Learning Bilingual Sentence Embeddings via Autoencoding and Computing Similarities with a Multilayer Perceptron

Yunsu Kim\textsuperscript{2} Hendrik Rosendahl\textsuperscript{1,2} Nick Rossenbach\textsuperscript{2} Jan Rosendahl\textsuperscript{2} Shahram Khadivi\textsuperscript{1} Hermann Ney\textsuperscript{2}
\textsuperscript{1}eBay, Inc., Aachen, Germany \{hrosendahl,skhadivi\}@ebay.com
\textsuperscript{2}RWTH Aachen University, Aachen, Germany \{surname\}@cs.rwth-aachen.de

Abstract

We propose a novel model architecture and training algorithm to learn bilingual sentence embeddings from a combination of parallel and monolingual data. Our method connects autoencoding and neural machine translation to force the source and target sentence embeddings to share the same space without the help of a pivot language or an additional transformation. We train a multilayer perceptron on top of the sentence embeddings to extract good bilingual sentence pairs from nonparallel or noisy parallel data. Our approach shows promising performance on sentence alignment recovery and the WMT 2018 parallel corpus filtering tasks with only a single model.

1 Introduction

Data crawling is increasingly important in machine translation (MT), especially for neural network models. Without sufficient bilingual data, neural machine translation (NMT) fails to learn meaningful translation parameters (Koehn and Knowles, 2017). Even for high-resource language pairs, it is common to augment the training data with web-crawled bilingual sentences to improve the translation performance (Bojar et al., 2018).

Using crawled data in MT typically involves two core steps: mining and filtering. Mining parallel sentences, i.e. aligning source and target sentences, is usually done with lots of heuristics and features: document/URL meta information (Resnik and Smith, 2003; Esplá-Gomis and Forcada, 2009), sentence lengths with self-induced lexicon (Moore, 2002; Varga et al., 2005; Etchegoyhen and Azpeitia, 2016), word alignment statistics and linguistic tags (Ștefănescu et al., 2012; Kaufmann, 2012).

Filtering aligned sentence pairs also often involves heavy feature engineering (Taghipour et al., 2011; Xu and Koehn, 2017). Most of the participants in the WMT 2018 parallel corpus filtering task use large-scale neural MT models and language models as the features (Koehn et al., 2018).

Bilingual sentence embeddings can be an elegant and unified solution for parallel corpus mining and filtering. They compress the information of each sentence into a single vector, which lies in a shared space between source and target languages. Scoring a source-target sentence pair is done by computing similarity between the source embedding vector and the target embedding vector. It is much more efficient than scoring by decoding, e.g. with a translation model.

Bilingual sentence embeddings have been studied primarily for transfer learning of monolingual downstream tasks across languages (Hermann and Blunsom, 2014; Pham et al., 2015; Zhou et al., 2016). However, few papers apply it to bilingual corpus mining; many of them require parallel training data with additional pivot languages (Espana-Bonet et al., 2017; Schwenk, 2018) or lack an investigation into similarity between the embeddings (Guo et al., 2018).

This work solves these issues as follows:

- We propose a simple end-to-end training approach of bilingual sentence embeddings with parallel and monolingual data only of the corresponding language pair.
- We use a multilayer perceptron (MLP) as a trainable similarity measure to match source and target sentence embeddings.
- We compare various similarity measures for embeddings in terms of score distribution, geometric interpretation, and performance in downstream tasks.
- We demonstrate competitive performance in sentence alignment recovery and parallel cor-
pus filtering tasks without a complex combination of translation/language models.

- We analyze the effect of negative examples on training an MLP similarity, using different levels of negativity.

## 2 Related Work

Bilingual representation of a sentence was at first built by averaging pre-trained bilingual word embeddings (Huang et al., 2012; Klementiev et al., 2012). The compositionality from words to sentences is integrated into end-to-end training in Hermann and Blunsom (2014).

Explicit modeling of a sentence-level bilingual embedding was first discussed in Chandar et al. (2013), training an autoencoder on monolingual sentence embeddings of two languages. Pham et al. (2015) jointly learn bilingual sentence and word embeddings by feeding a shared sentence embedding to n-gram models. Zhou et al. (2016) add document-level alignment information to this model as a constraint in training.

