An Autoencoder Approach to Learning Bilingual Word Representations

Sarath Chandar A P  APSARATHCHANDAR@GMAIL.COM
Indian Institute of Technology Madras, India.

Stanislas Lauly  STANISLAS.LAULY@USHERBROOKE.CA
Université de Sherbrooke, Canada.

Hugo Larochelle  HUGO.LAROCHELLE@USHERBROOKE.CA
Université de Sherbrooke, Canada.

Mitesh M. Khapra  MIKHAPRA@IN.IBM.COM
IBM Research India.

Balaraman Ravindran  RAVI@CSE.IITM.AC.IN
Indian Institute of Technology Madras, India.

Vikas Raykar  VIRAYKAR@IN.IBM.COM
IBM Research India.

Amrita Saha  AMRSAHA4@IN.IBM.COM
IBM Research India.

* Both authors contributed equally.

Abstract

Cross-language learning allows us to use training data from one language to build models for a different language. Many approaches to bilingual learning require that we have word-level alignment of sentences from parallel corpora. In this work we explore the use of autoencoder-based methods for cross-language learning of vectorial word representations that are aligned between two languages, while not relying on word-level alignments. We show that by simply learning to reconstruct the bag-of-words representations of aligned sentences, within and between languages, we can in fact learn high-quality representations and do without word alignments. Since training autoencoders on word observations presents certain computational issues, we propose and compare different variations adapted to this setting. We also propose an explicit correlation maximizing regularizer that leads to significant improvement in the performance. We empirically investigate the success of our approach on the problem of cross-language test classification, where a classifier trained on a given language (e.g., English) must learn to generalize to a different language (e.g., German). These experiments demonstrate that our approaches are competitive with the state-of-the-art, achieving up to 10-14 percentage point improvements over the best reported results on this task.

1. Introduction

Languages such as English, which have plenty of annotated resources at their disposal have better Natural Language Processing (NLP) capabilities than other languages that are not so fortunate in terms of annotated resources. For example, high quality POS taggers (Toutanova et al., 2003), parsers (Socher et al., 2013), sentiment analyzers (Liu, 2012) are already available for English but this is not the case for many other languages such as Hindi, Marathi, Bodo, Farsi, Urdu, etc. This situation was acceptable in the past when only a few languages dominated the digital content available online and elsewhere. However, the ever increasing number of languages on the web today has made it important to accurately process natural language data in such lesser-fortunate languages also. An obvious solution
to this problem is to improve the annotated inventory of these languages but the involved cost, time and effort act as a natural deterrent to this.

Another option is to exploit the unlabeled data available in a language. In this context, vectorial text representations have proven useful for multiple NLP tasks (Turian et al., 2010; Collobert et al., 2011). It’s been shown that meaningful representations, capturing syntactic and semantic similarity, can be learned from unlabeled data. Along with a (usually smaller) set of labeled data, these representations allow to exploit unlabeled data and improve the generalization performance on some given task, even allowing to generalize out of the vocabulary observed in the labeled data only (thereby, partly alleviating the problem of data sparsity).

While the majority of previous work on vectorial text representations has concentrated on the monolingual case, recent work has started looking at learning word and document representations that are aligned across languages (Klementiev et al., 2012; Zou et al., 2013; Mikolov et al., 2013). Such aligned representations can potentially allow the use of resources from a resource fortunate language to develop NLP capabilities in a resource deprived language (Yarowsky and Ngai, 2001; Das and Petrov, 2011; Mihalcea et al., 2007; Wan, 2009; Padó and Lapata, 2009). For example, if a common representation model is learned for representing English and German documents, then a classifier trained on annotated English documents can be used to classify German documents (provided we use the learned common representation model for representing documents in both languages).

