Effective Parallel Corpus Mining using Bilingual Sentence Embeddings

**Mandy Guo[a]*, Qinlan Shen[b]†*, Yinfei Yang[a]*, Heming Ge[a], Daniel Cer[a],
Gustavo Hernandez Abrego[a], Keith Stevens[a], Noah Constant[a],
Yun-Hsuan Sung[a], Brian Strope[a], Ray Kurzweil[a]**

[a]Google AI
Mountain View, CA, USA

[b]Carnegie Mellon University
Pittsburgh, PA, USA

Abstract

This paper presents an effective approach for parallel corpus mining using bilingual sentence embeddings. Our embedding models are trained to produce similar representations exclusively for bilingual sentence pairs that are translations of each other. This is achieved using a novel training method that introduces hard negatives consisting of sentences that are not translations but have some degree of semantic similarity. The quality of the resulting embeddings are evaluated on parallel corpus reconstruction and by assessing machine translation systems trained on gold vs. mined sentence pairs. We find that the sentence embeddings can be used to reconstruct the United Nations Parallel Corpus (Ziemski et al., 2016) at the sentence-level with a precision of 48.9% for en-fr and 54.9% for en-es. When adapted to document-level matching, we achieve a parallel document matching accuracy that is comparable to the significantly more computationally intensive approach of Uszkoreit et al. (2010). Using reconstructed parallel data, we are able to train NMT models that perform nearly as well as models trained on the original data (within 1-2 BLEU).

1 Introduction

Volumes of quality parallel training data are critical to neural machine translation (NMT) systems. While large distributed systems have proven useful for mining parallel documents (Uszkoreit et al., 2010; Antonova and Misyurev, 2011), these approaches are computationally intensive and rely on heavily engineered subsystems. Recent work has approached the problem by training lightweight end-to-end models based on word and sentence-level embeddings (Grégoire and Langlais, 2017; Bouamor and Sajjad, 2018; Schwenk, 2018). We propose a novel method for training bilingual sentence embeddings that proves useful for sentence-level mining of parallel data. Sentences are encoded using Deep Averaging Networks (DANs) (Iyyer et al., 2015), a simple bag of n-grams architecture that has been shown to provide surprisingly competitive performance on a number of tasks including sentence classification (Iyyer et al., 2015; Cer et al., 2018), conversation input-response prediction (Yang et al., 2018), and email response prediction (Henderson et al., 2017). Separate encoders are used for each language with candidate source and target sentences being paired based on the dot-product of their embedded representations. Training maximizes the dot-product score of sentence pairs that are translations of each other at the expense of sampled negatives. We contrast using random negatives with carefully selected hard negatives that challenge the model to distinguish between true translation pairs versus non-translation pairs that exhibit some degree of semantic similarity.

The efficiency of the sentence encoders and the use of a dot-product operation to score candidate sentence pairs is well suited for parallel corpus mining. Efficient encoders reduce the amount of computational resources required to obtain sentence embeddings for a large collection of unpaired sentences. Once the sentence embeddings are available, efficient nearest neighbour search (Vanderkam et al., 2013; Johnson et al., 2017) can be used to identify candidate translation pairs.

The language pairs English-French (en-fr) and English-Spanish (en-es) are used in our experiments. Our results show that introducing hard negative sentence pairs, which are semantically similar but that are not translations of each other, systematically outperforms using randomly selected negatives. Our method can be used to reconstruct the United Nations Parallel Corpus (Ziemski et al., 2016) at the sentence-level with a level of precision of 48.9% P@1 for en-fr and 54.9% P@1 for en-es. When we adapt our method to document-

[*] equal contribution
[†] Work done during an internship at Google AI.

Proceedings of the Third Conference on Machine Translation (WMT), Volume 1: Research Papers, pages 165–176
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64017

level pairings we achieve a matching accuracy that is comparable to that of the much heavier weight and more computationally intensive approach of Uszkoreit et al. (2010). Training an NMT model using the reconstructed corpus results in models that perform nearly as well as those trained on the original parallel corpus (within 1-2 BLEU). Finally, our method has a modest degree of correlation with the pair quality scores provided by Zipporah (Xu and Koehn, 2017). However, our method has higher agreement with human judgments, and our approach to filter the ParaCrawl corpus results in NMT systems with higher BLEU scores.

2 Approach

This section introduces our bilingual sentence embedding model and the translation candidate ranking task we use for training. We then present our method for selecting hard negative sentence pairs that are not translations of each other but have some degree of semantic similarity. Finally, we detail the use of our bilingual sentence embeddings to search for sentences that are translations of each other, as well as an adaptation to the matching process to parallel documents.

