Self-Supervised Learning for Pairwise Data Refinement

Gustavo Hernández Ábrego, Bowen Liang, Wei Wang, Zarana Parekh,
Yinfei Yang, Yunhsuan Sung
Google Research, Mountain View, CA. USA
{gustavoha,bowenl,wangwe,zarana,yinfeiy,yhsung}@google.com

Abstract

Pairwise data automatically constructed from weakly supervised signals has been widely used for training deep learning models. Pairwise datasets such as parallel texts can have uneven quality levels overall, but usually contain data subsets that are more useful as learning examples. We present two methods to refine data that are aimed at obtaining that kind of subsets in a self-supervised way. Our methods are based on iteratively training dual-encoder models to compute similarity scores. We evaluate our methods on de-noising parallel texts and training neural machine translation models. We find that: (i) The self-supervised refinement achieves most machine translation gains in the first iteration, but following iterations further improve its intrinsic evaluation. (ii) Machine translations can improve the de-noising performance when combined with selection steps. (iii) Our methods are able to reach the performance of a supervised method. Being entirely self-supervised, our methods are well-suited to handle pairwise data without the need of prior knowledge or human annotations.

1 Introduction

Deep learning models are widely adopted and have demonstrated their usefulness in many areas and applications. Despite their diversity, one common characteristic of these models is the large number of parameters that need to be adjusted during training (some recent models that have billions of parameters include T5 (Raffel et al., 2019) and GPT-2 (Radford et al., 2019)). This leads to the need of collecting large amounts of training examples. Pairwise data, that captures the relationship in two modalities, is used to train deep learning models such as Neural Machine Translation (NMT) (Wu et al., 2016), Question Answering (Wang et al., 2007), Image Captioning (Sharma et al., 2018), etc. To train this kind of models, large-scale data can often be obtained from weak signals like text co-occurrence (Yang et al., 2018) or dictionary n-gram matching (Uszkoreit et al., 2010). For example, in the machine translation community, the large amount of multilingual text available on the internet has naturally led to the idea of using internet data to train NMT models (Resnik, 1999). This approach has proven advantageous but it has the drawback that data mined this way is intrinsically noisy (Resnik and Smith, 2003). Despite the poor quality, usually this kind of data contains a helpful subset that can be recovered through a process of data cleaning or refinement. Data cleaning could be implemented with linguistic knowledge such as its script, vocabulary, syntax, etc. Alternatively, a model can be trained on "clean" or "trusted" pairs that are verified through manual annotation. Both options can be highly effective, but the former is limited in scope and error-prone, while the latter can be costly due to the number of required annotated examples.

In this paper we introduce two self-supervised methods to obtain data subsets from noisy pairwise data that can be helpful to train dual-encoder (D-E) and neural machine translation (NMT) models. As noisy pairwise data, in our experiments we use parallel texts mined from the internet. Our methods do not require external knowledge (e.g. syntactic rules), language-dependent heuristics (e.g. script verification) or synthetic positive or negative training examples. By eliminating the need of annotations, our methods directly address the data labelling bottleneck. Our methods employ D-E models (Gillick et al., 2018) to learn a shared embedded space from the co-located text in the sentence pairs mined from the internet. Following Chidambaram et al. (2018) we use the embedding distance in the learned space as a measure of cross-lingual similarity between sentences. Our hypothesis is that

Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, pages 435–446
December 4 - 7, 2020.

if higher scores are associated with cross-lingual similarity, pairs with higher scores will be closer to be actual translations of each other and, in that case, may be part of the data subset useful to train the models.

In our experiments, our methods show effective refining parallel texts mined from the internet. Much of the gains in the downstream evaluation are achieved in the first iteration of the method, but later iterations keep improving the D-E models. Despite being self-supervised, our methods show competitive performance when compared against a de-noising method that uses supervision.

2 Related Work

One line of the research that directly relates to our work is corpus filtering for training NMT models. Below we classify the related work into two categories depending on the amount of supervision needed (e.g. high quality parallel texts).

(Semi-)Supervised Methods Some data de-noising methods simply use filtering rules or heuristics such as language identification of both the source and target texts, vocabulary checks, language model (syntactic) verification, and so on. In contrast to rule-based approaches, approaches like Chen and Huang (2016) and Wang et al. (2018c) train classifiers to distinguish in-domain vs. out-of-domain (or clean vs. noisy) data with a small parallel corpus, while other approaches build reference models on larger amounts of high-quality data (Junczys-Dowmunt, 2018; Defauw et al., 2019). There are approaches that combine rules and heuristics with probabilistic models to determine the amount of noise in each sentence pair. In some cases these systems are designed as targeted efforts to denoise a particular dataset. Bicleaner (Sánchez-Cartagena et al., 2018), in relationship to the ParaCrawl (Esplà et al., 2019) data, is an example of that approach.