Such reuse of resources across languages has been tried in the past by projecting parameters learned from the annotated data of one language to another language (Yarowsky and Ngai, 2001; Das and Petrov, 2011; Mihalcea et al., 2007; Wan, 2009; Padó and Lapata, 2009). These projections are enabled by a bilingual resource such as a Machine Translation (MT) system. Recent attempts at learning common bilingual representations (Klementiev et al., 2012; Zou et al., 2013; Mikolov et al., 2013) aim to eliminate the need of such a MT system. Such bilingual representations have been applied to a variety of problems, including cross-language document classification (Klementiev et al., 2012) and phrase-based machine translation (Zou et al., 2013). A common property of these approaches is that a word-level alignment of bilingual corpora during training. Unlike previous approaches (Klementiev et al., 2012), we only require aligned sentences and do not rely on word-level alignments (e.g., extracted using GIZA++, as is usual), which simplifies the learning procedure. To do so, we propose a bilingual autoencoder model, that learns hidden encoder representations of paired bag-of-words sentences which are not only informative of the original bag-of-words but also predictive of each other. Word representations can then easily be extracted from the encoder and used in the context of a supervised NLP task. Specifically, we demonstrate the quality of these representations for the task of cross-language document classification, where a labeled data set can be available in one language, but not in another one. As we’ll see, our approach is able to reach state-of-the-art performance, achieving up to 10-14 percentage point improvements over the best previously reported results.

2. Autoencoder for Bags-of-Words

Let \( x \) be the bag-of-words representation of a sentence. Specifically, each \( x_i \) is a word index from a fixed vocabulary of \( V \) words. As this is a bag-of-words, the order of the words within \( x \) does not correspond to the word order in the original sentence. We wish to learn a \( D \)-dimensional vectorial representation of our words from a training set of sentence bag-of-words \( \{x^{(t)}\}_{t=1}^T \).

We propose to achieve this by using an autoencoder model that encodes an input bag-of-words \( x \) with a sum of the representations (embeddings) of the words present in \( x \), followed by a nonlinearity. Specifically, let matrix \( W \) be the \( D \times V \) matrix whose columns are the vector representations for each word. The encoder’s computation will involve summing over the columns of \( W \) for each word in the bag-of-word. We will note this encoder function \( \phi(x) \). Then, using a decoder, the autoencoder will be trained to optimize a loss function that measures how predictive of the original bag-of-words the encoder representation \( \phi(x) \) is.

There are different variations we can consider, in the design of the encoder/decoder and the choice of loss function. One must be careful however, as certain choices can be inappropriate for training on word observations, which are intrinsically sparse and high-dimensional. In this paper, we explore and compare two different approaches, described in the next two sub-sections.

2.1. Binary bag-of-words reconstruction training with merged mini-batches

In the first approach, we start from the conventional autoencoder architecture, which minimizes a cross-entropy loss that compares a binary vector observation with a decoder
reconstruction. We thus convert the bag-of-words $\mathbf{x}$ into a fixed-size but sparse binary vector $\mathbf{v}(\mathbf{x})$, which is such that $v(x)_i = 1$ if word $x_i$ is present in $\mathbf{x}$ or otherwise 0.

From this representation, we obtain an encoder representation by multiplying $\mathbf{v}(\mathbf{x})$ with the word representation matrix $\mathbf{W}$

$$\phi(\mathbf{x}) = \mathbf{h}(\mathbf{c} + \mathbf{Wv}(\mathbf{x}))$$  \hspace{1cm} (1)

where $\mathbf{h}(\cdot)$ is an element-wise non-linearity such as the sigmoid or hyperbolic tangent, and $\mathbf{c}$ is a $D$-dimensional bias vector. Encoding thus involves summing the word representation of the words present at least once in the bag-of-words.

To produce a reconstruction, we parametrize the decoder using the following non-linear form:

$$\hat{\mathbf{v}}(\mathbf{x}) = \text{sign}((\mathbf{V} \phi(\mathbf{x}) + \mathbf{b}))$$  \hspace{1cm} (2)

where $\mathbf{V} = \mathbf{W}^T$ and $\mathbf{b}$ is the bias vector of the reconstruction layer and $\text{sign}(a) = 1/(1 + \exp(-a))$ is the sigmoid non-linearity.

Then, the reconstruction is compared to the original binary bag-of-words as follows:

$$\ell(\mathbf{v}(\mathbf{x})) = -\sum_{i=1}^{|\mathbf{x}|} v(x)_i \log(\hat{v}(x)_i) + (1 - v(x)_i) \log(1 - \hat{v}(x)_i).$$  \hspace{1cm} (3)

Training then proceeds by optimizing the sum of reconstruction cross-entropies across the training set, e.g., using stochastic or mini-batch gradient descent.

Note that, since the binary bag-of-words are very high-dimensional (the dimensionality corresponds to the size of the vocabulary, which is typically large), the above training procedure which aims at reconstructing the complete binary bag-of-word, will be slow. Since we will later be training on millions of sentences, training on each individual sentence bag-of-words will be expensive.