2.1 Translation Candidates Ranking Task

Given a pair of sentences that are translations of each other x and y, the translation candidate ranking task attempts to rank the true translation y over all other sentences, $\mathcal{Y}$, in the given language. This can be accomplished by modeling the translation probability distribution $P(y \mid x)$. Provided with a scoring function ϕ that assesses the compatibility between x and y, the distribution can be expressed as the following log-linear model:

$$P(y \mid x) = \frac{e^{\phi(x,y)}}{\sum_{\bar{y}\in\mathcal{Y}} e^{\phi(x,\bar{y})}} \quad (1)$$

To avoid summing over all possible target sentences, the normalization term is approximated by summing over the compatibility score for matching x to $K-1$ sampled negatives together with the compatibility score for the positive candidate:

$$P_{approx}(y \mid x) = \frac{e^{\phi(x,y)}}{\sum_{k=1}^{K} e^{\phi(x,y_k)}} \quad (2)$$

This formulation is similar to early work on discriminative training of log-linear translation decoding models (Och and Ney, 2002). However, rather than using a weighted sum of manually engineered features, we define ϕ to be the dot-product of sentence embeddings for the source, $\mathbf{u}$, and target, $\mathbf{v}$, with $\phi(x, y) = \mathbf{u}^\top \cdot \mathbf{v}$. A similar log-linear sentence embedding based formulation of $P(y|x)$ has been previously used for conversation and e-mail response prediction (Henderson et al., 2017; Yang et al., 2018).

2.2 Bilingual Sentence Embeddings

Bilingual sentence embeddings are obtained using the dual-encoder architecture illustrated in Figure 1. We use Deep Averaging Networks (DANs) (Iyyer et al., 2015) to compute sentence-level embedding vectors by first averaging word and bi-gram level embeddings, denoted as $\Psi(x)$ and $\Psi(y)$, for the source and target sentences, respectively. [1] The word and bi-gram level embeddings are not pretrained but are rather learned during training of the sentence encoders. The averaged representation is provided to a feedforward deep neural network (DNN). Across hidden layers we include residual connections with a skip level of 1. The final bilingual sentence embeddings are $\mathbf{u}$ and $\mathbf{v}$, which are taken from the last layer of the source and target encoders, respectively. The dot-product of the sentence embeddings, $\mathbf{u}^T \cdot \mathbf{v}$, is used to compute the translation score, $\phi(x, y)$.

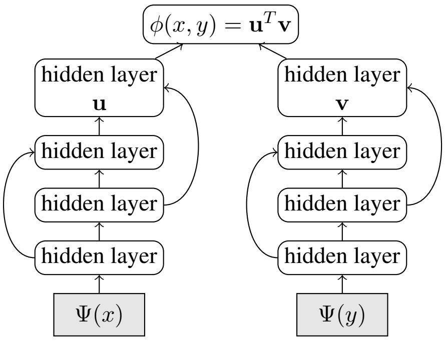

Figure 1: Dual-encoder architecture, where a group of hidden layers encodes source sentence x to $\mathbf{u}$ and a separate group encodes target sentence y to $\mathbf{v}$ such that the score $\phi(x, y)$ is the dot-product $\mathbf{u}^T \cdot \mathbf{v}$.

The dual-encoders are trained for the translation candidate ranking task by maximizing the log likelihood of P_{approx}. This objective is particularly

[1] Our implementation sums the word and bi-gram embeddings and then divides the result by $sqrt(n)$, where n is the sentence length. The intuition behind dividing by $sqrt(n)$ is as follows: We want our input embeddings to be sensitive to length. However, we also want to ensure that, for short sequences, the relative differences in the representations are not dominated by sentence length effects.

	Source (Target)		Negatives
en-fr	How to display and access shared files (Comment afficher et accéder aux fichiers partagés)	**Random**	Sa respiration devient laborieuse Benoit Faucon Lieu London
		Hard	Accès l’environnement des fichiers partagés Des éléments comme des fichiers de dossiers
	The General Delegation for Armaments (La délégation générale pour l’armement)	**Random**	RCS 871, où le juge Fauteux explique Avis sur les hôtels
		Hard	La 9e armée , commandée par le général Foch La délégation militaire hongroise composée de ...
en-es	Oil and gas investments (Inversiones en petróleo y gas)	**Random**	Alquiler mensual desde : 890 USD ¿Qué más se deja para preguntar?
		Hard	Petróleo y gas Petróleo y Gas Petroquímica página
	In Spain, it has clearly chosen the gratuity (En España, se ha elegido claramente la gratuidad)	**Random**	Ve el perfil completo de Fleishman León de montaña en roca
		Hard	Dejar propina es una costumbre chilena Este es un típico restaurante español de España

Table 1: Example of random negatives and hard negatives for en-fr and en-es.

well suited for mini-batch training. As illustrated in Figure 2, within a batch, each source and target translation pair serves as a positive example for that particular pairing with alternative pairings within the same batch treated as negative examples. Given an ordered collection of embeddings for source and target translation pairs, all of the dot-product scores necessary to compute P_{approx} can be determined using a single matrix multiplication of the encoding matrices, $\mathbf{U}$ and $\mathbf{V}^{\top}$.[2] After the matrix multiplication the scores assigned to true translation pairs can be found on the diagonal while the scores for incorrect pairings are off-diagonal.

Within our experiments, models differ in their selection of the $K - 1$ sampled negatives. Our preliminary models make use of the random sampling strategy that has been proven successful in prior work (Henderson et al., 2017; Yang et al., 2018). Using this strategy consists of randomly composing batches of translation pairs and using the matrix multiplication approach described above to obtain within batch negatives for each incorrect pairing We employ random shuffling during training resulting in different random negatives for each $\mathbf{u}_i$ across epochs. As described below we also explore introducing additional hard negatives. This is achieved by extending the target embeddings matrix $\mathbf{V}$ with the sentence embeddings for the hard negatives, which introduces additional off-diagonal values within the matrix of dot-product scores.