Unsupervised Methods In contrast to the supervised methods, unsupervised methods do not require good-quality data to be available. Recent work (Zhang et al., 2020) leverages pre-trained language models and synthetic data (Vyas et al., 2018), in place of true supervision. Some efforts focus on using monolingual corpora and align them through bootstrapping in order to generate sentence pairs (Tran et al., 2020; Ruiter et al., 2020), while others train a model with noisy data directly to generate embeddings and score the data (Chaudhary et al., 2019). Wang et al. (2018b) use *two* NMT models taken from two training epochs to decide which data to use in order to improve the training efficiency and to show a de-noising effect. Our methods here try to take advantages of all of these approaches. Koehn et al. (2018) and Koehn et al. (2019) summarize findings of the WMT corpus filtering efforts, though our work here primarily examines a self-supervised method in the context of de-noising, rather than on a targeted filtering effort.

Our methods are unsupervised. We use dual-encoder models, rather than an encoder-decoder architecture, to model pairwise data and let the model self-supervise itself or, further, be co-trained with an NMT model to refine the training data.

3 Dual-Encoder Model

Dual-encoder (D-E) models have demonstrated to be an effective learning framework applied to both supervised (Henderson et al., 2017; Gillick et al., 2019) and unsupervised tasks (Cer et al., 2018; Chidambaram et al., 2018). A multi-task D-E model consists of two encoders and a combination function for each of the tasks. In the context of the D-E framework, the selection of bilingual text can be interpreted as a ranking problem where, with y_i as the true target of source sentence x_i, $P(y_i|x_i)$ is ranked above all the other target candidates in $\mathcal{Y}$. $P(y_i|x_i)$ can be expressed as a log-linear model but, for practical reasons, we approximate the full set of target candidates $\mathcal{Y}$ with a sample (Henderson et al., 2017). When training in a batch, $P(y_i|x_i)$ can be approximated as:

$$P(y_i|x_i) \approx \frac{e^{\phi(x_i,y_i)}}{e^{\phi(x_i,y_i)} + \sum_{n=1,n\neq i}^{N} e^{\phi(x_i,y_n)}} \quad (1)$$

where N is the size of a batch and ϕ is a similarity function. In such a way, model training can be done by optimizing a log-likelihood loss function:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{e^{\phi(x_i,y_i)}}{e^{\phi(x_i,y_i)} + \sum_{n=1,n\neq i}^{N} e^{\phi(x_i,y_n)}} \quad (2)$$

Based on the results of Yang et al. (2019a) with additive margin softmax (Wang et al., 2018a), we modify our loss function to include margin m:

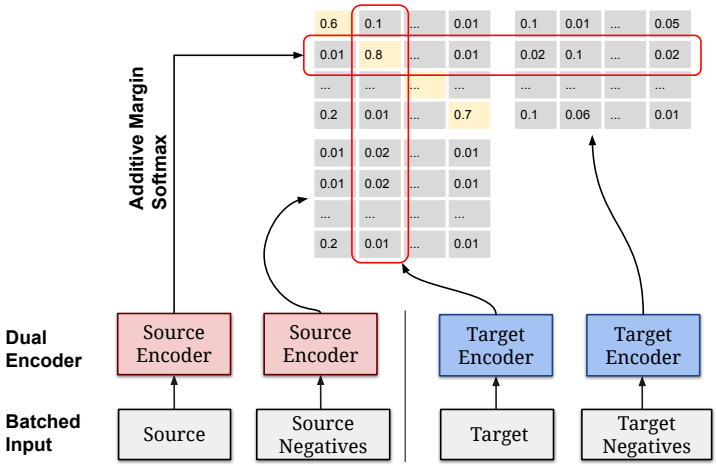


Figure 1: D-E model training with hard negatives. The encoders with the same color share parameters. The dot product scoring function makes it easy to compute pairwise scores by doing matrix multiplications. The highlighted diagonal indicates the dot products of the source and target texts. The additive margin softmax is applied at every row (source→target) and column (target→source).

$$\mathcal{L}_{ams} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{\phi(x_i,y_i)-m}}{e^{\phi(x_i,y_i)-m}+\sum_{n=1,n\neq i}^{N}e^{\phi(x_i,y_n)}} \quad (3)$$

When using the dot product as similarity function ϕ, a single matrix multiplication can be used to efficiently compute scores for all the examples in the batch. When set to learn from clean cross-lingual paired texts, a D-E model can be used to learn strong cross-lingual embeddings for bitext retrieval as shown in Guo et al. (2018) and Yang et al. (2019a). The challenge is to learn similar embeddings when training D-E models on noisy data.

