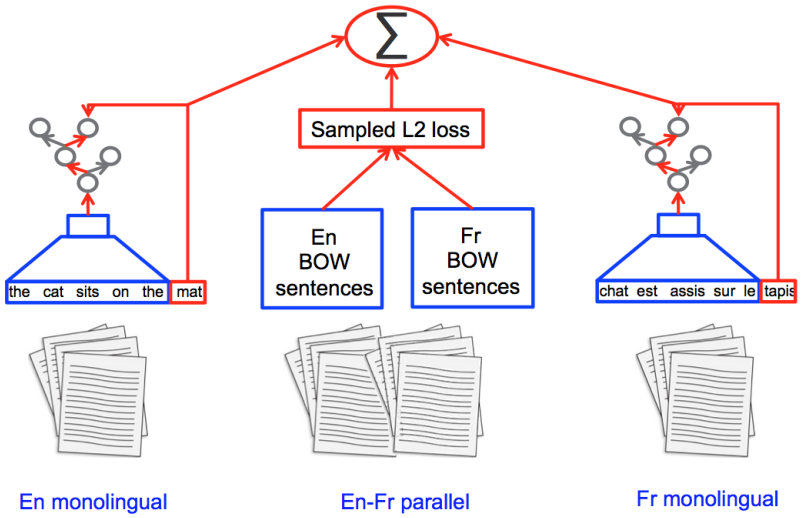


Figure 1: Schematic of the proposed BilBOWA model architecture for inducing bilingual word embeddings. Two monolingual skipgram models are jointly trained while enforcing a sampled L_2-loss which aligns the embeddings such that translation-pairs are assigned similar embeddings in the two languages (see text for more details).

A different approach is to leverage sentence-aligned parallel data and train a model to learn similar representations for the aligned sentences. This is the approach followed by Hermann et al. [11]. The advantage of this approach is that it is fast due to the noise-contrastive training criterion. The main drawbacks of this method are that it can only train on **limited parallel data**, which is expensive to obtain and not necessarily written in the same style or register as the domain where the features might be applied (i.e. there is a strong **domain bias**).

Another approach is to **jointly optimize** the monolingual objectives $J(\cdot)$, with the cross-lingual objectives enforced as a **cross-lingual regularizer**. To do this, we define a cross-lingual regularization term $\Omega(\cdot)$, and optimize everything jointly over the dataset $\mathcal{D}$, e.g. in the bilingual setting (see Figure 1 for a schematic):

$$J^{\text{TOTAL}} = \min_{\theta_1,\theta_2} \sum_{j\in\{1,2\}} \sum_{i\in\mathcal{D}} \underbrace{J_j(X_i^j;\theta_i)}_{\text{monolingual objective}} + \underbrace{\Omega(\theta_1,\theta_2)}_{\text{cross-lingual objective}} \quad . \tag{1}$$

This approach was shown to be useful by Klementiev et al. [12]. The advantages of this formulation are that it enables one to train on any available monolingual data, which is both more abundant and less biased than the parallel-only approach, since one can train on data which resembles the data you will be applying the learned features to. The disadvantage is that the original model of Klementiev et al. is extremely slow to train. The training complexity stems both from how they implement their monolingual and cross-lingual objectives. For the monolingual objective they train a standard neural language model for which the complexity of the output softmax layer grows with the output vocabulary size. Therefore, in order to evaluate their model they had to reduce the output vocabulary to only the 3000 most frequent words. The second reason for the slow training times is that their cross-lingual objective considers the interactions between all pairs of words between the source and target vocabulary *at each training step*, which scales as the product of the two vocabularies. This work addresses these two issues individually, and in the following section we discuss the cross-lingual component in more detail.