Recently, sequence-to-sequence NMT models were adapted to learn cross-lingual sentence embeddings. Schwenk and Douze (2017) connect multiple source encoders to a shared decoder of a pivot target language, forcing the consistency of encoder representations. Schwenk (2018) extend this work to use a single encoder for many source languages. Both methods rely on N-way parallel training data, which are seriously limited to certain languages and domains. Artetxe and Schwenk (2018b) relax this data condition to pairwise parallel data including the pivot language, but it is still unrealistic for many scenarios (see Section 4.2). In contrast, our method needs only parallel and monolingual data for source and target languages of concern without any pivot languages.

Hassan et al. (2018) train a bidirectional NMT model with a single encoder-decoder, taking the average of top-layer encoder states as the sentence embedding. They do not include any details on the data or translation performance before/after the filtering with this embedding. Junczys-Dowmunt (2018) apply this method to WMT 2018 parallel corpus filtering task, yet showing significantly worse performance than a combination of translation/language models. Our method shows comparable results to such model combinations in the same task.

Guo et al. (2018) replace the decoder with a feedforward network and use the parallel sentences as input to the two encoders. Similarly to our work, the feedforward network measures the similarity of sentence pairs, except that the source and target sentence embeddings are combined via dot product instead of concatenation. Their model, however, is not directly optimizing the source and target sentences to be translations of each other; it only attaches two encoders in the output level without a decoder.

Based on the model of Artetxe and Schwenk (2018b), Artetxe and Schwenk (2018a) scale cosine similarity between sentence embeddings with average similarity of the nearest neighbors. Searching for the nearest neighbors among hundreds of millions of sentences may cause a huge computational problem. On the other hand, our similarity calculation is much quicker and support batch computation while preserving strong performance in parallel corpus filtering.

Neither of the above-mentioned methods utilize monolingual data. We integrate autoencoding into NMT to maximize the usage of parallel and monolingual data together in learning bilingual sentence embeddings.

## 3 Bilingual Sentence Embeddings

A bilingual sentence embedding function maps sentences from both the source and target language into a single joint vector space. Once we obtain such a space, we can search for a similar target sentence embedding given a source sentence embedding, or vice versa.

### 3.1 Model

In this work, we learn bilingual sentence embeddings via NMT and autoencoding given parallel and monolingual corpora. Since our purpose is to pair source and target sentences, translation is a natural base task to connect sentences in two different languages. We adopt a basic encoder-decoder approach from Sutskever et al. (2014). The encoder produces a fixed-length embedding of a source sentence, which is used by the decoder to generate the target hypothesis.

First, the encoder takes a source sentence $f_1^j = f_1, ..., f_j, ..., f_J$ (length $J$) as input, where each $f_j$ is a source word. It computes hidden representations $h_j \in \mathbb{R}^D$ for all source positions $j$:

$$h_j' = \text{enc}_{arc}(f_j')$$

(1)
Figure 1: Our proposed model for learning bilingual sentence embeddings. A decoder (above) is shared over two encoders (below). The decoder accepts a max-pooled representation from either one of the encoders as its first state $s_0$, depending on the training objective (Equation 7 and 8).

$\text{enc}_{\text{src}}$ is implemented as a bidirectional recurrent neural network (RNN). We denote a target output sentence by $e'_1 = e_1, ..., e_i, ..., e_I$ (length $I$). The decoder is an unidirectional RNN whose internal state for a target position $i$ is:

$$s_i = \text{dec}(s_{i-1}, e_{i-1})$$  \hspace{1cm} (2)

where its initial state is element-wise max-pooling of the encoder representations $h^f_I$:

$$s_0 = \text{maxpool}(h^f_I)$$

$$= \left[ \max_{j=1,...,J} h_{ij1}, \ldots, \max_{j=1,...,J} h_{ijd} \right]^\top$$  \hspace{1cm} (3)

We empirically found that the max-pooling performs much better than averaging or choosing the first ($h_1$) or last ($h_J$) representation. Finally, an output layer predicts a target word $e_i$:

$$p_\theta(e_i | e^{-1}_i, f^I_i) = \text{softmax}(\text{linear}(s_i))$$  \hspace{1cm} (4)

where $\theta$ denotes a set of model parameters.