3.1 Model Configuration

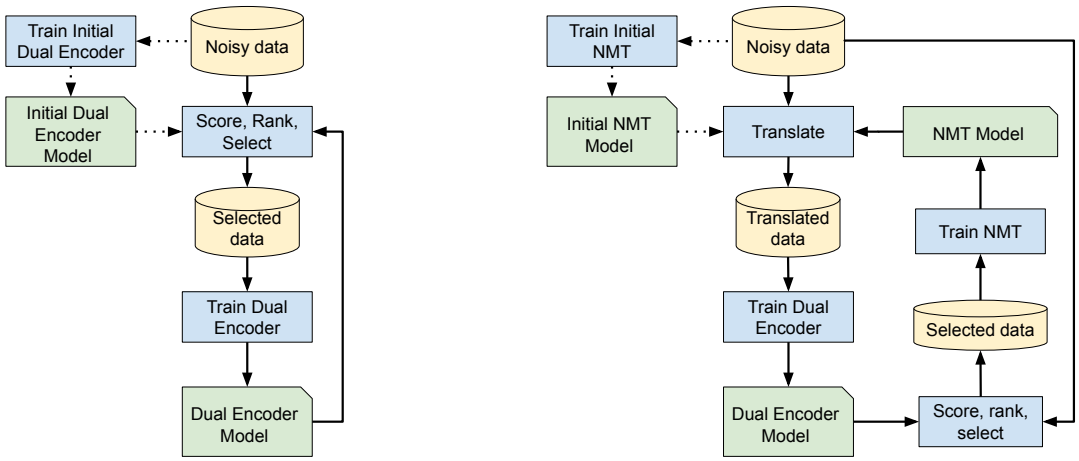
In our experiments we use D-E models with hard negatives sampling (Guo et al., 2018). Similar to Yang et al. (2019a), our models are trained bidirectional so the rankings in both directions, source to target and target to source, are optimized. But in contrast to Yang et al. (2019a) we do not share the parameters between the source and target encoders. In our initial experiments training NMT models we found that, under noisy conditions, there is improvement of close to 1 BLEU point when using D-E models that use specific encoders for each language. Figure 1 illustrates our training approach. For our encoders we use 3-layer transformers (Vaswani et al., 2017) in the encoders with hidden layers of size 512 and 8 attention heads.

We build vocabularies for each language separately. Given the noise in the data, the vocabularies might not include all words in the source or target languages. We control the prevalence of words in the expected language with the vocabulary size. Our reasoning is that large vocabularies are more likely to include words in languages other than the expected. 200k most frequent words are used and 200k extra buckets are reserved for the out-of-vocabulary words found in the training. We use character- and word-level features to model the source and target inputs. For character-level representations, we decompose each word into all character n-grams within a range. For word-level representation, we sum the embeddings for its character n-grams and its word embedding. The final sentence representation is the output of the transformer layers as a 500-dimensional vector. We train the D-E models using SGD for 40M steps with a learning rate of 0.001. A fixed value of margin 0.2 is used in equation 3.

4 Our Approach: Self-Supervised Learning for Data Refinement

4.1 Training with Hard Negatives

As described in equation 1, and illustrated in figure 1, for every source sentence we use all target sentences, except its own, as negatives in a batch. We also augment the batch with hard negatives to improve the contrast between true translations and any other random sentence pairing. We mine the hard negatives using a separate D-E model to retrieve, for every sentence, the top N candidates that are not its counterpart in the pair. It is important



(a) Iterative filtering.

(b) Machine translation iterative filtering.

Figure 2: **[Iterative filtering (IF)]**: the scores of the dual-encoder are used to select the training material for the next model. **[Machine translation iterative filtering (MT-IF)]**: The D-E model is used to score the forward-translations from the NMT model, only the top-ranking sentence pairs are used to train the next NMT model.

to notice that the hard negatives in our method are retrieved, not generated or synthesized. We mine the hard negatives offline from the sentences in the ParaCrawl v1.0 data, or from the translations only when using translations as target sentences. Our negative-mining D-E has DNN layers, instead of transformer ones, with a reduced embedding size (25-dimensional). We mine hard negatives for both the source and target sentences. As shown in figure 1, the hard negatives are specific to each one of the sentence pairs but, when added to the batch, we use them as additional random negatives for all the other source sentences in the batch. We use a batch size of 128 examples and 5 hard-negatives per example. We augment the batch row-wise with hard negatives mined for the target sentences, and column-wise with hard negatives for the source.

In our self-supervised approach, we train D-E models with one dataset and use the models that we train to score the same data. Our hypothesis is that the scores are useful to rank the data in a way that makes it easy to filter out the noise. It is natural to believe that, in principle, a data-model cycle like this may not lead to much improvement because the trained models tend memorize the training data, including the noise. We break this cycle by adding a selection step to the process and avoiding to train the models with the same examples all the time. We propose a self-supervised method for pairwise data refinement based on data "iterative filtering" (*IF*). With this method we refine data that we use to train NMT models. By including the downstream task in our method, we formulate a second method as an extension of the first one. We regard this second method as "machine translation-iterative filtering" (*MT-IF*). Both methods are illustrated in figure 2.

4.2 Iterative Filtering

We use the dot product between source and target embeddings as proxy of cross-lingual similarity. Once we score and select data to train one model, we can use that model to score and select data for the next one in an iterative way. The details of this method are shown in figure 2a and explained in algorithm 1.

We bootstrap this method by training an initial D-E model with all the pre-filtered data. It is important to notice that in each iteration we train the D-E model with a subset of the data (the selected data), but we score the entire set. This allows the method to recover useful data that may have been discarded in earlier iterations.