3 Joint-training Challenges

Consider the regularized cross-lingual objective presented in Eqn 1. This formulation captures the intuition that we want to learn representations which model their individual languages well (the first term) while the $\Omega(\cdot)$ regularizer encourages representations to be similar for words that are related

across the two languages. Conceptually, this regularizer consists of minimizing a distance function between the representations $\mathbf{r}_i$ learned for words w_i in the two domains, weighted by how similar they are, i.e.

$$\Omega(\theta_1, \theta_2) = \sum_{w_i \in V_1} \sum_{w_j \in V_2} \text{sim}(w_i, w_j) \cdot \text{distance}(\mathbf{r}_i, \mathbf{r}_j). \tag{2}$$

where we use θ to denote all model parameters and $\mathbf{r}_i$ to denote the embedding learned for word w_i. In other words, when this weighted sum (and hence its contribution to the total objective) is low, we can be sure that words across languages that are similar (i.e. high $\text{sim}(w_{ij})$) will be embedded nearby each other.

Specifically, in the bilingual setting, word similarities can be expressed as a matrix $\mathbf{A}$ where a_{ij} encodes the translation "score" of word i in one language (e.g. English, denoted e) with word j in the other (e.g. French, denoted f). If the K-dimensional word embedding row-vectors $\mathbf{r}_i$ are stacked to form a (V, K)-dimensional matrix $\mathbf{R}$, then we can express what we will refer to as the **exact cross-lingual objective** as follows:

$$\Omega_{\mathbf{A}}(\theta_1, \theta_2) = \sum_i \sum_j a_{ij} ||\mathbf{r}_i^e - \mathbf{r}_j^f||^2 \tag{3}$$

$$= (\mathbf{R}^e - \mathbf{R}^f)^\top \mathbf{A} (\mathbf{R}^e - \mathbf{R}^f). \tag{4}$$

where subscript $\mathbf{A}$ indicates that the alignments are fixed. $\mathbf{A}$ captures the relationships between all V_1 words in the one language with respect to all V_2 words in the other language, and is indeed also the source of the two main challenges in this formulation, namely:

1. how to derive or learn which words to pair as translation pairs (i.e. deriving/learning $\mathbf{A}$);
2. how to **efficiently** evaluate $\Omega(\cdot)$ during training, since naively evaluating it scales as the product of the two vocabulary sizes $O(V_1 V_2)$ *at each training step*.

In the following section we describe the model and how we address both the high cost of evaluating the monolingual objectives, as well as the high cost of evaluating the vocabulary interactions.

4 The Model

As we argue in Section 3, the primary challenges with existing bilingual embedding models are their **computational complexity** (due to an expensive softmax or an expensive regularization term, or both) and, perhaps even more importantly, the strong **domain bias** that is introduced by models that train only on parallel data such as Europarl [11]. In this section we introduce the BilBOWA model which addresses both theses issues (see Figure 1 for a schematic overview of the model). As an overview: First, we replace the standard softmax objective with a more efficient noise-contrastive objective, allowing monolingual training updates to scale independently of the vocabulary size. Second, we only consider sampled, bag-of-words sentence-aligned data for the cross-lingual objective. This avoids the need for estimating word alignments, but moreover, the computation of the regularization term reduces to only the words in the observed sample (compared to considering the $O(V^2)$ worst-case possible interactions at each training step in the naive case).

4.1 Learning Monolingual Features: A Bag-of-words Language Modelling Objective

Since we do not care about language modelling, but more about feature learning, an alternative to the softmax is to use a **noise-contrastive approach** to score valid, observed combinations of words against randomly sampled, unlikely combinations of words. This idea was introduced by Collobert and Weston [6] where they optimized a margin between the observed score and the noise scores. In their formulation, scores were computed on *sequences* of words, but in Mikolov et al. [15] this idea was taken one step further and successfully applied to *bag-of-word* representations of contexts in their continuous bag-of-words (CBOW) and skipgram models trained using the **negative sampling training objective**. Any of these objectives would yield comparable speedup and could be used in our architecture. In this work we opted for the skipgram model trained using negative sampling.

Specifically, for the English case (other languages follow symmetrically), let the **input word** embedding matrix be $\mathbf{R} \in \mathbb{R}^{(V,K)}$ for a V-dimensional vocabulary and K-dimensional embeddings.