Note that the decoder has access to the source sentence only through $s_0$, which we take as the sentence embedding of $f^I_i$. This assumes that the source sentence embedding contains sufficient information for translating to a target sentence, which is desired for a bilingual embedding space.

However, this plain NMT model can generate only source sentence embeddings through the encoder. The decoder cannot process a new target sentence without a proper source language input. We can perform decoding with an empty source input and take the last decoder state $s_I$ as the sentence embedding of $e'_I$, but it is not compatible with the source embedding and contradicts the way in which the model is trained.

Therefore, we attach another encoder of the target language to the same (target) decoder:

$$\hat{h}_i^I = \text{enc}_{\text{tgt}}(e'_i)$$  \hspace{1cm} (5)

$$s_0 = \left[ \max_{i=1,\ldots,I} \hat{h}_{1i}, \ldots, \max_{i=1,\ldots,I} \hat{h}_{D,i} \right]^\top$$  \hspace{1cm} (6)

$\text{enc}_{\text{tgt}}$ has the same architecture as $\text{enc}_{\text{src}}$. The model has now an additional information flow from a target input sentence to the same target (output) sentence, also known as sequential autoencoder (Li et al., 2015).

Figure 1 is a diagram of our model. A decoder is shared between NMT and autoencoding parts; it takes either source or target sentence embedding and does not differentiate between the two when producing an output. The two encoders are constrained to provide mathematically consistent representations over the languages (to the decoder).

Note that our model does not have any attention component (Bahdanau et al., 2014). The attention mechanism in NMT makes the decoder attend to encoder representations at all source positions. This is counterintuitive for our purpose; we need to optimize the encoder to produce a single representation vector, but the attention model allows the encoder to distribute information over many different positions. In our initial experiments, the same model with the attention mechanism showed exorbitantly bad performance, so we removed it in the main experiments of Section 4.

### 3.2 Training and Inference

Let $\theta_{\text{enc}_{\text{src}}}$, $\theta_{\text{enc}_{\text{tgt}}}$, and $\theta_{\text{dec}}$ the parameters of the source encoder, the target encoder, and the (shared) decoder, respectively. Given a parallel corpus $P$ and a target monolingual corpus $M_{\text{tgt}}$, the training criterion of our model is the cross-entropy on two input-output paths. The NMT objective (Equation 7) is for training $\theta_1 = \{\theta_{\text{enc}_{\text{src}}}, \theta_{\text{dec}}\}$, and the autoencoding objective (Equation 8) is for training $\theta_2 = \{\theta_{\text{enc}_{\text{tgt}}}, \theta_{\text{dec}}\}$:

$$L_{\text{emb}}(\theta) = -\sum_{(e'_I, f^I_I) \in P} \log p_\theta(e'_I | f^I_I)$$  \hspace{1cm} (7)
Let \( \theta = \{\theta_1, \theta_2\} \). During the training, each mini-batch contains examples of the both objectives with a 1:1 ratio. In this way, we prevent one encoder from being optimized more than the other, forcing the two encoders produce balanced sentence embeddings that fit to the same decoder.

The autoencoding part can be trained with a separate target monolingual corpus. To provide a stronger training signal for the shared embedding space, we use also the target side of \( \mathcal{P} \); the model learns to produce the same target sentence from the corresponding source and target inputs.

In order to guide the training to bilingual representations, we initialize the word embedding layers with a pre-trained bilingual word embedding. The word embedding for each language is trained with a skip-gram algorithm (Mikolov et al., 2013), later mapped across the languages with adversarial training (Conneau et al., 2018) and self-dictionary refinements (Artetxe et al., 2017).