Thus, we propose a simple trick, which exploits the bag-of-words structure of the input. Assuming we are performing mini-batch training (where a mini-batch contains a list of bag-of-words), we simply propose to merge the bags-of-words of the mini-batch into a single bag-of-word, and revert back to stochastic gradient descent. The resulting effect is that each update is as efficient as in stochastic gradient descent, but the number of updates per training epoch is divided by the mini-batch size. As we’ll see in the experimental section, we’ve found this trick to still produces good word representations, while sufficiently reducing training time.

We note that, additionally, we could have used the stochastic approach proposed by Dauphin et al. (2011) for reconstructing binary bag-of-words representations of documents, to further improve the efficiency of training. They use importance sampling to avoid reconstructing the whole $V$-dimensional input vector.

### 2.2. Tree-based decoder training

The previous autoencoder architecture worked with a binary vectorial representation of the input bag-of-word. In the second autoencoder architecture we investigated, we considered an architecture that instead works with the bag (unordered list) representation more directly.

Firstly, the encoder representation will now involve a sum of the representation of all words, reflecting the relative frequency of each word:

$$\phi(\mathbf{x}) = h(c + \sum_{i=1}^{|\mathbf{x}|} \mathbf{W}_i x_i).$$  \hspace{1cm} (4)

Notice that this implies that the scaling of the pre-activation (the input to the non-linearity) can vary between bags-of-words, depending on how many words it contains. Thus, we’ll optionally consider using the average of the representations, as opposed to the sum (this choice is cross-validated in our experiments).

Moreover, decoder training will assume that, from the decoder’s output, we can obtain a probability distribution over any word $\hat{x}$ observed at the reconstruction output layer $p(\hat{x} \mid \phi(\mathbf{x}))$. Then, we can treat the input bag-of-words as a $|\mathbf{x}|$-trials multinomial sample from that distribution and use as the reconstruction loss its negative log-likelihood:

$$\ell(\mathbf{x}) = -\sum_{i=1}^{|\mathbf{x}|} \log p(\hat{x} = x_i \mid \phi(\mathbf{x})).$$  \hspace{1cm} (5)

We now must ensure that the decoder can compute $p(\hat{x} = x_i \mid \phi(\mathbf{x}))$ efficiently from $\phi(\mathbf{x})$. Specifically, we’d like to avoid a procedure scaling linearly with the vocabulary size $V$, since $V$ will be very large in practice. This precludes any procedure that would compute the numerator of $p(\hat{x} = w \mid \phi(\mathbf{x}))$ for each possible word $w$ separately and normalize so it sums to one.

We instead opt for an approach borrowed from the work on neural network language models (Morin and Bengio, 2005; Mnih and Hinton, 2009). Specifically, we use a probabilistic tree decomposition of $p(\hat{x} = x_i \mid \phi(\mathbf{x}))$. Let’s assume each word has been placed at the leaf of a binary tree. We can then treat the sampling of a word as a stochastic path from the root of the tree to one of the leaves.

We denote as $l(x)$ the sequence of internal nodes in the path from the root to a given word $x$, with $l(x)_1$ always corresponding to the root. We will denote as $\pi(x)$ the vector of associated left/right branching choices on that path, where $\pi(x)_k = 0$ means the path branches left at internal node
\(l(x)_k\) and branches right if \(\pi(x)_k = 1\) otherwise. Then, the probability \(p(\hat{x} = x | \phi(x))\) of reconstructing a certain word \(x\) observed in the bag-of-words is computed as

\[
p(\hat{x} | \phi(x)) = \prod_{k=1}^{\left|\pi(\hat{x})\right|} p(\pi(\hat{x})_k | \phi(x))
\]

(6)

where \(p(\pi(\hat{x})_k | \phi(x))\) is output by the decoder. By using a full binary tree of words, the number of different decoder outputs required to compute \(p(\hat{x} | \phi(x))\) will be logarithmic in the vocabulary size \(V\). Since there are \(|x|\) words in the bag-of-words, at most \(O(|x| \log V)\) outputs are required from the decoder. This is of course a worst case scenario, since words will share internal nodes between their paths, for which the decoder output can be computed just once. As for organizing words into a tree, as in Larochelle and Lauly (2012) we used a random assignment of words to the leaves of the full binary tree, which we have found to work well in practice.