Algorithm 1 Iterative filtering

```
τ ← selection threshold
D-E = TrainDualEncoder(data)
while D-E improves do
    scored data = Score(data; D-E)
    ranked data = Rank(scored data)
    selected data = Select(ranked data; τ)
    D-E = TrainDualEncoder(selected data)
end while
```

4.3 Machine Translation Iterative Filtering

In this method, the D-E model selects data to train an NMT model, rather than to train another D-E model. The NMT model then produces translations to train the D-E model. This way, the D-E and NMT models boost each other in a "co-training" way. The key to this method is to use the NMT model to generate the training data for the D-E model in order to improve its de-noising capabilities. Algorithm 2 explains this idea and figure 2b illustrates it.

As before, in every iteration the whole dataset is scored and ranked so sentence pairs that ranked low early on can be recovered in later iterations. In principle, forward-translation does not seem to be a good way to generate training data. One can anticipate that the models are prone to mimic the training data, including the noise. Just as in our first method, we break the cycle by adding a selection step based on the D-E scores and using only the top-ranking data to train the next NMT model.

5 Experimental Setup

Machine Translation Model To assess if we can recover useful subsets from noisy data, we train Transformer-Big (Vaswani et al., 2017) NMT models using data refined with our methods. To train the models, we split the source and target texts into pieces using bilingual sentence piece models (Kudo and Richardson, 2018) that were trained with the ParaCrawl v1.0 data only. We train for a maximum of 200k steps using (Shazeer and Stern, 2018) and pick the best checkpoint according to the performance on a validation set. The models are trained on Google's Cloud TPU v3 with batch size 3072. In all our experiments, the configuration of the NMT models is kept the same with the only difference being the training data.

Algorithm 2 Machine translation iterative filtering

```
τ ← selection threshold
NMT = TrainNMT(data)
while D-E improves or NMT improves do
    translated data = Translate(data; NMT)
    D-E = TrainDualEncoder(translated data)
    scored data = Score(data; D-E)
    ranked data = Rank(scored data)
    selected data = Select(ranked data; τ)
    NMT = TrainNMT(selected data)
end while
```

	en-fr	en-de
All sentence pairs	4,235 M	4,591 M
Pre-filtered	289 M	282 M
70th percentile (for NMT)	87 M	85 M
80th percentile (for D-E)	58 M	56 M

Table 1: Number of sentence pairs in the ParaCrawl v1.0 data, and after prefiltering and selection.

Data In our experiments we use two language pairs: English to French (en-fr) and English to German (en-de). We use ParaCrawl v1.0 (Esplà et al., 2019) as training data. We apply light-weight prefiltering steps to remove sentence pairs that: (i) are duplicated, (ii) have identical source and target texts, (iii) have empty sentences, or (iv) have a large difference in the number of tokens. For the last case, we compute the ratio of source over target tokens as: $\rho = \frac{n_S+\alpha}{n_T+\alpha}$ with n_S and n_T being the number of tokens in the source and in the target respectively, and α a token count tolerance. With an α of 15, we discard a sentence pair if ρ is greater than 1.5. Similarly for the ratio of target over source tokens. We use WMT newstest 2012-2013 (Bojar et al., 2014) as the development set and we evaluate on two sets: WMT newstest 2014 and news discussion test 2015 for en-fr; WMT newstest 2014 and 2015 for en-de.

Evaluation As described in section 3, we trained the D-E models as rankers. Thus, we use the BUCC 2018 mining task (Zweigenbaum et al., 2018) as an intrinsic metric for the model. The task data consists of corpora for four language pairs including fr-en and de-en. For each language pair, the shared task provides a monolingual corpus for each language and a ground truth list containing true translation pairs. The task is to construct a list of translation pairs from the monolingual corpora, and evaluate them in terms of the F1 compared to the ground truth.

To test the end-performance of the NMT models in terms BLEU scores, we compute the detokenized and case-sensitive BLEU scores against the original references using an in-house reimplementation of the `mteval-v14.pl` script .

Iterative Selection In our experiments we ran 3 iterations of the *IF* method and 3 iterations of the *MT-IF* one.

To define the value of the selection thresholds, we conducted initial experiments to explore the

impact of the threshold when selecting the data to train the D-E models. Figure 3 shows the BUCC results, in terms of the best F1 measure and the area under the precision-recall curve (AUCPR), for D-E models trained with data selected using different thresholds. Even though there is not a single threshold that works best for both languages, models trained with data selected from the 70th or 80th percentiles produce the best results. Using either very low (below 0.2) or very high thresholds (above 0.95) leads to D-E models with lower results. We set the selection thresholds for the data to train the D-E models and to train the NMT models separately. For the former we use data on the 80th percentile, and on the 70th percentile for the latter. Our intuition was that we can be more stringent when selecting data to train the D-E because only high-ranking examples may be true translations to learn from. Table 1 shows the number of sentences in the ParaCrawl v1.0 en-fr and en-de datasets and the amount of sentences that the pre-filtering and selection steps, at the different thresholds, let through. The large number of sentence pairs that are eliminated via pre-filtering give an indication of how much noise there is in the data. It is worth noticing that the subset of data that we deem "useful" is two orders of magnitude smaller than the original data.