Our model can be built also in the opposite direction, i.e. with a target-to-source NMT model and a source autoencoder:

\[
L_{emb}(\theta) = - \sum_{(f^i_1,e^i_1) \in \mathcal{P}} \log p_{\theta_2}(f^i_1 | e^i_1) - \sum_{f^i_1 \in \mathcal{M}_{\text{src}}} \log p_{\theta_1}(f^i_1 | f^i_1) \tag{9}
\]

Once the model is trained, we need only the encoders to query sentence embeddings. Let \( \mathbf{a} \) and \( \mathbf{b} \) be embeddings of a source sentence \( f^i_1 \) and a target sentence \( e^i_1 \), respectively:

\[
\mathbf{a} = \text{maxpool}(\text{enc}_{\text{src}}(f^i_1)) \tag{11}
\]

\[
\mathbf{b} = \text{maxpool}(\text{enc}_{\text{tgt}}(e^i_1)) \tag{12}
\]

### 3.3 Computing Similarities

The next step is to evaluate how close the two embeddings are to each other, i.e. to compute a similarity measure between them. In this paper, we consider two types of similarity measures.

**Predefined mathematical functions** Cosine similarity is a conventional choice for measuring the similarity in vector space modeling of information retrieval or text mining (Singhal, 2001). It computes the angle between two vectors (rotation) and ignore the lengths:

\[
\cos(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a} \cdot \mathbf{b}}{||\mathbf{a}|| ||\mathbf{b}||} \tag{13}
\]

Euclidean distance indicates how much distance must be traveled to move from the end of a vector to that of the other (transition). We reverse this distance to use it as a similarity measure:

\[
\text{Euclidean}(\mathbf{a}, \mathbf{b}) = -||\mathbf{a} - \mathbf{b}|| \tag{14}
\]

However, these simple measures, i.e. a single rotation or transition, might not be sufficient to define the similarity of complex natural language sentences across different languages. Also, the learned joint embedding space is not necessarily perfect in the sense of vector space geometry; even if we train it with a decent algorithm, the structure and quality of the embedding space are highly dependent on the amount of parallel training data and its domain. This might hinder the simple functions from working well for our purpose.

**Trainable multilayer perceptron** To model relations of sentence embeddings by combining rotation, shift, and even nonlinear transformations, We train a small multilayer perceptron (MLP) (Bishop et al., 1995) and use it as a similarity measure. We design the MLP network \( q(\mathbf{a}, \mathbf{b}) \) as a simple binary classifier whose input is a concatenation of source and target sentence embeddings: \([\mathbf{a}; \mathbf{b}]^T\). It is passed through feedforward hidden layers with nonlinear activations. The output layer has a single node with sigmoid activation, representing how probable the source and target sentences are translations of each other.

To train this model, we must have positive examples (real parallel sentence pairs, \( \mathcal{P}_{\text{pos}} \)) and negative examples (nonparallel or noisy sentence pairs, \( \mathcal{P}_{\text{neg}} \)). The training criterion is:

\[
L_{\text{sim}} = - \sum_{(\mathbf{a}, \mathbf{b}) \in \mathcal{P}_{\text{pos}}} \log q(\mathbf{a}, \mathbf{b}) - \sum_{(\mathbf{a}, \mathbf{b}) \in \mathcal{P}_{\text{neg}}} (1 - \log q(\mathbf{a}, \mathbf{b})) \tag{15}
\]

which naturally fits to the main task of interest: parallel corpus filtering (Section 4.2). Note that the output of the MLP can be quite biased to the extremes (0 or 1) in order to clearly distinguish good and bad examples. This has both advantages and disadvantages as explained in Section 5.1.

Our MLP similarity can be optimized differently for each embedding space. Furthermore, the user can inject domain-specific knowledge into the MLP similarity by training only with in-domain parallel data. The resulting MLP would devalue
not only nonparallel sentence pairs but also out-of-domain instances.

4 Evaluation

We evaluated our bilingual sentence embedding and the MLP similarity on two tasks: sentence alignment recovery and parallel corpus filtering. The sentence embedding was trained with WMT 2018 English-German parallel data and 100M German sentences from the News Crawl monolingual data\(^1\), where we use German as the autoencoded language. All sentences were lowercased and limited to the length of 60. We learned the byte pair encoding (Sennrich et al., 2016) jointly for the two languages with 20k merge operations. We pre-trained bilingual word embeddings on 100M sentences from the News Crawl data for each language using FASTTEXT (Bojanowski et al., 2017) and MUSE (Conneau et al., 2018).

