NRC Parallel Corpus Filtering System for WMT 2019

Gabriel Bernier-Colborne
NRC-CNRC
National Research Council Canada
2107, chemin de la Polytechnique
Montreal, Quebec H3T 1J4, Canada
Gabriel.Bernier-Colborne@nrc-cnrc.gc.ca

Chi-kiu Lo
NRC-CNRC
National Research Council Canada
1200 Montreal Road
Ottawa, Ontario K1A 0R6, Canada
Chikiu.Lo@nrc-cnrc.gc.ca

Abstract

We describe the National Research Council Canada team's submissions to the parallel corpus filtering task at the Fourth Conference on Machine Translation.

1 Introduction

The WMT19 shared task on parallel corpus filtering was essentially the same as last year's edition (Koehn et al., 2018b), except under low-resource conditions: the language pairs were Nepali-English and Sinhala-English instead of German-English, and the data participants were allowed to use was constrained. The aim of the challenge was to identify high-quality sentence pairs in a noisy corpus crawled from the web using ParaCrawl (Koehn et al., 2018a), in order to train machine translation (MT) systems on the clean data. Specifically, participating systems must produce a score for each sentence pair in the test corpora, this score indicating the quality of that pair. Then samples containing 1M or 5M words would be used to train MT systems. Participants were ranked based on the performance of these MT systems on a test set of Wikipedia translations (Guzmán et al., 2019), as measured by BLEU (Papineni et al., 2002). Participants were provided with a few small sources of parallel data, covering different domains, for each of the two low-resource languages, as well as a third, related language, Hindi (which uses the same script as Nepali). The provided data also included much larger monolingual corpora for each of the four languages (en, hi, ne, si).

Cleanliness or quality of parallel corpora for MT systems is affected by a wide range of factors, e.g., the parallelism of the sentence pairs, the fluency of the sentences in the output language, etc. Previous work (Goutte et al., 2012; Simard, 2014) showed that different types of errors in the parallel training data degrade MT quality in different ways.

Intuitively, cross-lingual semantic textual similarity is one of the most important properties of high-quality sentence pairs. Lo et al. (2016) scored cross-lingual semantic textual similarity in two ways, either using a semantic MT quality estimation metric, or by first translating one of the sentences using MT, and then comparing the result to the other sentence, using a semantic MT evaluation metric. At last year's edition of the corpus filtering task, Lo et al. (2018)'s supervised submissions were developed in the same philosophy using a new semantic MT evaluation metric, YiSi-1.

This year, the National Research Council (NRC) Canada team submitted 4 systems to the corpus filtering task, which use different strategies to evaluate the parallelism of sentence pairs. Two of these systems exploit the quality estimation metric YiSi-2, the third uses a deep Transformer network (Vaswani et al., 2017), and the fourth is an ensemble combining these approaches.

In this paper, we describe the 4 systems we submitted, which have three main components: pre-filtering rules, sentence pair scoring, and reranking to improve vocabulary coverage. The systems vary in the way they score sentence pairs. Official results indicate our best systems were ranked 3rd or 4th out of over 20 submissions in most test settings, the ensemble system providing the most robust results.

2 System architecture

There are a wide range of factors that determine whether a sentence pair is good for training MT systems. Some of the more important properties of a good training corpus include:

- High parallelism in the sentence pairs, that constitutes translation adequacy.

- High fluency and grammaticality, especially for sentences in the output language, that constitutes translation fluency.

- High vocabulary coverage, especially in the input language, which should help make the translation system more robust.

- High variety of sentence lengths, which should also improve robustness.

The systems developed by the NRC exploit different strategies to identify a set of sentence pairs that has these properties. The four systems shared the same pipeline architecture:

1. Initial filtering to remove specific types of noise

2. Sentence pair scoring

3. Re-ranking to improve vocabulary coverage

The difference between our 4 submissions is in the way sentence pairs were scored. We used YiSi-2 for two of our submissions, a deep Transformer network exploiting transfer learning for the third, and an ensemble that combines scores from YiSi-2 and several deep Transformer networks.