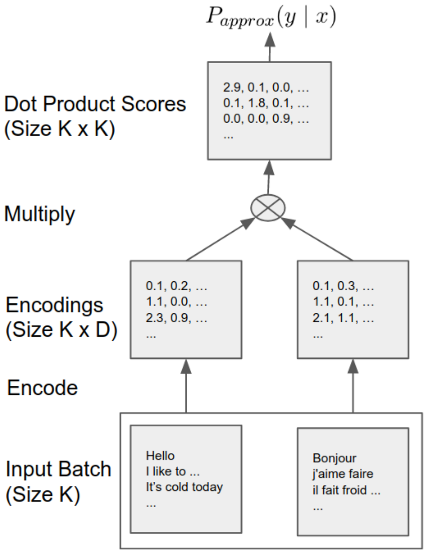

Figure 2: Matrix multiply trick for dot product model with random sampling.

2.3 Semantically Similar Hard Negatives

As illustrated in Table 1, randomly selected negatives result in many pairings that are obviously incorrect without requiring a careful assessment of whether the source and randomly sampled targets are true translations. Within a mini-batch, the model could likely achieve a reasonable level of performance by simply identifying which source and target sentences are on the same topic or are otherwise semantically related. However, when mining for parallel data, extracting sentence pairs that are not translations of each other but that are rather merely topically related is expected to harm downstream MT systems that are trained on the erroneous pairs. Given the increased sensitivity of NMT models to data quality issues, perfor-

[2] The encoding matrices are composed of the ordered sentence embeddings for all of the source and target sentences within a batch, $\mathbf{U} = (\mathbf{u}_0, \mathbf{u}_1, ..., \mathbf{u}_{k-1})$ and $\mathbf{V} = (\mathbf{v}_0, \mathbf{v}_1, ..., \mathbf{v}_{k-1})$.

mance might even be harmed by including semantically similar sentences with sufficient differences in meaning between them. [3]

We improve the mining of true translation pairs by making model training more challenging through the introduction of *hard negatives* – semantically similar translations that are close but not quite identical to the correct translation. The hard negatives are selected using a baseline model trained with randomly sampled negatives. For each source sentence, we identify M hard negatives with target embeddings that achieve high dot-product scores with the source sentence embedding but that are not the correct translation. Examples of hard negatives extracted using the baseline model are provided in Table 1. Compared to the random negatives, hard negatives are semantically more similar to the correct target translation.

As described above, the hard negatives are appended to the target embedding matrix $\mathbf{V}$. Therefore, instead of training with K candidates, each translation input will be compared with $K + K * M$ candidates, where K is the batch size. In practice, getting hard negatives for the entire dataset is very time consuming. We only obtain hard negatives for 20$\%$ of the data and use random negative sampling for the remainder of the training set.

2.4 Mining Parallel Data

One approach to mining parallel data with bilingual sentence-level embeddings is to independently pair individual source and target candidates based on the similarity of their embeddings. Prior work that explored this approach found that the resulting mined sentence pairs produced poor BLEU scores when used for MT training unless they were combined with traditional human translated corpora with known alignments (Schwenk, 2018). We explore both sentence-level and document-level mining of parallel corpora. For document-level mining, we introduce a novel selection criterion that takes into account the confidence of sentence alignments within a document and sentence position information.

2.4.1 Document Matching

Parallel documents are identified as follows: For a given source document, we first run an approximate nearest neighbor (ANN) search for each sentence in the document. This gives us N target sentences for each source sentence (ranked in order of closest match). Let Y be the bag of all target sentences that appear as a match for at least one source sentence. Then for each sentence in Y, we look up the document from which they came. We score each candidate document using Eq 3.[4] This scoring function takes into account the sentence-level nearest neighbor rank of the match for source sentence x to target sentence y in the document being scored, $r(x, y)$. The match rank is linearly combined with a normalized confidence score, $f_1(x, y)$, for the match between x and y as well as the absolute difference between the sentence position index of the source and target sentences, $f_2(x, y)$. The sum of the scoring terms is weighted by the hyperparameters, w_1 and w_2.

$$\sum_{y \in D \cap Y} -r(x,y) + w_1 * f_1(x,y) + w_2 * f_2(x,y) \tag{3}$$

2.4.2 Calibrated Confidence Score

The raw dot product score, $\phi(x, y)$, is a poor choice for the confidence score, $f_1(x, y)$. The score from $\phi(x, y)$ provides a relative metric of a translated sentence's match quality with respect to the source sentence, but it is not a globally consistent measurement of how good a translation pair is. Scores are not necessarily in the same range nor do they have comparable relative values for different input source sentences. As a result, if we choose $\phi(x, y)$ to score confidence, there is no single threshold we can use to filter out bad results.

In order to obtain more consistent confidence scores, we propose a novel score normalization model based on dynamic scaling and shifting of the dot product scores. As illustrated in Figure 3, the dynamic scaling and shifting values are computed from the source embedding, $\mathbf{u}$, and a pointwise squaring of the values within the source embedding, $\mathbf{u}^2$. The vectors $\mathbf{u}$ and $\mathbf{u}^2$ are concatenated. The scale and bias terms are computed as

[3] e.g., adding or removing important details according to the sentence similarity scale proposed by Agirre et al. (2012).