As for all other NLMs, the model is trained on sampled word-context pairs (w_t, h) where w_t is an integer target word id, and h is a sequence of the context word ids. We denote by bold $\mathbf{h}$ the K-dimensional context-vector representation for a set of context words $w_i \in h$. Finally, let $\mathbf{x}$ denote the bag-of-words, one-hot, sparse vector representing the input context words, i.e. a V-length vector with ones at the observed context-word ids and zeros everywhere else. We can then write $\mathbf{h}$ as the bag-of-words sum of its individual word embedding row-vectors $\mathbf{r}_i = \mathbf{R}_{[i,:]}$,

$$\mathbf{h} = \sum_{w_i \in h} \mathbf{r}_{w_i} = \mathbf{R}^\top \mathbf{x}. \tag{5}$$

Now, assume at each training step we have the context-vector representation $\mathbf{h}$ and the associated K-dimensional target-word embedding $\mathbf{q_t}$. Note that the **output embeddings** $\mathbf{q}_i = \mathbf{Q}_{[i,:]}$ are distinct from the input embeddings $\mathbf{r}_i = \mathbf{R}_{[i,:]}$. One motivation for this is that it allows the model to learn that a word is not likely to be its own context, e.g. $P(\text{dog}|\text{dog})$ is unlikely, which it cannot represent if the context-word dog and the target word dog uses the same vector [10]. We then sample k 'negative' word embeddings $\mathbf{q}_n$ from the unigram distribution $P(w)$, and maximize the following **negative sampling** criterion over the dataset $\mathcal{D}$:

$$\max_\theta \mathop{\mathbb{E}}_{(h,w_t)\sim\mathcal{D}} \left[\log \sigma(\mathbf{h}^\top \mathbf{q}_{w_t}) + \sum_{i=1}^{k} \mathop{\mathbb{E}}_{w_i \sim P_n(w)} \log \sigma(-\mathbf{h}^\top \mathbf{q}_{w_i}) \right] \tag{6}$$

where $\sigma(z) = 1/(1+e^{-z})$ is the sigmoid function. By only summing over the k noise words during each training step, this procedure scales only in the number of noise words we sample per training case (typically 5 to 15) and not in the entire vocabulary V (typically tens to hundreds of thousands).

4.2 Learning Cross-lingual Features: The Ω term

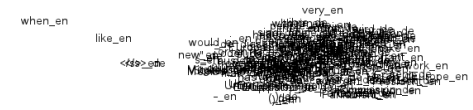
Besides learning how words in one language relate to each other (previous section), our main goal is to learn how words between the two languages relate to each other. Recall that the **exact cross-lingual objective** $\Omega_{\mathbf{A}}(\theta_1, \theta_2)$ (Eqn 3) penalizes the Euclidian distance between the two embedding spaces ($\mathbf{R}^e$ and $\mathbf{R}^f$) proportional to their alignment frequency. Previous work approached this step by performing a word-alignment step prior to training to learn the alignment matrix $\mathbf{A}$. However, performing this alignment step requires running Giza++ [17] or FastAlign [8] and training HMM word-alignment models. This is both computationally costly and also noisy. We would like to learn the translation correspondences without utilizing word alignments. In order to do that, we directly exploit the parallel training data. As a first step, notice that since the alignment weights can be made to sum to one, we can interpret the alignment weights as a distribution and write Eqn 3 as an expectation over the distribution of English and French word alignment probabilities $a_{ij} = P(w_i^e, w_j^f)$,

$$\Omega_{\mathbf{A}}(\theta_1, \theta_2) = \mathop{\mathbb{E}}_{(i,j)\sim P(w^e, w^f)} \left[||\mathbf{r}_i^e - \mathbf{r}_j^f||^2 \right] \tag{7}$$