6 Results

6.1 Intrinsic Dual-Encoder Evaluation

Table 2 shows the BUCC mining task results for the D-E models trained with our methods in terms of F1 and AUCPR. As baseline we include the results of a D-E model trained with all the ParaCrawl v1.0 data after pre-filtering. The baseline performs poorly in both en-fr and en-de. The D-E models trained with the *IF* data produce good mining results starting from the very initial models, i.e. when using D-E models trained using hard negatives but no selection yet. The significant gains of IF_0 over the baseline confirm our observations about the positive impact of hard negatives in cross-lingual tasks (Guo et al., 2018). In subsequent iterations (indices 1 to 3 in table 2) selection is used and the D-E models show steady improvement. The improvement in the AUCPR and F1 of the D-E models trained with the *MT-IF* data is quite remarkable. The performance for models trained with data from the first iteration of this method surpass the performance of models trained with the the third iteration of the *IF* data and keep improving, but seem to plateau around the second iteration. For reference, we include in table 2 the AUCPR and F1 from embeddings generated with the public "universal-sentence-encoder-multilingual-large" v2 (Yang et al., 2019b) from TFHub[1] to show the performance of a D-E model trained on multiple large and non-public industry datasets. As expected, training on this kind of data is far better than de-noising, but the evaluation shows that our methods do a good job refining data, especially considering how much noise there is in the ParaCrawl datasets to start with.

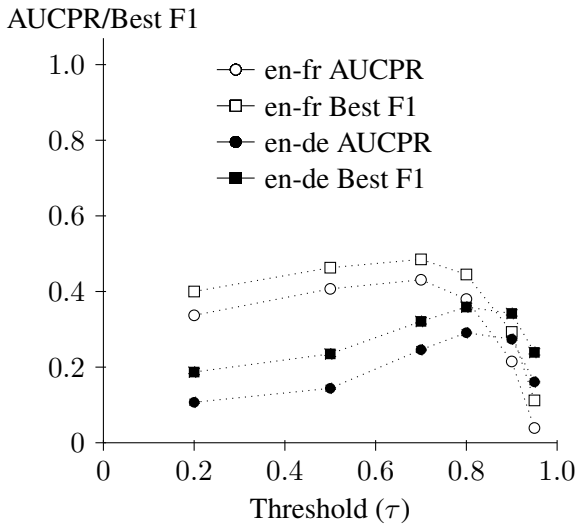


Figure 3: BUCC mining results of dual encoder models trained with data selected at different thresholds.

6.2 Translation Evaluation

To illustrate the end-performance of our methods, table 3 shows the BLEU scores (Papineni et al., 2002) of NMT models trained with data subsets selected with our methods. The D-E models used to score the data in each iteration correspond to the same models reported in table 2. As baseline we use an NMT model trained with all the sentence pairs just after pre-filtering, i.e. selection is not used yet. For both our methods the NMT models show considerable improvement over the baseline. It is interesting that the initial NMT (IF_0 in table 3), shows good improvement in spite of using a D-E whose only difference over baseline is the use of hard negatives. There is also noticeable improvement between the IF_0 and IF_1 results pointing to the fact that our process of scoring, ranking and selection is also useful to improve the

[1] https://tfhub.dev/google/universal-sentence-encoder-multilingual-large/2

Method	en-fr		en-de	
	AUCPR	Best F1	AUCPR	Best F1
Pre-filtered data (baseline)	0.068	0.149	0.020	0.069
IF_0	0.246	0.330	0.094	0.179
IF_1	0.380	0.445	0.291	0.359
IF_2	0.570	0.600	0.372	0.415
IF_3	0.622	0.642	0.390	0.432
$MT\text{-}IF_1$	0.641	0.673	0.545	0.566
$MT\text{-}IF_2$	0.664	0.697	**0.600**	**0.620**
$MT\text{-}IF_3$	**0.676**	**0.707**	0.593	0.608
USE multi-lingual	0.824	0.812	0.861	0.815

Table 2: BUCC mining results of the dual-encoder models. The index in each experiment denotes the iteration. The USE multi-lingual model was trained using non-public industry datasets.

Method	en-fr		en-de	
	newstest2014	newsdiscusstest2015	newstest2014	newstest2015
Pre-filtered data (baseline)	0.303	0.297	0.196	0.239
IF_0	0.324	0.315	0.237	0.276
IF_1	0.342	0.343	**0.239**	0.281
IF_2	0.340	**0.352**	0.237	0.279
IF_3	0.342	0.348	0.236	0.283
Forward-translated data	0.305	0.306	0.203	0.243
$MT\text{-}IF_1$	0.342	0.346	0.237	0.280
$MT\text{-}IF_2$	0.343	0.348	0.235	0.283
$MT\text{-}IF_3$	**0.346**	0.349	0.236	**0.286**

Table 3: BLEU scores of the trained NMT models and the baseline models. The index in each experiment denotes the iteration.