[4] Selecting the target document that appears the most in Y should give us a rough idea of which target document is most likely to be the translation of a source document. However, this approach is quite naive since we are ignoring many pieces of information: 1. The rank at which each target sentence appeared, 2. The dot product score between the target sentence and the source sentence, and 3. The indices of the target sentence and the source sentence (i.e. the position of the sentences within their respective documents). Since the first two factors indicate the model's confidence in the sentence match, it is desirable to incorporate this information into our scoring of document matches.

a weighted sum of the concatenated vectors values. After the dynamic scaling and bias terms are used to calibrate the dot-product score, the resulting calibrated dot-product is passed to a sigmoid in order to obtain a final confidence value between 0 and 1. The weights used to compute the scale and bias terms are trained on held out supervised data.

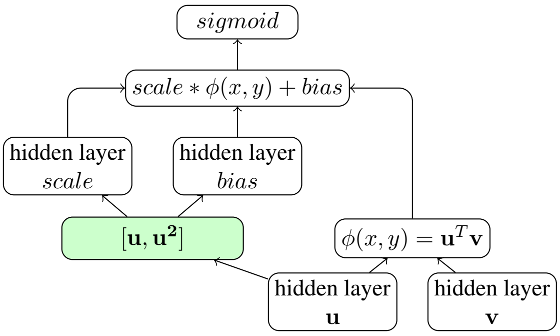
$sigmoid$
$scale * \phi(x,y) + bias$
hidden layer $scale$
hidden layer $bias$
$[\mathbf{u}, \mathbf{u^2}]$
$\phi(x,y) = \mathbf{u}^T\mathbf{v}$
hidden layer $\mathbf{u}$
hidden layer $\mathbf{v}$

Figure 3: Scoring model based on dual-encoder architecture.

It is worth noting that because the hidden layers for $scale$ and $bias$ only use features from the source embeddings, it will not affect the ranking of targets. Thus, we still always use dot-product similarity, $\phi(x,y)$, to retrieve targets via nearest neighbor search. For document-level matching, we convert the dot-product values into the calibrated confidence scores, $f_1(x,y)$, without needing to reinspect the target embeddings.

3 Experiments

We train our proposed model on two language pairs: English-French (en-fr) and English-Spanish (en-es). First, we evaluate the performance on the translation candidate ranking task, comparing the dual-encoder architectures with random negative sampling versus using hard negatives. Then, we present results for document-level matching using Uszkoreit et al. (2010)'s method as a strong baseline. We explore training NMT systems using our method to both filter and re-construct parallel corpora. Finally, we assess the level or agreement between our method and human judgments.

3.1 Data

For training the model, we construct a parallel corpus using a system similar to the approach described in Uszkoreit et al. (2010). The final constructed corpus contains around 600M en-fr sentence pairs and 470M en-es sentence pairs. To assess the quality of the parallel corpus, we ask human annotators to manually evaluate the constructed pairs. The human annotators judge whether 200 randomly selected sentence pairs for both en-fr and en-es are GOOD or BAD translations. We find that the GOOD translation rate is around 80% for both language pairs. The constructed parallel corpus is split into two parts: a training set (90%) and a held-out dev set (10%), with the held-out dev set being used for our preliminary reconstruction experiments.

The UN corpus (Ziemski et al., 2016) is used for additional corpus reconstruction experiments. The corpus consists of 800k manually translated UN documents from 1990 to 2014 for the six official UN languages. 86k of these documents are fully aligned at the sentence-level for all 15 language pairs. We make use of the fully aligned en-fr and en-es document pairs and extract all aligned sentence pairs from those document pairs. There are a total of 11.3 million aligned sentence pairs each for en-fr and en-es. Assuming that we have no knowledge about which documents and sentences are aligned, the task is to reconstruct the document and sentence pairs.

We evaluate trained translation models on wmt13 (Bojar et al., 2013) and wmt14 (Bojar et al., 2014) for en-es and en-fr, respectively. Translation models are trained using data taken from the parallel corpus described above that was constructed using Uszkoreit et al. (2010)'s method. Additional translation experiments make use of ParaCrawl[5], a dataset containing 4 billion noisy translation pairs for en-fr and 2 billion pairs for en-es. Within Paracrawl, each pair contains pre-computed scores by Zipporah (Xu and Koehn, 2017) and the Bicleaner tool, which estimates the translation quality of the pair. We make use of the Zipporah scores to compare translation models trained on filtered versions of the corpus selected using Zipporah versus our method.

3.2 Experimental Configuration

Model configuration and hyperparameters for our sentence embedding models are set mostly based on defaults taken from prior work with very minimal tuning on the held-out dev set. For each language, we build a vocabulary consisting of 200 thousands unigram and 200 thousands bi-gram tokens. All inputs are tokenized and normalized be-

[5] https://paracrawl.eu

fore being fed to the model. We employ an SGD optimizer with a batch size of 128. The learning rate is set to 0.01 with a learning decay of 0.96 every 5 million steps. We train for 50 million steps.

For each encoder layer, we employ a four-layer DNN model which contains 320, 320, 500 and 500 hidden units for each layer respectively. We apply a ReLU activation in the first three layers and no activation in the final layer. We enable residual connections between layers with a skip level of 1. There is no parameter sharing between the source and target encoder layers. The size of the unigram and bi-gram embeddings is set to 320 and the embeddings are updated during the training process. The sentence embedding size is set to 512 for both source and target languages.