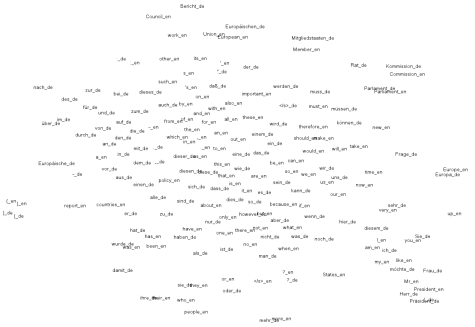
Since data is paired at the sentence level, we know that translation pairs for the *en* sentence occur in the *fr* sentence, but we do not know where. We therefore make a naive assumption and assume that each observed *en* word can potentially align with each observed *fr* word (i.e. we **assume a uniform word alignment model**), for each word *in the observed sentence pairs*. We leave it to future work to explore more advanced alignment models. Under this assumption, we can then approximate Eqn 7 by sampling S m-length English and n-length French sentence-pairs (s_e, s_f) from the parallel training data:

$$\Omega_{\mathbf{A}}(\theta_1, \theta_2) \approx \frac{1}{S} \sum_{(s_e, s_f)\in S} \frac{1}{mn} \sum_{i \in s_e} \sum_{j \in s_f} ||\mathbf{r}_i^e - \mathbf{r}_j^f||^2 \tag{8}$$

Notice that under a uniform alignment model, at time t each word in the sampled English sentence $s_e^{(t)}$ will be updated towards all words in the French sentence $s_f^{(t)}$. We can precompute this by simply updating each word towards the summed bag-of-words **sentence-vector** defined similarly to Eqn 5. Specifically, we set $S = 1$ and at training step t, we optimize the following **sampled, approximate**



(a) No parallel subsampling.



(b) With parallel subsampling.

Figure 2: A t-SNE visualization of the *same* 100 most frequent English and German words trained **(a)** *without* and **(b)** *with* parallel subsampling.

cross-lingual objective:

$$\Omega_{\mathbf{A}}^{(t)}(\theta_1, \theta_2) \triangleq ||\frac{1}{m} \sum_{w_i \in s_e^{(t)}}^{m} \mathbf{r}_i^e - \frac{1}{n} \sum_{w_j \in s_f^{(t)}}^{n} \mathbf{r}_j^f||^2 \tag{9}$$

where $s_*^{(t)}$ denotes the t'th sampled sentence-pair drawn from the parallel corpus. In other words, **the BilBOWA-loss minimizes a sampled L_2-loss between the bag-of-words sentence vectors of the parallel corpus**. On its own, this objective is degenerate since all embeddings would converge to the trivial solution (by collapsing all embeddings to the same value), but coupled as a regularizer with the monolingual losses, we find that it works well in practice. By sampling training sentences from the parallel document distribution, this objective efficiently approximates Eqn 3 (the more two words are observed together in a parallel sentence-pair, the stronger the embeddings for the two words will be pushed together, i.e. proportional to a_{ij}).

4.3 Subsampling for better results

Eqn 9 is an approximation of Eqn 3. We are really interested in estimating the global word-alignment statistics for a word-pair, i.e. a_{ij}. However, by sampling words at the sentence-level, the local alignment statistics are skewed by the words' *unigram frequencies of occurrence* in a given sentence (i.e. regardless of alignment). Hence in practice, we find that Eqn 9 *overregularizes the frequent words*. A simple solution to this is to **subsample words from the parallel sentences according to their unigram frequencies of occurrence**, effectively flattening the unigram distribution to a uniform distribution. In practice we found this to learn finer-grained cross-lingual embeddings for the frequent words, as illustrated in Figure 2.

5 Implementation and Training Details

We implemented our model in C as an extension of the popular open-source `word2vec` toolkit[2]. The implementation launches a monolingual skipgram model as a separate thread for each language, as well as a cross-lingual thread, and all threads access the shared embedding parameters asynchronously. For training the model, we make use of minibatch online asynchronous stochastic gradient descent (ASGD), where at time step t, parameter θ is updated as

$$\theta^{(t)} = \theta^{(t-1)} - \eta \frac{\partial J^{\text{TOTAL}}}{\partial \theta} \tag{10}$$