Our sentence embedding model has 1-layer RNN encoder/decoder, where the word embedding and hidden layers have a size of 512. The training was done with stochastic gradient descent with initial learning rate of 1.0, batch size of 120 sentences, and maximum 800k updates. After 100k updates, we reduced the learning rate by a factor of 0.9 for every 50k updates.

Our MLP similarity model has 2 hidden layers of size 512 with ReLU (Nair and Hinton, 2010) with maximum 1,000 updates. For a positive training set, we used newstest2007-2015 from WMT (around 21k sentences). Unless otherwise noted, we took a comparable size of negative examples from the worst-scored sentence pairs of ParaCrawl\(^2\) English-German corpus. The scoring was done with our bilingual sentence embedding and cosine similarity.

Note that the negative examples are selected via cosine similarity but the similarity values are not used in the MLP training (Equation 15). Thus it does not learn to mimic the cosine similarity function again, but has a new sorting of sentence pairs—also encoding the domain information.

4.1 Sentence Alignment Recovery

In this task, we corrupt the sentence alignments of a parallel test set by shuffling one side, and find the original alignments; also known as corpus reconstruction (Schwenk and Douze, 2017).

Given a source sentence, we compute a similarity score with every possible target sentence in the data and take the top-scored one as the alignment. The error rate is the number of incorrect sentence alignments divided by the total number of sentences. We compute this also in the opposite direction and take an average of the two error rates. It is an intrinsic evaluation for parallel corpus mining. We choose two test sets: WMT newstest2018 (2998 lines) and IWSLT tst2015 (1080 lines).

As baselines, we used character-level Levenshtein distance and length-normalized posterior scores of German→English/English→German NMT models. Each NMT model is a 3-layer base Transformer (Vaswani et al., 2017) trained on the same training data as the sentence embedding.

| Method                           | Error [%] WMT | Error [%] IWSLT |
|----------------------------------|--------------|-----------------|
| Levenshtein distance             | 37.4         | 54.6            |
| NMT de-en + en-de                | 1.7          | 13.3            |
| Our method (Cosine similarity)    | 4.3          | 13.8            |
| Our method (MLP similarity)      | 89.9         | 72.6            |

Table 1: Sentence alignment recovery results. Our method results use cosine similarity except the last row.

Table 1 shows the results. The Levenshtein distance gives a poor performance. NMT models are better than the other methods, but takes too long to compute posteriors for all possible pairs of source and target sentences (about 12 hours for the WMT test set). This is absolutely not feasible for a real mining task with hundreds of millions of sentences.

Our bilingual sentence embeddings (with using cosine similarity) show error rates close to the NMT models, especially in the IWSLT test set. Computing similarities between embeddings is extremely fast (about 3 minutes for the WMT test set), which perfectly fits to mining scenarios.

However, the MLP similarity performs bad in aligning sentence pairs. Given a source sentence, it puts all reasonably similar target sentences to the score 1 and does not precisely distinguish between them. Detailed investigation of this behavior is in Section 5.1. As we will find out, this is ironically very effective in parallel corpus filtering.
4.2 Parallel Corpus Filtering

We also test our methods in the WMT 2018 parallel corpus filtering task (Koehn et al., 2018).

Data The task is to score each line of a very noisy, web-crawled corpus of 104M parallel lines (ParaCrawl English-German). We pre-filtered the given raw corpus with the heuristics of Rossenbach et al. (2018). Only the data for WMT 2018 English-German news translation task is allowed to train scoring models. The evaluation procedure is: subsample top-scored lines which amounts to 10M/100M words, train a small NMT model with the subsampled data, and check its translation performance. We follow the official pipeline except that we train 3-layer Transformer NMT model using Sockeye (Hieber et al., 2017) for evaluation.

Baselines We have three comparative baselines: 1) random sampling, 2) bilingual sentence embedding learned with a third pivot target language (Schwenk and Douze, 2017), 3) combination of source-to-target/target-to-source NMT and source/target LM (Rossenbach et al., 2018), a top-ranked system in the official evaluation.

Note that the second method violates the official data condition of the task since it requires parallel data in German-Pivot and English-Pivot. This method is not practical when learning multilingual embeddings for English and other languages, since it is hard to collect pairwise parallel data involving a non-English pivot language (except among European languages). We trained this method with N-way parallel UN corpus (Ziemski et al., 2016) with French as the pivot language. The size of this model is the same as that of our autoencoding-based model except the word embedding layers.