The calibrated confidence score is trained jointly with the translation candidate ranking task but with a stop gradient that prevents the confidence task from modifying the bilingual sentence encoders. The tasks are trained in a multitask framework with multiple workers, where 90% of the workers optimize the translation candidate ranking task and the remaining 10% optimize the confidence task. We use the same configuration for confidence as when training the translation candidate ranking task. Both use the same batch size 128, meaning there is 1 positive and 127 negative candidates selected for each pass over an example. We apply a dropout of 0.4 before feeding the feature vector $[\mathbf{u}, \mathbf{u^2}]$ into the hidden layers that calculate $scale$ and $bias$.

3.3 Dev Set Sentence-level Matching

We first evaluate the trained models on the translation target retrieval task and use precision at N (P@N) as our evaluation metric. For every source sentence in the dev set, we run the model and find the nearest neighbors from a set of possible target sentences. Previous work (Henderson et al., 2017; Yang et al., 2018) usually evaluated P@N from 100 examples (1 positive and 99 negatives). We find that this does not work well for the translation target ranking task. Rather, the P@N of 100 metric goes up to 99.9% quickly and provides no differentiation between models trained with different configurations.

In this work, we evaluate the P@N from the true target sentence (positive) and 10 million random selected target (negatives) given a source sentence. We score all selected targets using the translation pair scoring model and rank them accordingly. The P@N score evaluates if the true translation target (positive) is in the top N target candidates. We evaluate the model with random sampling and M hard negatives for M=5, 10, 20. Recall that the number of negatives is equal to the batch size for the models trained with random sampling. The number of negatives for hard negative models, however, is $K + K * M$ where K is the batch size. To make a fair comparison, we also evaluate a model trained with additional random samples, by augmenting the number of random negatives to $K + K * 20$.

Table 2 shows the P@N results of the proposed models for N=1, 3, 10. The model with random negatives provides a strong baseline for finding the right translation target, with a P@1 metric of 70.49% for en-fr and 67.81% for en-es. The augmented random negative model performs better than the base random negative model for en-es. However, the hard negative models outperform the random negative models across all metrics. Even with only 5 hard negatives, the P@1 metrics improved by 8% for en-fr and 3% for en-es. The addition of more hard negatives, however, does not always further improve performance.

4 Reconstructing the United Nations Corpus

In this section, we demonstrate that the proposed model can be used to efficiently reconstruct the United Nations (UN) Parallel Corpus (Ziemski et al., 2016).

4.1 UN Sentence-level Matching

We first apply the dual-encoder model to mine target candidates at the sentence-level. As mentioned in section 1, one of the advantages of the dual-encoder model is that it is straightforward to use it to encode the source and target sentences separately. Taking advantage of this property, we first pre-encode all target sentences into a target database, and then we iterate through the source sentences to retrieve the potential targets for each one of them using an approximated nearest neighbour (ANN) search (Vanderkam et al., 2013). The target sentence retrieval pipeline using ANN search is shown in Figure 4.

Once again we first use P@N as the evaluation metric for target retrieval, for N=1, 3, 10. We evaluate the two random sampling models and a

Negative Selection Approach	en-fr			en-es		
	P@1	P@3	P@10	P@1	P@3	P@10
Random Negatives	70.49	80.03	86.39	67.81	77.37	84.42
Random Negatives (Augmented)	70.67	79.99	86.14	70.47	79.79	86.33
(5) Hard Negatives	78.31	85.30	89.52	73.46	82.37	87.75
(10) Hard Negatives	77.06	84.04	88.70	74.92	83.29	88.14
(20) Hard Negatives	78.29	85.06	89.58	74.84	82.86	88.23

Table 2: Precision at N (P@N) results on the evaluation set for models built using the random negatives and (M) hard negatives. Models attempt to select the true translation target for a source sentence against 10M randomly selected targets.

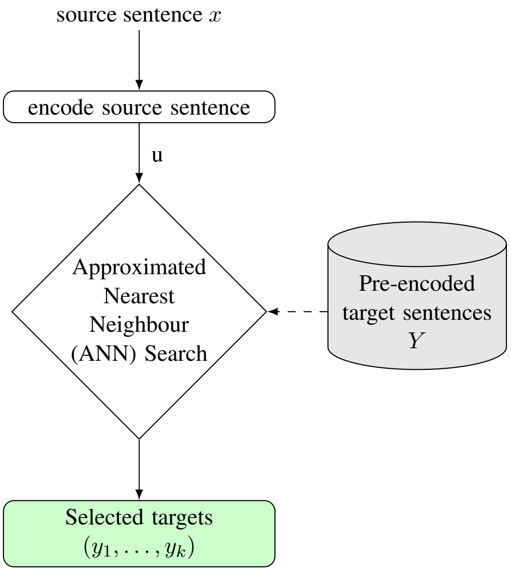
source sentence x
encode source sentence
u
Approximated Nearest Neighbour (ANN) Search
Pre-encoded target sentences Y
Selected targets $(y_1, \ldots, y_k)$

Figure 4: Target sentence retrieval pipeline.

hard negative model with 20 hard negatives for each example. As shown in table 3, with random negatives, the P@1 metric is 34.83% for en-fr and 44.89% for en-es. Adding hard negatives boosts the performance on all metrics, improving the P@1 metric more than 10% absolute in both en-fr and en-es – 48.9% for en-fr and 54.9% for en-es.