[2] `https://code.google.com/p/word2vec/`

Our initial implementation synchronized updates between threads, but we found that simply clipping individual updates to $[-0.1, 0.1]$ per thread was sufficient to ensure training stability and considerably improved training speed. For monolingual training data, we use the freely available, pretokenized Wikipedia datasets [1]. For cross-lingual training we use the freely-available Europarl v7 corpus [13]. Unlike the approach of Klementiev et al. [12] however, we do not need to perform a word-alignment step first. Instead our implementation trains directly on the raw parallel text files obtained after applying the standard preprocessing scripts that come with the data to tokenize, recase and remove all empty sentence-pairs. Embedding matrices were initialized by drawing from a zero mean, unit-variance gaussian distribution. The negative sampling objectives (Eqn 6) require us to sample k noise words per training pair from the unigram $P(w)$ *en* and *fr* distributions. A naive implementation of this step[3] can easily run in $O(V)$ per sampling step, negating the computational advantage of the noise-contrastive loss approach. However, there exist efficient algorithms for sampling from a multinomial in $O(1)$ with $O(V)$ setup cost [20], which is what we have used. [4] Doing one training update therefore amounts to selecting a context-target (h, w_t)-pair from each language and sampling k noise words for each according to their unigram distribution. We set k to the average number of words per sentence in the data, which was 25. Next we sample a random pair of parallel sentences from the parallel data. Finally, we make an update to all parameters according to Eqn 10, for which gradients are easy to compute due to the log-linear nature of the model.

6 Experiments

The BilBOWA model proposed in this paper enables one to learn cross-lingual distributed representations of words by exploiting only sentence-aligned training data. In this section we present experiments which evaluate the utility of the induced representations. We evaluate the same embeddings induced using our proposed method in the cross-lingual **document classification** task used by Klementiev et al. This task tests *semantic transfer* of information across languages. We also evaluate our method in a **word-level translation** task which tests fine-grained *lexical transfer*.

6.1 Cross-lingual Document Classification

We use an **exact replication**[5] of the cross-lingual document classification (**CLDC**) setup used by Klementiev et al. [12] to evaluate their cross-lingual embeddings. The CLDC task setup is as follows: The goal is to classify documents in a target language using only labelled documents in a source language. In other words, we train a classifier on the labelled training data in the source language and then attempt to apply the classifier as-is to the target data (known as "direct transfer"). Documents are represented as the tf-idf-weighted sum of the embedding vectors of the words that appear in the documents. Importantly, though, note that our method does not utilize any prior word-alignment information and instead we aim to learn this information purely from the bilingual corpora.

Similar to Klementiev et al. [12], we induce cross-lingual embeddings for the English-German language pair, and use the induced representations to classify a subset of the English and German sections of the Reuters RCV1/RCV2 multilingual corpora [14] as pertaining to one of four categories: `CCAT` (Corporate/Industrial), `ECAT` (Economics), `GCAT` (Government/Social), and `MCAT` (Markets).

For the classification experiments, 15,000 documents (for each language) were randomly selected from the RCV1/2 corpus, with one third (5,000) used as the **test set** and the remainder divided into **training sets** of sizes between 100 and 10,000, and a separate, **held-out validation set** of 1,000 documents used during the development of our models. Since our setup exactly mirrors Klementiev et al, we use the same baselines, namely: the *majority* class baseline, *glossed* (replacing words in the target document by their most frequently aligned words in the source language), and a stronger *MT* baseline (translating target documents into the source language using an SMT system).

[3] Such as the so-called "inverse" method of sampling from the CDF of $P(w)$ using a number drawn from the uniform distribution.

[4] See also `https://hips.seas.harvard.edu/blog/2013/03/03/the-alias-method-efficient-sampling-with-many-discrete-outcomes/`

[5] Obtained from the authors.

Method	*en* → *de*	*de* → *en*	Training Time (min)
Majority Baseline	46.8	46.8	-
Glossed Baseline	65.1	68.6	-
MT Baseline	68.1	67.4	-
Klementiev et al.	77.6	71.1	14,400 (10 days)
Bilingual Auto-encoders (BAEs)	**91.8**	72.8	4,800 (3.5 days)
BiCVM	83.7	71.4	15
BilBOWA (this work)	86.5	**75**	**6**

Table 1: Classification accuracy and training times for our method compared to Klementiev et al. [12], Bilingual Auto-encoders [5], and the BiCVM model [11], on an exact replica of the Reuters cross-lingual document classification task. These methods were all trained with access to the same data using the same 40-dimensional embeddings. Baseline results are from Klementiev.