NMT models. The second half of table 3 shows the BLEU scores when the NMT models are added to the refinement process in the *MT-IF* method. For a better reference, we train an NMT model with forward-translated sentence pairs using the baseline NMT model. Crucially, there is no selection on the forward-translated data to train this model. This NMT model does not show improvement relative to the baseline NMT model and confirms that distilling new training examples from forward translations provides little or no gain. In contrast to the BUCC evaluation from table 2, the downstream task does not seem to require several iterations to show good results. The BLEU scores of later iterations in the process only improve marginally as opposed to the steady improvement observed in the BUCC task.

6.3 Supervised vs Self-Supervised

We use Bicleaner to compare our methods against a supervised approach on the task of de-noising the ParaCrawl data, with the important caveat that

	en-fr	en-de
70th percentile after lang ID	29 M	27 M
Bicleaner v1.2	25 M	17 M

Table 4: Number of sentence pairs of selected data after language identification and in Bicleaner v1.2.

Bicleaner is not only supervised but tailored to de-noise this data. In that sense, our method would be in disadvantage especially because our D-E models were not trained with any signal related to the identity of the language. To add this missing component to our method, we use language identification as a post-processing step on the refined data. We use a pre-trained language identification method from Zhang et al. (2018) to filter out pairs where the source or target texts do not match the expected language. As around 30% of the training data gets discarded (table 1 vs table 4), the scores of the remaining data need to be re-ranked in preparation for the selection step. We train new NMT models using only the sentence pairs that get ranked in the

70th percentile and filtered by the language identification. We compare the models against similar NMT models trained with the Bicleaner v1.2 data downloaded from the ParaCrawl website[2]. Table 4 shows the number of sentence pairs used to train the NMT models after applying language identification and in the Bicleaner v1.2 data.

To isolate the effects of language identification, we compare NMT models trained with data from our methods against similar models trained with data that went through language identification also but, as in previous baselines, no selection was used.

As shown in Table 5, using language identification on the training data boosts the performance. The NMT models trained with the data refined with our methods still show considerable improvement over not using selection, making evident that there is still much room for data refinement after language identification. Our method shows very competitive results against the NMT models trained using the Bicleaner v1.2 data, surpassing the BLEU scores in en-fr and getting very similar performance in en-de. It is interesting that, with the addition of language identification, our self-supervised method can remove noise just as effectively as a targeted effort to denoise the ParaCrawl data.

6.4 Iterative Data Refinement

To verify the effectiveness of our methods in finding useful subsets contained in the noisy data, we analyze the results of our models when scoring true sentence pairs versus scoring pairs that are not actual translations. For this analysis, we leverage the BUCC mining task and compute the dot products of "ground truth" pairs using our D-E models. Figure 4 shows box plots of the dot products for both en-fr and en-de BUCC data. For reference, we compute the dot products of the "nearest negative" of each source sentence. We reuse the retrieval results from the D-E intrinsic evaluation (subsection 6.1) to define the nearest negative as the target sentence with the highest dot product that is not its actual translation. This leads to 9,086 ground truth and nearest negative dot products for en-fr and 9,580 for en-de whose distributions are displayed in the box plots in figure 4. Starting with the baseline D-E models, the dot products of the ground truth and the nearest negative are very close in value. This is evident by the fact that their difference (also plotted in figure 4) is very close to 0. The difference starts to grow with the IF_0 models, showing that hard-negatives are useful to increase the separation between the dot products of both classes. For the IF method, the difference between ground truth and nearest negative keeps growing steadily with every iteration. This confirms the progression observed in the AUCPR and F1 measures in table 2. For the $MT\text{-}IF$ models, the score difference between ground truth and nearest negative is already significant in the first iteration, but it does not progress much further in later iterations. This also confirms the observations for these models in the BUCC mining results from table 2. The fact that the dot products of our models show good levels of separation of each class corroborate, from the data analysis standpoint, that both our methods are effective in separating useful data samples from the noisy dataset.

[2]https://paracrawl.eu/v1

6.5 Discussion

Intrinsic vs downstream evaluations Our self-supervised methods seem to naturally improve the quality of the refined data, as measured by the results of the BUCC parallel text mining task. However, most of the BLEU score gains are achieved on the first iteration. One possible explanation is that the BUCC evaluation is a closer match to the ranking task used to train the D-E model. Another possibility is that, given that different sequences can produce the same BLEU scores, there may be improvements in the translation quality that the BLEU scores do not reflect. Making the method more aware of the downstream translation task and gaining insight into the translation quality are interesting lines of future work.

Language identification impact In noisy data, language identification seems to play a significant role. In our experiments we applied it as a post-process but we are interested in applying it as part of the pre-filtering process, or integrated as part of our scoring models in the future.

Breaking the data-model memorization cycle Training NMT models directly with translated data did not produce gains over the baseline. But we found significant gains when instead we used the translated data to train D-E models and used the models to score and select data to in turn train the NMT models. We see this as confirmation that it is possible to break the data-model memorization cycle by co-training models using different training goals.