4.2 UN Document-level Matching

In our final reconstruction experiment, we make use of the document-level matching method outlined in section 2.4.1. The hyperparameters N, w_1, and w_2 are set to 10, 5, and -2, respectively, based on prior experiments with the translation matching task on the dev set. We compare using the document matching score proposed by Eq. (3) to scoring document pairs by counting the number of Viterbi aligned sentences linking the two together. As a strong baseline, we also include the application of Uszkoreit et al. (2010)'s method to the UN dataset.

Table 4 shows the document matching accuracies. Using Eq. (3) to score document matches outperforms counting mutually aligned sentences. Moreover, while our approach is simpler and less computationally intensive than Uszkoreit et al. (2010)'s, it obtains a promising level of performance.

5 Evaluation Using a Translation Model

As a proof of concept on using our mined translation pairs as training data, we train translation models with original versus mined parallel sentence pairs from UN corpus, and with filtered ParaCrawl data using Zipporah score versus using our model's confidence score. We evaluate on wmt13 (Bojar et al., 2013) and wmt14 (Bojar et al., 2014) testing sets for en-es and en-fr, respectively, with performance assessed using BLEU (Papineni et al., 2002).

The translation models are based on Transformer architecture (Vaswani et al., 2017), and make use of a model dimension of 512 and a hidden dimension of 2048, with 6 layers and 8 attention heads. The models use the Adam optimizer with the training schedule described in Vaswani et al. (2017). For each language pair, sentence pairs are segmented using a shared 32,000 wordpiece vocabulary (Schuster and Nakajima, 2012). Sentence pairs are then batched together by approximate sequence length with variable batch sizes based on sequence length. The average batch size per step is 120 pairs per batch. We train each model until convergence (approximately 120K steps).

Negative Selection Approach	en-fr			en-es		
	P@1	P@3	P@10	P@1	P@3	P@10
Random Negative	34.83	47.99	61.20	44.89	58.13	70.36
Random Negative (Augmented)	36.51	49.07	61.37	47.08	59.55	71.34
(20) Hard Negative	48.90	62.26	73.03	54.94	67.78	78.06

Table 3: Precision at N (P@N) of target sentence retrieval on the UN corpus. Models attempt to select the true translation target for a source sentence from the entire corpus (11.3 million aligned sentence pairs.)

Matching method	en-fr	en-es
Alignment Counts	82.1	85.1
Our approach Eq. (3)	89.0	90.4
Uszkoreit et al. (2010)	93.4	94.4

Table 4: Accuracy of document matching on UN corpus.

5.1 Mined UN Corpus

We compare translation models trained on the reconstructed UN corpora for en-fr and en-es with models trained on the original UN pairs, which we use as Oracle models.

We examine two versions of the reconstructed corpora. In the first version, we take the highest scoring match at the sentence-level as the mined parallel sentence pairs and these pairs are then filtered by their calibrated confidence score[6] with default threshold 0.5. In the second version, we perform document-level matching over the UN dataset. Within paired documents, we follow Uszkoreit et al. (2010) and employ a dynamic programming sentence alignment algorithm informed by sentence length and multilingual probabilistic dictionaries. In both versions, we drop sentence pairs where both sides are either identical or a language detector declares them to be in the wrong language. As a post-processing step, the resulting translations are resegmented using the Moses tokenizer and true-cased before evaluation (Koehn et al., 2007).

Table 5 shows the results obtained from the models trained on the different variations of the parallel data. The models trained with mined pairs perform very close to the Oracle model, demonstrating the effectiveness of the proposed parallel corpus mining approach. Training on the mined sentence-level pairs even does slightly better than using the Oracle data for en-es. This is presumably because the mined pairs are cleaner due to the filtering step. We notice, however, that training on the UN corpus gives translation results that are much lower than the state-of-the-art on the WMT evaluation sets. This is likely due to the fact that the UN parallel corpus is small and drawn from a particularly restricted domain.

	en-fr (wmt14)	**en-es (wmt13)**
Mined sentence-level	29.63	29.03
Mined document-level	30.05	27.09
Oracle	30.96	28.81

Table 5: BLEU scores on WMT testing sets of the NMT models trained on original UN pairs (Oracle) and on two versions of mined UN corpora.

	en-fr (wmt14)	**en-es (wmt13)**
WMT	38.38	32.69
Our data	39.81	33.75
Zipporah	39.29	33.58
WMT + Our data	40.30	34.15
WMT + Zipporah	39.29	34.07

Table 6: BLEU scores on WMT testing sets of the NMT models trained on different data: 1) WMT training sets, 2) filtered ParaCrawl data, and 3) combined data of WMT and filtered ParaCrawl.

5.2 Filtered ParaCrawl data

We compare the performance of training translation models[7] on ParaCrawl data filtered using Zipporah scores versus our scoring method. For this experiment, our confidence score is fine-tuned on the ParaCrawl corpus using an additional 900k positive and 900k negative examples selected based on having extreme Zipporah

[6] The confidence model is trained with a dev set which consist of 1/10 of UN corpora, these data are removed from training.