6.1.1 Results

Since our method is a faster version of the model proposed by Klementiev et al., we first compare directly to their results. Results are summarized in Table 1. To make our results comparable to other methods, our models were trained on the English-German Europarl data. Notat that this is a parallel corpus, but we do not exploit its parallel nature. We use a vocabulary size of $46,678$ for English and $47,903$ for German, vs Klementiev et al.'s $3K$ (at the output layer). We significantly improve upon their results, while training in 6 minutes versus the original 10 days (14,400 minutes) for a total factor $2,400$ speedup. This demonstrates that the BilBOWA loss is both a computationally efficient and accurate approximation of the cross-lingual objective implemented by Klementiev.

Next, we compare our method to the current state-of-the-art embedding methods trained using *the same embedding dimensionality* of 40 (to make results comparable to the original Klementiev results) and *the same training data*. The current state-of-the-art on this task is 91.8 (en2de) and 72.8 (de2en) reported using the Bilingual Auto-encoder (BAE) model by [5]. Hermann et al. [11] report 83.7 and 71.4 with the BiCVM model, trained using the same data that we trained on. As shown, our model outperforms the BiCVM on both tasks, and outperforms BAEs on German to English with the current state-of-the-art result of 75%. Overall, our method is also the fastest.

6.2 WMT11 Word Translation

Next, we evaluated the induced cross-lingual embeddings on the word translation task used by Mikolov et al. [15] using the publicly-available WMT11 data[6].

6.2.1 Setup and baselines

In this task, the authors extracted the $6K$ most frequent words from the WMT11 English-Spanish data, and then used the online Google Translate service to derive dictionaries by translating these source words into the target language (individually for English and Spanish). Since their method requires translation-pairs for training, they used the first $5K$ most frequent words to learn the "translation matrix", and then evaluated their method on the remaining $1K$ words used as a test set. To translate a source word, one finds its k nearest neighbours in the target language embedding space, and then evaluate the translation precision $P@k$ as the fraction of target translations that are within the top-k words returned using the specific method. Our method does not require translation-pairs for training, so we simply test on the same $1K$ test-pairs.

We use as baselines the same two methods described in Mikolov et al. [15]. "Edit Distance" ranks words based on their edit-distance. "Word Co-occurrence" is based on distributional similarity: For each word w, one first constructs a word co-occurrence vector which counts the words with which w co-occurs within a 10-word window in the corpus. The word count vectors are then mapped from the source to the target language using the dictionary. Finally, for each test word, the word with the most similar vector in the target language is selected as its translation.

[6] `http://www.statmt.org/wmt11/`

Method	En→Sp P@1	Sp→En P@1	En→Sp P@5	Sp→En P@5
Edit Distance	13	18	24	27
Word Co-occurrence	30	19	20	30
Mikolov et al., 2013	33	35	**51**	52
BilBOWA (This work)	**39** (+6)	**44** (+11)	**51**	**55** (+3)

Table 2: Results for the translation task measured as word translation accuracy (out of 100, higher is better) evaluated on the top-1 and top-5 words as ranked by the method. Cross-lingual embeddings are induced and distance in the embedded space are used to select word translation pairs. $+x$ indicates improvement in absolute precision over the previous state-of-the-art on this task [15].

6.2.2 Results

We induced 40-dimensional embeddings using the English and Spanish Wikipedias and Europarl as parallel data. The results on the English-Spanish translation tasks are summarized in Table 2. Our model improves on both the baselines and on *Mikolov et al.*'s method on both tasks and is significantly more accurate for the P@1 prediction. For the English to Spanish translation, we improve *absolute word translation accuracy* by 6 percentage points, for a *relative error reduction* of 8.95 %. For the Spanish to English task, we improve absolute word translation accuracy by 11 percent, for a relative error reduction of 16.9%. This indicates that our model is able to learn fine-grained translation equivalences by using only the raw text, sentence-aligned parallel data, despite the lack of word-level alignments or training dictionaries.