The results are shown in Table 2, where cosine similarity was used by default for sentence embedding methods except the last row. Pivot-based sentence embedding (Schwenk and Douze, 2017) improves upon the random sampling, but it has an impractical data condition. The four-model combination of NMT models and LMs (Rossenbach et al., 2018) provide 1-3% more BLEU improvement. Note that, for the third method, each model costs 1-2 weeks to train.

Our bilingual sentence embedding method greatly improves over the random sampling baseline up to 5.3% BLEU in the 10M-word case and 5.1% BLEU in the 100M-word case. With our MLP similarity, the improvement in BLEU is up to 12.3% and 8.2% in the 10M-word case and the 100M-word case, respectively. It outperforms the pivot-based embedding method significantly and gets close to the performance of the four-model combination. Note that we use only a single model trained with only given parallel/monolingual data for the corresponding language pair, i.e. English-German. In contrast to sentence alignment recovery experiments, the MLP similarity boosts the filtering performance by a large margin.

5 Analysis

In this section, we provide more in-depth analyses to compare 1) various similarity measures and 2) different choices of the negative training set for the MLP similarity model.

5.1 Similarity Measures

In Table 3, we compare sentence alignment recovery performance with different similarity measures.

Euclidean distance shows a worse performance than cosine similarity. This means that in a sentence embedding space, we should consider rotation more than transition when comparing two
Table 3: Sentence alignment recovery results with different similarity measures (newstest2018).

| Similarity   | de-en | en-de | Average |
|--------------|-------|-------|---------|
| Euclidean    | 7.9   | 99.8  | 53.8    |
| Cosine       | 4.3   | 4.2   | 4.3     |
| CSLS         | 1.9   | 2.2   | 2.1     |
| MLP          | 85.0  | 94.8  | 89.9    |

vectors. Particularly, the English→German direction has a peculiarly bad result with Euclidean distance. This is due to a hubness problem in a high-dimensional space, where some vectors are highly likely to be nearest neighbors of many others.

Figure 2 illustrates that Euclidean distance is more prone to the hubs than cosine similarity. Assume that German sentence embeddings \(a_n\) and English sentence embeddings \(b_n\) should match to each other with the same index \(n\), e.g. \((a_1,b_1)\) is a correct match. With cosine similarity, the nearest neighbor of \(a_n\) is always \(b_n\) for all \(n = 1, ..., 4\) and vice versa, considering only the angles between the vectors. However, when using Euclidean distance, there is a discrepancy between German→English and English→German directions: The nearest neighbor of each \(a_n\) is \(b_n\), but the nearest neighbor of all \(b_n\) is always \(a_4\). This leads to a serious performance drop only in English→German. The figure is depicted in a two-dimensional space for simplicity, but the hubness problem becomes worse for an actual high-dimensional space of sentence embeddings.

Cross-domain similarity local scaling (CSLS) is developed to counteract the hubness problem by penalizing similarity values in dense areas of the embedding distribution (Conneau et al., 2018):

\[
\text{CSLS}(a, b) = 2 \cdot \cos(a, b) - \frac{1}{K} \sum_{b' \in \text{NN}(a)} \cos(a, b') - \frac{1}{K} \sum_{a' \in \text{NN}(b)} \cos(a', b) \tag{16}
\]

where \(K\) is the number of nearest neighbors. CSLS outperforms cosine similarity in our experiments. For a large-scale mining scenario, however, the measure requires heavy computations for the penalty terms (Equation 17 and 18), i.e. nearest neighbor search in all combinations of source and target sentences and sorting the scores over e.g. a few hundred million instances.

The MLP similarity is not performing well as opposed to its results in parallel corpus filtering. To explain this, we depict score distributions of cosine and MLP similarity over the ParaCrawl corpus in Figure 3. As for cosine similarity, only...
The requested URL / dictionary / m / _mar_eisimpleir.htm was not found on this server. Additionally, a 404 Not Found error was encountered while trying to use an ErrorDocument to handle the request.

Becoming Prestigious In The Right Way how I Feel About School

According to his report, the Turkish army was aiming to provoke an international intervention against Syria.