Finally, we need to choose a parametrized form for the decoder. We choose the following non-linear form:

\[
p(\pi(\hat{x})_k = 1 | \phi(x)) = \text{sign}(b_{l(\hat{x})_k} + V_{l(\hat{x})_k} \cdot \phi(x))
\]

(7)

where \(b\) is a \((V-1)\)-dimensional bias vector and \(V\) is a \((V - 1) \times D\) matrix. Each left/right branching probability is thus modeled with a logistic regression model applied on the encoder representation of the input bag-of-words \(\phi(x)\).

### 3. Bilingual autoencoders

Let’s now assume that for each sentence bag-of-words \(x\) in some source language \(X\), we have an associated bag-of-words \(y\) for the same sentence translated in some target language \(Y\) by a human expert.

Assuming we have a training set of such \((x, y)\) pairs, we’d like to use it to learn representations in both languages that are aligned, such that pairs of translated words have similar representations.

To achieve this, we propose to augment the regular autoencoder proposed in Section 2 so that, from the sentence representation in a given language, a reconstruction can be attempted of the original sentence in the other language. Specifically, we now define language specific word representation matrices \(W^X\) and \(W^Y\) corresponding to the languages of the words in \(x\) and \(y\) respectively. Let \(V^X\) and \(V^Y\) also be the number of words in the vocabulary of both languages, which can be different. The word representations however are of the same size \(D\) in both languages. For the binary reconstruction autoencoder, the bag-of-words representations extracted by the encoder becomes

\[
\phi(x) = h(c + W^X v(x)), \quad \phi(y) = h(c + W^Y v(y))
\]

and are similarly extended for the tree-based autoencoder. Notice that we share the bias \(c\) before the nonlinearity across encoders, to encourage the encoders in both languages to produce representations on the same scale.

From the sentence in either languages, we want to be able to perform a reconstruction of the original sentence in any of the languages. In particular, given a representation in any language, we’d like a decoder that can perform a reconstruction in language \(X\) and another decoder that can reconstruct in language \(Y\). Again, we use decoders of the form proposed in either Section 2.1 or 2.2 (see Figures 1 and 2), but let the decoders of each language have their own parameters \((b^X, V^X)\) and \((b^Y, V^Y)\).

This encoder/decoder decomposition structure allows us to learn a mapping within each language and across the languages. Specifically, for a given pair \((x, y)\), we can train the model to (1) construct \(y\) from \(x\) (loss \(\ell(x, y)\)), (2) construct \(x\) from \(y\) (loss \(\ell(y, x)\)), (3) reconstruct \(x\) from itself (loss \(\ell(x)\)) and (4) reconstruct \(y\) from itself (loss \(\ell(y)\)). We follow this approach in our experiments and optimize the sum of the corresponding 4 losses during training.

### 3.1. Cross-lingual correlation regularization

The bilingual encoder proposed above can be further enriched by ensuring that the embeddings learned for a given pair \((x, y)\) are highly correlated. We achieve this by adding a correlation term to the objective function. Specifically, we could optimize

\[
\ell(x, y) + \ell(y, x) - \lambda \cdot \text{cor}(\phi(x), \phi(y))
\]

(8)

where \(\text{cor}(\phi(x), \phi(y))\) is the correlation between the encoder representations learned for \(x\) and \(y\) and \(\lambda\) is a scaling factor which ensures that the three terms in the loss function have the same range. Note that this approach could be used for either the binary bag-of-words or the tree-based reconstruction autoencoders.

### 3.2. Document representations

Once we learn the language specific word representation matrices \(W^X\) and \(W^Y\) as described above, we can use them to construct document representations, by using their columns as vector representations for words in both languages. Now, given a document \(d\) written in language \(Z \in \{X, Y\}\) and containing \(m\) words, \(z_1, z_2, \ldots, z_m\), we represent it as the tf-idf weighted sum of its words’ representations:

\[
\psi(d) = \sum_{i=1}^{m} \text{tf-idf}(z_i) \cdot W^Z_{z, z_i}
\]

(9)

We use the document representations thus obtained to train
An Autoencoder Approach to Learning Bilingual Word Representations

Figure 1. Illustration of a binary reconstruction error based bilingual autoencoder that learns to reconstruct the binary bag-of-words of the English sentence “the dog barked” from its French translation “le chien a jappé”.