[7] Using the same model parameters as earlier experiments.

scores.[8] With Zipporah, we select all examples from ParaCrawl with a Zipporah score greater than or equal to 0, which is the threshold used in the official release. There are 43 million such pairs in en-fr and 24 million in en-es. We then select the same number of pairs from the ParaCrawl data that have the highest scores from our fine-tuned model. As illustrated in Table 6, the performance achieved by the ParaCrawl trained models on the WMT test data is quite high, both achieves better performance comparing with the baseline model trained on WMT training set. This suggests that filtered ParaCrawl data is a good source of general-purpose training material. Models trained on our filtered data slightly outperform those trained on data filtered by Zipporah. Row 4 and 5 also show the performance of models trained on the combined data of WMT and our filtered ParaCrawl and combined data of WMT and Zipporah filtered data respectively[9]. Combining the datasets further improves the translation performance about 0.5 blue score, and model trained on WMT and our filtered ParaCrawl data achieves the best performance.

5.3 Qualitative Analysis of Filtered ParaCrawl Data

On the ParaCrawl corpus we find that the Pearson's r between Zipporah and our calibrated confidence scores is only 0.4. This correlation is quiet low given the level of translation performance achieved by both methods when they are used to select training pairs for an NMT system and suggests that the two methods may provide complementary information.

We access the agreement of the two methods on extreme score values.[10] We sample a balanced data set consisting of 100k pairs with extreme positive Zipphora values and 100k pairs with extreme negative values. At a threshold of 0.5 and without an fine-tuning, our method agrees with the extreme Zipporah scores with an accuracy of 78.2% for en-fr and 80.5% for en-es. However, using the confidence scores fine-tuned to ParaCrawl from section 5.2, we achieve a high level of agreement of 98.4% for en-fr and 98.6% with fine-tuning.

	en-fr	en-es
zipporah	72.0	74.0
our model	76.0	74.5

Table 7: GOOD translation rate (%) annotated by translation professionals.

We perform an evaluation using human judgments comparing our scoring model against Zipporah scores on the ParaCrawl data. As in the filtering experiments, we select all examples from ParaCrawl with a Zipporah score greater than or equal to zero and then select a matching number of pairs with the highest scores from our model. We then sample 200 examples from each set and send them to translation professionals for evaluation. Each example is examined by one annotator that labels the pair as either a GOOD or BAD translation. A GOOD translation means more than 70% of a sentence is correctly translated in the paired sentences, meaning most of the information is conveyed.

Table 7 shows the GOOD translation rate for each sampled subset. The performance between the two approaches is close for en-es and the proposed score normalization model is 4% better for en-fr. In our analysis of the BAD translation pairs, one common failure pattern from the proposed model is that one of the sentences is only partially translated in the other sentence. This is likely because we are still missing enough of these types of hard negatives in the training data. We also find our model produces more pairs where the sentences on both sides are identical. These identical pairs are mostly labeled as BAD translations because they are unlikely to be actual translations.

6 Related Work

The problem of obtaining high-quality parallel corpora, or bitexts, is one of the most critical issues in machine translation. One longstanding approach for extracting parallel corpora is to mine documents from the web (Resnik, 1999). Much of the previous work on parallel document mining has relied on using metadata, such as document titles (Yang and Li, 2002), publication dates (Munteanu and Marcu, 2005, 2006) or document structure (Chen and Nie, 2000; Resnik and Smith, 2003; Shi et al., 2006), to identify bitexts.

[8] Extreme positive score values from Zipporah are considered to be those in the top 1% of the agreement scores found in the ParaCrawl corpus. Extreme negative score values are considered to be agreement scores in the bottom 50% of the Zipporah scores for ParaCrawl.

[9] The sizes of WMT training set and filtered ParaCrawl are very close, so we simply mix the data together without any up sampling or down sampling.

[10] For this analysis we use the same definition of extreme Zipporah scores as in section 5.2

Another direction, however, is to identify bitexts using only textual information, as the metadata associated with documents can often be sparse or unreliable (Uszkoreit et al., 2010). Some text-based approaches for identifying bitexts rely on methods such as n-gram scoring (Uszkoreit et al., 2010), named entity matching (Do et al., 2009), and cross-language information retrieval (Utiyama and Isahara, 2003; Munteanu and Marcu, 2005).

There is active research on using embedding-based approaches where texts are mapped to an embedding space in order to determine whether they are bitexts. Grégoire and Langlais (2017) use a Siamese network (Yin et al., 2015) to map source and target language sentences into the same space, then classify whether the sentences are parallel based on labelled data. Hassan et al. (2018) obtain English and Chinese sentence embeddings in a shared space by averaging encoder states from a bilingual shared encoder NMT system. The cosine similarity between these sentence embeddings is then used as a measure of cross-lingual similarity between the sentences, which can then be used to filter out noisy sentence pairs. Schwenk (2018) use a similar approach but learn a joint embedding over nine languages. Our model differs from previous approaches, as it uses a dual-encoder architecture instead of an encoder-decoder architecture. Not only is the dual-encoder architecture is more efficient (Henderson et al., 2017), it also allows us to directly train toward extracting parallel sentences from a collection of candidates.

7 Conclusion

In this paper, we present an effective parallel corpus mining approach using sentence embeddings produced by a bilingual dual-encoder model. The proposed model encodes source sentences and target sentences into sentence embeddings separately and then calculates the dot-product score for these two embedding vectors to assess translation pair quality. We propose the selection of hard negatives that consist of semantically similar sentence pairs that are not translations of each other. Our experiments reveal that using hard negatives improves the ability of our model to identify true translation pairs. We find the proposed method to be useful for both mining and filtering parallel data. Our method compares favorably to Zipporah for filtering, while for mining it provides a lightweight alternative to Uszkoreit et al. (2010)'s method.