Method	en-fr		en-de	
	newstest2014	newsdiscusstest2015	newstest2014	newstest2015
Pre-filtered data lang ID	0.336	0.346	0.239	0.279
Bicleaner v1.2 data	0.363	0.370	**0.274**	**0.316**
IF_1	**0.369**	**0.373**	0.263	0.306
IF_2	**0.369**	0.369	0.267	0.308
IF_3	0.366	0.372	0.269	0.314
$MT\text{-}IF_1$	0.361	0.365	0.263	0.308
$MT\text{-}IF_2$	0.363	0.370	0.262	0.303
$MT\text{-}IF_3$	0.360	0.364	0.259	0.303

Table 5: BLEU scores of the NMT models using language identification and compared against Bicleaner.

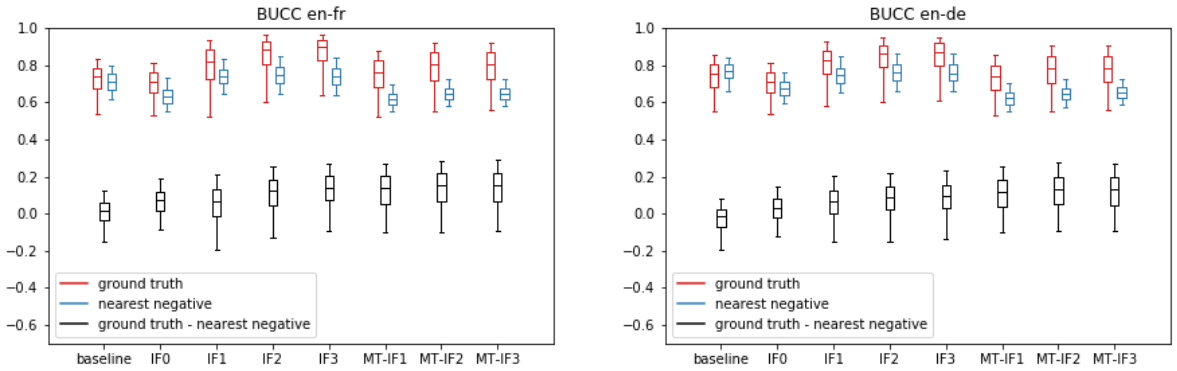


Figure 4: Dot product distributions for the ground truth and nearest negative from the BUCC mining task. The box plots represent the (5,25,50,75,95)-percentile of the dot product distribution for each method and iteration.

7 Conclusions

We introduced two self-supervised methods to refine pairwise data aimed at selecting useful subsets from noisy data. In our experiments we used parallel texts mined from the internet as example of the weakly constructed pairwise data to refine. Our methods do not require linguistic knowledge or human annotated data. They use iterative selection of the data to train two kinds of models. Our first method is based on self-boosting dual-encoder models iteratively. We applied this method to denoise data to train NMT models. Our second method integrates the NMT models into the iterative process to generate translations that, after a selection step, are used to train the dual-encoder models. Our results show that most of the gains in terms of BLEU score can be achieved in the first iteration of our methods, but later iterations keep improving the performance of the dual-encoder models in the BUCC evaluation. In our experiments, using translated text in combination with a selection step helped to improve the de-noising capabilities of the dual-encoder models. We observed that selection is effective to break the model-data memorization cycle. One characteristic that our self-supervised methods do not seem to capture well is an indication of the language identity. If we use language identification on the denoised data as a post-processing step, the performance, in terms of BLEU scores, turns very competitive against supervised targeted efforts tailored to remove noise from the dataset. These results encourage us to pursue future lines of work that include using cross-attention in the pairwise data to better capture the relationship in the pairs. Also, specific to parallel sentences mined from the internet, we would like to explore ways to include language identification in the models. On the other hand, it seems natural to leverage the self-supervision characteristics of our methods and apply them to language pairs where noisy internet data may be available but annotated data is not. Lastly, we are interested in expanding our methods to other pairwise data such as text-image pairs.

Acknowledgments

The authors would like to thank Noah Constant and the anonymous reviewers for their valuable feedback and suggestions to improve this paper.

References

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.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169–174, Brussels, Belgium. Association for Computational Linguistics.

Vishrav Chaudhary, Yuqing Tang, Francisco Guzmn, Holger Schwenk, and Philipp Koehn. 2019. Low-resource corpus filtering using multilingual sentence embeddings. *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*.

Boxing Chen and Fei Huang. 2016. Semi-supervised convolutional networks for translation adaptation with tiny amount of in-domain data. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning (CoNLL)*, pages 314–323.

Muthuraman Chidambaram, Yinfei Yang, Daniel Cer, Steve Yuan, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Learning cross-lingual sentence representations via a multi-task dual-encoder model. *CoRR*, abs/1810.12836.

A. Defauw, Sara Szoc, Anna Bardadym, Joris Brabers, Frederic Everaert, Roko Mijic, K. Scholte, Tom Vanallemeersch, Koen Van Winckel, and J. V. D. Bogaert. 2019. Misalignment detection for web-scraped corpora: A supervised regression approach. *Informatics*, 6:35.