4. Related Work

Recent work that has considered the problem of learning bilingual representations of words usually has relied on word-level alignments. Klementiev et al. (2012) propose to train simultaneously two neural network languages models, along with a regularization term that encourages pairs of frequently aligned words to have similar word embeddings. Zou et al. (2013) use a similar approach, with a different form for the regularizer and neural network language models as in (Collobert et al., 2011). In our work, we specifically investigate whether a method that does not rely on word-level alignments can learn comparably useful multilingual embeddings in the context of document classification.

Looking more generally at neural networks that learn multilingual representations of words or phrases, we mention the work of Gao et al. (2013) which showed that a useful linear mapping between separately trained monolingual skip-gram language models could be learned. They too however rely on the specification of pairs of words in the two languages to align. Mikolov et al. (2013) also propose a method for training a neural network to learn useful representations of phrases (i.e. short segments of words), in the context of a phrase-based translation model. In this case, phrase-level alignments (usually extracted from word-level alignments) are required.

Figure 2. Illustration of a tree-based bilingual autoencoder that learns to construct the bag-of-words of the English sentence “the dog barked” from its French translation “le chien a jappé”. The horizontal blue line across the input-to-hidden connections highlights the fact that these connections share the same parameters (similarly for the hidden-to-output connections).

5. Experiments

The techniques proposed in this paper enable us to learn bilingual embeddings which capture cross-language similarity between words. We propose to evaluate the quality of these embeddings by using them for the task of cross-language document classification. We follow the same setup as used by Klementiev et al. (2012) and compare with their method. The set up is as follows. A labeled data set of documents in some language $\mathcal{X}$ is available to train a classifier, however we are interested in classifying documents in a different language $\mathcal{Y}$ at test time. To achieve this, we leverage some bilingual corpora, which importantly is not labeled with any document-level categories. This bilingual corpora is used instead to learn document representations in both languages $\mathcal{X}$ and $\mathcal{Y}$ that are encouraged to be invariant to translations from one language to another. The hope is thus that we can successfully apply the classifier trained on document representations for language $\mathcal{X}$ directly to the document representations for language $\mathcal{Y}$. We use English (EN) and German (DE) as the language pair for all our experiments.

5.1. Data

For learning the bilingual embeddings, we used the English German section of the Europarl corpus (Koehn, 2005) which contains roughly 2 million parallel sentences. As mentioned earlier, unlike Klementiev et al. (2012), we do not use any word alignments between these parallel sentences. We use the same pre-processing as used by Klem-
be summarized as follows:

Our overall procedure for cross language classification can

5.2. Cross language classification

Note that Klementiev et al. (2012) use the word-aligned

5.3. Different models for learning embeddings

Next for the cross language classification experiments, we

6. Results and Discussions

Before discussing the results of cross language classification,

embeddings learned by our method. For this, we perform a small experiment where we select a few English words and list the top 10 English and German words which are most similar to these words (in terms of the Euclidean distance between their embeddings as learned by BAE-cr/corr). Table 1 shows the result of this experiment. For example, Table 1 shows that in all the cases the German word which is closest to a given English word is actually the translation of that English word. Also, notice that the model is able to capture semantic similarity between words by embedding semantically similar words (such as, (january, march), (gesagt, sagte), (market, commercial), etc.) close to each other. The results of this experiment suggest that these bilingual embeddings should be useful for any cross language classification task as indeed shown by the results presented in the next section. The supplementary material also includes a 2D visualization of the word embeddings in both languages, generated using the t-SNE dimensionality reduction algorithm (van der Maaten and Hinton, 2008).

6.1. Comparison of the performance of different models

We now present the cross language classification results obtained by using the embeddings produced by each of the 3 models described above. We also compare our models with the following approaches, for which the results are reported in Klementiev et al. (2012):

- Klementiev et al.: This model uses word embeddings learned by a multitask neural network language model with a regularization term that encourages pairs of frequently aligned words to have similar word embeddings. From these embeddings, document representations are computed as described in Section 3.2.
- MT: Here, test documents are translated to the language of the training documents using a Machine Translation (MT) system. MOSES\footnote{http://www.statmt.org/moses/}, a standard phrase-based MT system, using default parameters and a 5-gram language model was trained on the Europarl v7 corpus (same as the one used for inducing our bilingual embeddings).
- Majority Class: Every test document is simply assigned the Majority class prevalent in the training data.