To pay tribute to all people and workers who have been fighting for freedom or fallen in demonstrations and demand the immediate release of all detainees.

| German sentence | English sentence | Similarity |
|-----------------|------------------|------------|
| the requested URL / dictionary / m / _mar_eisimpleir.htm was not found on this server. Additionally, a 404 Not Found error was encountered while trying to use an ErrorDocument to handle the request. | how I Feel About School | Cosine: 0.185, MLP: 0.000 |
| nach dieser Aussage sollte die türkische Armee somit eine internationale Intervention gegen Syrien provozieren. | according to his report, the Turkish army was aiming to provoke an international intervention against Syria. | Cosine: 0.563, MLP: 1.000 |
| allen Menschen und Beschäftigten, die um Freiheit kämpfen oder bei Kundgebungen ums Leben kamen, Achtung zu bezeugen und die unverzügliche Freilassung aller Inhaftierten zu fordern | to pay tribute to all people and workers who have been fighting for freedom or fallen in demonstrations and demand the immediate release of all detainees | Cosine: 0.427, MLP: 0.999 |

Table 4: Example sentence pairs in the ParaCrawl corpus (Section 4.2) with their similarity values.
Negative examples | BLEU [%]
-------------------|----------
Random sampling    | 33.3     
20% worst          | 29.9     
40% worst          | 33.3     
60% worst          | 33.7     
80% worst          | 32.1     
100% worst         | 25.7     

Table 5: Parallel corpus filtering results (10M-word task) with different negative sets for training MLP similarity (newstest2016, i.e. the validation set).

100%.

4. Take the last 100k lines for each portion.

A negative set from the 20%-worst part stands for relatively less problematic sentence pairs, intending for elaborate classification among perfect parallel sentences (positive set) and almost perfect ones. With the 100%-worst examples, we focus on removing absolutely nonsense pairing of sentences. As a simple baseline, we also take 100k sentences randomly without scoring, representing mixed levels of negativity.

The results in Table 5 show that a moderate level of negativity (60%-worst) is most suitable for training an MLP similarity model. If the negative set contains too many excellent examples, the model may mark acceptable parallel sentence pairs with zero scores. If the negative set consists only of certainly nonparallel sentence pairs, the model is weak in discriminating mid-quality instances, some of which are crucial to improve the translation system.

Random selection of sentence pairs also works surprisingly well compared to carefully tailored negative sets. It does not require us to score and sort the raw corpus, so it is very efficient, sacrificing performance slightly. We hypothesize that the average negative level of this random set is also moderate and similar to that of the 60%-worst.

6 Conclusion

In this work, we present a simple method to train bilingual sentence embeddings by combining vanilla RNN NMT (without attention component) and sequential autoencoder. By optimizing a shared decoder with combined training objectives, we force the source and target sentence embeddings to share their space. Our model is trained with parallel and monolingual data of the corresponding language pair, with neither pivot languages nor N-way parallel data. We also propose to use a binary classification MLP as a similarity measure for matching source and target sentence embeddings.

Our bilingual sentence embeddings show consistently strong performance in both sentence alignment recovery and the WMT 2018 parallel corpus filtering tasks with only a single model. We compare various similarity measures for bilingual sentence matching, verifying that cosine similarity is preferred for a mining task and our MLP similarity is very effective in a filtering task. We also show that a moderate level of negativity is appropriate for training the MLP similarity, using either random examples or mid-range scored examples from a noisy parallel corpus.

Future work would be regularizing the MLP training to obtain a smoother distribution of the similarity scores, which could supplement the weakness of the MLP similarity (Section 5.1). Furthermore, we plan to adjust our learning procedure towards the downstream tasks, e.g. with an additional training objective to maximize the cosine similarity between the source and target encoders (Arivazhagan et al., 2019). Our method should be tested also on many other language pairs which do not have parallel data involving a pivot language.