Miquel Esplà, Mikel Forcada, Gema Ramírez-Sánchez, and Hieu Hoang. 2019. ParaCrawl: Web-scale parallel corpora for the languages of the EU. In *Proceedings of Machine Translation Summit XVII Volume 2: Translator, Project and User Tracks*, pages 118–119, Dublin, Ireland. European Association for Machine Translation.

Daniel Gillick, Sayali Kulkarni, Larry Lansing, Alessandro Presta, Jason Baldridge, Eugene Ie, and Diego Garcia-Olano. 2019. Learning dense representations for entity retrieval. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 528–537, Hong Kong, China. Association for Computational Linguistics.

Daniel Gillick, Alessandro Presta, and Gaurav Singh Tomar. 2018. End-to-end retrieval in continuous space. *arXiv preprint arXiv:1811.08008*.

Mandy Guo, Qinlan Shen, Yinfei Yang, Heming Ge, Daniel Cer, Gustavo Hernández Ábrego, Keith Stevens, Noah Constant, Yun-hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Effective parallel corpus mining using bilingual sentence embeddings. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 165–176. Association for Computational Linguistics.

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.

Marcin Junczys-Dowmunt. 2018. Dual conditional cross-entropy filtering of noisy parallel corpora. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 901–908, Belgium, Brussels. Association for Computational Linguistics.

Philipp Koehn, Francisco Guzman, Vishrav Chaudhary, and Juan Pino. 2019. Findings of the wmt 2019 shared task on parallel corpus filtering for low-resource conditions. In *Proceedings of the Fourth Conference on Machine Translation*, Florence, Italy. Association for Computational Linguistics.

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*, Belgium, Brussels. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. 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.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv e-prints*.

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.

Dana Ruiter, Cristina España-Bonet, and Josef van Genabith. 2020. Self-induced curriculum learning in neural machine translation. *arXiv preprint arXiv:2004.03151*.

Víctor M. Sánchez-Cartagena, Marta Bañón, Sergio Ortiz-Rojas, and Gema Ramírez-Sánchez. 2018. Prompsit's submission to wmt 2018 parallel corpus filtering shared task. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, Brussels, Belgium. Association for Computational Linguistics.

Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. 2018. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565, Melbourne, Australia. Association for Computational Linguistics.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. *arXiv preprint arXiv:1804.04235*.

Chau Tran, Yuqing Tang, Xian Li, and Jiatao Gu. 2020. Cross-lingual retrieval for iterative self-supervised training. *arXiv preprint arXiv:2006.09526*.

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.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Yogarshi Vyas, Xing Niu, and Marine Carpuat. 2018. Identifying semantic divergences in parallel text without annotations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1503–1515, New Orleans, Louisiana. Association for Computational Linguistics.

Feng Wang, Jian Cheng, Weiyang Liu, and Haijun Liu. 2018a. Additive margin softmax for face verification. *IEEE Signal Processing Letters*, 25(7):926–930.

Mengqiu Wang, Noah A. Smith, and Teruko Mitamura. 2007. What is the Jeopardy model? a quasi-synchronous grammar for QA. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 22–32, Prague, Czech Republic. Association for Computational Linguistics.

Rui Wang, Masao Utiyama, and Eiichiro Sumita. 2018b. Dynamic sentence sampling for efficient training of neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 298–304, Melbourne, Australia. Association for Computational Linguistics.

Wei Wang, Taro Watanabe, Macduff Hughes, Tetsuji Nakagawa, and Ciprian Chelba. 2018c. Denoising neural machine translation training with trusted data and online data selection. In *Proceedings of the Third Conference on Machine Translation*, pages 133–143. Association for Computational Linguistics.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.

Yinfei Yang, Gustavo Hernández Ábrego, Steve Yuan, Mandy Guo, Qinlan Shen, Daniel Cer, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2019a. Improving multilingual sentence embedding using bi-directional dual encoder with additive margin softmax. *CoRR*, abs/1902.08564.

Yinfei Yang, Daniel Cer, Amin Ahmad, Mandy Guo, Jax Law, Noah Constant, Gustavo Hernández Ábrego, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2019b. Multilingual universal sentence encoder for semantic retrieval. *arXiv preprint arXiv:1907.04307*.

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 *Proceedings of The Third Workshop on Representation Learning for NLP*, pages 164–174, Melbourne, Australia. Association for Computational Linguistics.

Boliang Zhang, Ajay Nagesh, and Kevin Knight. 2020. Parallel corpus filtering via pre-trained language models. *arXiv:2005.06166*.

Yuan Zhang, Jason Riesa, Daniel Gillick, Anton Bakalov, Jason Baldridge, and David Weiss. 2018. A fast, compact, accurate model for language identification of codemixed text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 328–337, Brussels, Belgium. Association for Computational Linguistics.

Pierre Zweigenbaum, Serge Sharoff, and Reinhard Rapp. 2018. Overview of the third bucc shared task: Spotting parallel sentences in comparable corpora. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Paris, France. European Language Resources Association (ELRA).