Table 2 summarizes the results obtained using 1K training data with different models. We report results in both direc-
Table 2. Classification Accuracy for training on English and German with 1000 labeled examples

<table>
<thead>
<tr>
<th></th>
<th>EN → DE</th>
<th>DE → EN</th>
</tr>
</thead>
<tbody>
<tr>
<td>BAE-tr</td>
<td>80.2</td>
<td>68.2</td>
</tr>
<tr>
<td>BAE-cr</td>
<td>78.2</td>
<td>63.6</td>
</tr>
<tr>
<td>BAE-cr/corr</td>
<td>91.8</td>
<td>72.8</td>
</tr>
<tr>
<td>Klementiev et al.</td>
<td>77.6</td>
<td>71.1</td>
</tr>
<tr>
<td>MT</td>
<td>68.1</td>
<td>67.4</td>
</tr>
<tr>
<td>Majority Class</td>
<td>46.8</td>
<td>46.8</td>
</tr>
</tbody>
</table>

6.2. Effect of varying training size

Next, we evaluate the effect of varying the amount of supervised training data for training the classifier, with either BAE-tr, BAE-cr/corr or Klementiev et al. (2012) embeddings. We experiment with training sizes of 100, 200, 500, 1000, 5000 and 10000. These results for EN-DE and DE-EN are summarized in Figure 3 and Figure 4 respectively. We observe that BAE-cr/corr clearly outperforms the other models at almost all data sizes. More importantly, it performs remarkably well at very low data sizes (t=100) which suggests that it indeed learns very meaningful embeddings which can generalize well even at low data sizes.

6.3. Effect of coarser alignments

The excellent performance of BAE-cr/corr suggests that merging mini-batches into single bags-of-words does not significantly impact the quality of the word embeddings. In other words, not only do we not need to rely on word-level alignments, but exact sentence-level alignment is also not essential to reach good performances. It is thus natural to ask the effect of using even coarser level alignments. We check this by varying the size of the merged mini-batches from 5, 25 to 50, for both BAE-cr/corr and BAE-tr. The cross language classification results obtained by using these coarser alignments are summarized in Table 3.

Table 3. Classification Accuracy for training on English and German with coarser alignments for 1000 labeled examples

<table>
<thead>
<tr>
<th></th>
<th>Sent. per doc</th>
<th>EN → DE</th>
<th>DE → EN</th>
</tr>
</thead>
<tbody>
<tr>
<td>BAE-tr</td>
<td>5</td>
<td>84.0</td>
<td>67.7</td>
</tr>
<tr>
<td>BAE-tr</td>
<td>25</td>
<td>83.0</td>
<td>63.4</td>
</tr>
<tr>
<td>BAE-tr</td>
<td>50</td>
<td>75.9</td>
<td>68.6</td>
</tr>
<tr>
<td>BAE-cr/corr</td>
<td>5</td>
<td>91.75</td>
<td>72.78</td>
</tr>
<tr>
<td>BAE-cr/corr</td>
<td>25</td>
<td>88.0</td>
<td>64.5</td>
</tr>
<tr>
<td>BAE-cr/corr</td>
<td>50</td>
<td>90.2</td>
<td>49.2</td>
</tr>
</tbody>
</table>

Figure 3. Crosslingual classification accuracy results with English documents for the train set and German documents for the test set

Table 3. Classification Accuracy for training on English and German with 1000 labeled examples

7. Conclusion and Future Work

We presented evidence that meaningful bilingual word representations could be learned without relying on word-level alignments and can even be successful on fairly coarse sentence-level alignments. In particular, we showed that even though our model does not use word level alignments, it is able to outperform a state of the art word representation learning method that exploits word-level alignments. In addition, it also outperforms a strong Machine Translation based baseline. We observed that using a correlation based regularization term leads to better bilingual embeddings which are highly correlated and hence perform better for cross language classification tasks.

As future work we would like to investigate extensions of our bag-of- words bilingual autoencoder to bags-of-n-grams, where the model would also have to learn representations for short phrases. Such a model should be particularly useful in the context of a machine translation system.
An Autoencoder Approach to Learning Bilingual Word Representations

We would also like to explore the possibility of converting our bilingual model to a multilingual model which can learn common representations for multiple languages given different amounts of parallel data between these languages.

References