Acknowledgement

We thank our teammates from Descartes, Translate and other Google groups for their feedback and suggestions. Special thanks goes to Wei Wang and Melvin Johnson for extracting translation data and discussions on NMT models.

References

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 2012. Semeval-2012 task 6: A pilot on semantic textual similarity. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 385–393, Montréal, Canada. Association for Computational Linguistics.

Alexandra Antonova and Alexey Misyurev. 2011. Building a web-based parallel corpus and filtering out machine-translated text. In *Proceedings of the 4th Workshop on Building and Using Comparable Corpora: Comparable Corpora and the Web*, pages 136–144. Association for Computational Linguistics.

Ondřej Bojar, Christian Buck, Chris Callison-Burch, Christian Federmann, Barry Haddow, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. 2013. Findings of the 2013 Workshop on Statistical Machine Translation. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 1–44, Sofia, Bulgaria. Association for Computational Linguistics.

Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, et al. 2014. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the ninth workshop on statistical machine translation*, pages 12–58.

Houda Bouamor and Hassan Sajjad. 2018. H2@bucc18: Parallel sentence extraction from comparable corpora using multilingual sentence embeddings. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Paris, France. European Language Resources Association (ELRA).

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder. *CoRR*, abs/1803.11175.

Jiang Chen and Jian-Yun Nie. 2000. Parallel web text mining for cross-language ir. In *Content-Based Multimedia Information Access-Volume 1*, pages 62–77. Centre de Hautes Etudes Internationale D'Informatique Documentaire.

Thi-Ngoc-Diep Do, Viet-Bac Le, Brigitte Bigi, Laurent Besacier, and Eric Castelli. 2009. Mining a comparable text corpus for a vietnamese-french statistical machine translation system. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 165–172. Association for Computational Linguistics.

Francis Grégoire and Philippe Langlais. 2017. A deep neural network approach to parallel sentence extraction. *arXiv preprint arXiv:1709.09783*.

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*.

Matthew Henderson, Rami Al-Rfou, Brian Strope, Yun-Hsuan Sung, László Lukács, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, and Ray Kurzweil. 2017. Efficient natural language response suggestion for smart reply. *CoRR*, abs/1705.00652.

Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III. 2015. Deep unordered composition rivals syntactic methods for text classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1681–1691, Beijing, China. Association for Computational Linguistics.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2017. Billion-scale similarity search with gpus. *arXiv preprint arXiv:1702.08734*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 177–180, Stroudsburg, PA, USA. Association for Computational Linguistics.

Dragos Stefan Munteanu and Daniel Marcu. 2005. Improving machine translation performance by exploiting non-parallel corpora. *Computational Linguistics*, 31(4):477–504.

Dragos Stefan Munteanu and Daniel Marcu. 2006. Extracting parallel sub-sentential fragments from non-parallel corpora. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, pages 81–88. Association for Computational Linguistics.

Franz Josef Och and Hermann Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 295–302, Stroudsburg, PA, USA. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Philip Resnik. 1999. Mining the web for bilingual text. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 527–534. Association for Computational Linguistics.

Philip Resnik and Noah A Smith. 2003. The web as a parallel corpus. *Computational Linguistics*, 29(3):349–380.

M. Schuster and K. Nakajima. 2012. Japanese and korean voice search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5149–5152.

Holger Schwenk. 2018. Filtering and mining parallel data in a joint multilingual space. *arXiv preprint arXiv:1805.09822*.

Lei Shi, Cheng Niu, Ming Zhou, and Jianfeng Gao. 2006. A dom tree alignment model for mining parallel data from the web. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, pages 489–496. Association for Computational Linguistics.

Jakob Uszkoreit, Jay M. Ponte, Ashok C. Popat, and Moshe Dubiner. 2010. Large scale parallel document mining for machine translation. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 1101–1109, Stroudsburg, PA, USA. Association for Computational Linguistics.

Masao Utiyama and Hitoshi Isahara. 2003. Reliable measures for aligning japanese-english news articles and sentences. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 72–79. Association for Computational Linguistics.

Dan Vanderkam, Rob Schonberger, Henry Rowley, and Sanjiv Kumar. 2013. Nearest neighbor search in google correlate. Technical report, Google.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Hainan Xu and Philipp Koehn. 2017. Zipporah: a fast and scalable data cleaning system for noisy web-crawled parallel corpora. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2945–2950.

Christopher C Yang and Kar Wing Li. 2002. Mining english/chinese parallel documents from the world wide web. In *Proceedings of the 11th International World Wide Web Conference, Honolulu, USA*.

Yinfei Yang, Steve Yuan, Daniel Cer, Sheng-yi Kong, Noah Constant, Petr Pilar, Heming Ge, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Learning semantic textual similarity from conversations. In *The 3rd Workshop on Representation Learning for NLP (RepL4NLP)*, Melbourne, Australia. Association for Computational Linguistics.

Wenpeng Yin, Hinrich Schütze, Bing Xiang, and Bowen Zhou. 2015. Abcnn: Attention-based convolutional neural network for modeling sentence pairs. *arXiv preprint arXiv:1512.05193*.

Michal Ziemski, Marcin Junczys-Dowmunt, and Bruno Pouliquen. 2016. The united nations parallel corpus v1. 0. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation*, LREC '16. European Language Resources Association.