7 Conclusion

In this work we introduced BilBOWA[7], a computationally-efficient model for inducing bilingual distributed word representations directly from raw text without requiring word-alignments or dictionaries. Instead, the model directly utilizes a limited amount of parallel data to learn bilingual word representations. BilBOWA combines advances in training monolingual word embeddings with a particularly efficient novel sampled cross-lingual objective. The result is that the required computations per training step scales only with the number of words in the sentences, thereby enabling efficient large-scale cross-lingual training. We evaluated BilBOWA on English-German cross-lingual document classification and achieved state-of-the-art results while reducing training time to only a few minutes, rather than several days as for most previous approaches. We also evaluated on an English-Spanish word-translation task where we improve upon the previous state of the art in relative word translation error rate.

References

[1] Rami Al-Rfou', Bryan Perozzi, and Steven Skiena. Polyglot: Distributed word representations for multilingual nlp. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 183–192, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.

[2] Y Bengio, R Ducharme, and P Vincent. A neural probabilistic language model. *Journal of Machine Learning Research*, 2003.

[3] Yoshua Bengio and J-S Senecal. Adaptive importance sampling to accelerate training of a neural probabilistic language model. *Neural Networks, IEEE Transactions on*, 19(4):713–722, 2008.

[4] J. Blitzer, R. McDonald, and F. Pereira. Domain adaptation with structural correspondence learning. In *Conference on Empirical Methods in Natural Language Processing*, Sydney, Australia, 2006.

[5] Sarath Chandar, Stanislas Lauly, Hugo Larochelle, Mitesh M. Khapra, Balaraman Ravidran, Vikas Raykar, and Amrita Saha. An autoencoder approach to learning bilingual word representations. *Proceedings of NIPS 2014*, 2014.

[7] `https://code.google.com/p/word2vec/`

[6] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537, 2011.

[7] Hal Daumé III. Frustratingly easy domain adaptation. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2009.

[8] Chris Dyer, Victor Chahuneau, and Noah A Smith. A simple, fast, and effective reparameterization of ibm model 2. ACL, 2013.

[9] Manaal Faruqui and Chris Dyer. Improving vector space word representations using multilingual correlation. In *Proceedings of EACL 2014*, 2014.

[10] Yoav Goldberg and Omer Levy. word2vec explained: Deriving mikolov et al.s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*, 2014.

[11] Karl Moritz Hermann and Phil Blunsom. Multilingual distributed representations without word alignment. *arXiv preprint arXiv:1312.6173*, 2013.

[12] Alexandre Klementiev, Ivan Titov, and Binod Bhattarai. Inducing crosslingual distributed representations of words. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, Bombay, India, December 2012.

[13] Philipp Koehn. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86, 2005.

[14] David D Lewis, Yiming Yang, Tony G Rose, and Fan Li. Rcv1: A new benchmark collection for text categorization research. *The Journal of Machine Learning Research*, 5:361–397, 2004.

[15] Tomas Mikolov, Quoc V Le, and Ilya Sutskever. Exploiting similarities among languages for machine translation. In *International Conference on Learning Representations (ICLR)*, 2013.

[16] Andriy Mnih and Yee Whye Teh. A fast and simple algorithm for training neural probabilistic language models. In *Proceedings of the 29th International Conference on Machine Learning (ICML)*, 2012.

[17] Franz Josef Och and Hermann Ney. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51, 2003.

[18] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *Knowledge and Data Engineering, IEEE Transactions on*, 22(10):1345–1359, 2010.

[19] J. Turian, L. Ratinov, and Y. Bengio. Word representations: A simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 384–394. Association for Computational Linguistics, 2010.

[20] Alastair J Walker. An efficient method for generating discrete random variables with general distributions. *ACM Transactions on Mathematical Software (TOMS)*, 3(3):253–256, 1977.

[21] Will Y Zou, Richard Socher, Daniel Cer, and Christopher D Manning. Bilingual word embeddings for phrase-based machine translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2013.