Acknowledgments

This work has received funding from the European Research Council (ERC) (under the European Union’s Horizon 2020 research and innovation programme, grant agreement No 694537, project ”SEQCLAS”), the Deutsche Forschungsgemeinschaft (DFG; grant agreement NE 572/8-1, project ”CoreTec”), and eBay Inc. The GPU cluster used for the experiments was partially funded by DFG Grant INST 222/1168-1. The work reflects only the authors’ views and none of the funding agencies is responsible for any use that may be made of the information it contains.
References

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Roeo Aharoni, Melvin Johnson, and Wolfgang Macherey. 2019. The missing ingredient in zero-shot neural machine translation. arXiv:1903.07091.

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

Mikel Artetxe and Holger Schwenk. 2018b. Margin-based parallel corpus mining with multilingual sentence embeddings. arXiv:1811.01136.

Mikel Artetxe and Holger Schwenk. 2018a. Margin of 6th International Conference on Learning Representations (ICLR 2018).

Christopher M Bishop et al. 1995. Neural networks for pattern recognition. Oxford university press.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics, 5:135–146.

Ondřej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, and Christof Monz. 2018. Findings of the 2018 conference on machine translation (wmt18). In Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers, pages 272–307, Belgium, Brussels.

AP Sarath Chandar, Mitesh M Khapra, Balaraman Ravindran, Vikas Raykar, and Amrita Saha. 2013. Multilingual deep learning. In Deep Learning Workshop at NIPS.

Alexis Conneau, Guillaume Lample, Marc’Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word translation without parallel data. In Proceedings of 6th International Conference on Learning Representations (ICLR 2018).

Dan Stefănescu, Radu Ion, and Sabine Hunsicker. 2012. Hybrid parallel sentence mining from comparable corpora. In Proceedings of the 16th Conference of the European Association for Machine Translation, pages 137–144.

Cristina Espana-Bonet, Adám Csaba Varga, Alberto Barrón-Cedeño, and Josef van Genabith. 2017. An empirical analysis of nmt-derived interlingual embeddings and their use in parallel sentence identification. IEEE Journal of Selected Topics in Signal Processing, 11(8):1340–1350.

Miquel Esplá-Gomis and Mikel L. Forcada. 2009. Biextor, a free/open-source software to harvest translation memories from multilingual websites. In Proceedings of MT Summit XII, Ottawa, Canada.

Thierry Etchegoyhen and Andoni Azpeitia. 2016. Set-theoretic alignment for comparable corpora. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), volume 1, pages 2009–2018.

Mandy Guo, Qinlan Shen, Yinfei Yang, Heming Ge, Daniel Cer, Gustavo Hernandez Abrego, Keith Stevens, Noah Constant, Yun-hsuan Sung, Brian Strope, et al. 2018. Effective parallel corpus mining using bilingual sentence embeddings. In Proceedings of the Third Conference on Machine Translation: Research Papers, pages 165–176.

Hany Hassan, Anthony Aue, Chang Chen, Vishal Chowdhary, Jonathan Clark, Christian Federmann, Xuedong Huang, Marcin Junczys-Dowmunt, William Lewis, Mu Li, et al. 2018. Achieving human parity on automatic chinese to english news translation. arXiv preprint arXiv:1803.05567.

Karl Moritz Hermann and Phil Blunsom. 2014. Multilingual models for compositional distributed semantics. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), volume 1, pages 58–68.

Felix Hieber, Tobias Domhan, Michael Denkowski, David Vilar, Artem Sokolov, Ann Clifton, and Matt Post. 2017. Sockeye: A toolkit for neural machine translation. arXiv preprint arXiv:1712.05690.

Eric H Huang, Richard Socher, Christopher D Manning, and Andrew Y Ng. 2012. Improving word representations via global context and multiple word prototypes. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1, pages 873–882. Association for Computational Linguistics.

Marcin Junczys-Dowmunt. 2018. Dual conditional cross-entropy filtering of noisy parallel corpora. In Proceedings of the Third Conference on Machine Translation: Shared Task Papers, pages 888–895.

Max Kaufmann. 2012. Jmaxalign: A maximum entropy parallel sentence alignment tool. Proceedings of COLING 2012: Demonstration Papers, pages 277–288.

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

Philipp Koehn, Huda Khayrallah, Kenneth Heafield, and Mikel L. Forcada. 2018. Findings of the wmt 2018 shared task on parallel corpus filtering. In Proceedings of the Third Conference on Machine Translation: Shared Task Papers, pages 726–739.