2.1 Initial filtering

The pre-filtering steps of our submissions are mostly the same as those in Lo et al. (2018). We remove: 1) duplicates after masking email, web addresses and numbers, 2) the majority of number mismatches, 3) sentences in the wrong language according to the pyCLD2 language detector[1] and 4) long sentences (either side has more than 150 tokens).

An additional pre-filtering rule included in this year's submissions is the removal of pairs where over 50% of the Nepali/Sinhalese text is comprised of English, numbers or punctuation.

2.2 Sentence pair scoring

We experimented with different strategies to score sentence pairs. These are described in the following subsections.

2.2.1 YiSi-2: cross-lingual semantic MT evaluation metric

YiSi[2] is a unified semantic MT quality evaluation and estimation metric for languages with different levels of available resources. YiSi-1 measures the similarity between a machine translation and human references by aggregating weighted distributional (lexical) semantic similarities, and optionally incorporating shallow semantic structures.

YiSi-2 is the bilingual, reference-less version, which uses bilingual word embeddings to evaluate cross-lingual lexical semantic similarity between the input and MT output. While YiSi-1 successfully served in the WMT2018 parallel corpus filtering task, YiSi-2 showed comparable accuracy on identifying clean parallel sentences on a hand-annotated subset of test data in our internal experiments (Lo et al., 2018).

Like YiSi-1, YiSi-2 can exploit shallow semantic structures as well. However, there is no semantic role labeler for Nepali/Sinhalese readily available off-the-shelf, thus the version of YiSi-2 used in this work is purely based on cross-lingual lexical semantic similarity. In addition, instead of evaluating through the bag of trigrams to reward the same word order between the two sentences as in YiSi-1, YiSi-2 evaluates through the bag of unigrams to allow reordering between the two sentences in the two languages. Here is a simplified version of YiSi without using shallow semantic structures and bag of n-grams (it is the same as the original version of YiSi (Lo, 2019) with the hyperparameter β set to 0 and n to 1):

$$
\begin{aligned}
&v(u) = \text{embedding of unit } u \\
&w(u) = idf(u) = log(1 + \tfrac{|\mathbb{U}|+1}{|\mathbb{U}_{\exists u}|+1}) \\
&s(e,f) = cos(v(e), v(f))
\end{aligned}
$$

$$
s_p(\overrightarrow{e}, \overrightarrow{f}) = \frac{\sum_a \max_b w(e_a) \cdot s(e_a, f_b)}{\sum_a w(e_a)}
$$

$$
s_r(\overrightarrow{e}, \overrightarrow{f}) = \frac{\sum_b \max_a w(f_b) \cdot s(e_a, f_b)}{\sum_b w(f_b)}
$$

$$
\begin{aligned}
&\text{precision} = s_p(\overrightarrow{e_{\text{sent}}}, \overrightarrow{f_{\text{sent}}}) \\
&\text{recall} = s_r(\overrightarrow{e_{\text{sent}}}, \overrightarrow{f_{\text{sent}}})
\end{aligned}
$$

$$
\text{YiSi} = \frac{\text{precision} \cdot \text{recall}}{\alpha \cdot \text{precision} + (1-\alpha) \cdot \text{recall}}
$$

$$
\text{YiSi-2} = \text{YiSi}(E{=}\text{NE/SI}, F{=}\text{EN})
$$

[1] https://github.com/aboSamoor/pycld2

[2] YiSi is the romanization of the Cantonese word 意思 ('meaning').

model	training data				dict.	size
	lang.	domain	#sent	#word	#pair	#vocab
supervised	ne	IT and religious	563k	8M	—	34k
	en			5M		46k
	si	IT and subtitles	647k	6M		43k
	en			5M		33k
unsupervised	ne	wiki	92k	5M	9k	55k
	en	news	779M	13B		3M
	si	wiki	156k	8M	8k	72k
	en	news	779M	13B		3M

Table 1: Statistics of data used to train the bilingual word embeddings for evaluating cross-lingual lexical semantic similarity in YiSi-2.

where $\mathbb{U}$ is the set of all tested sentences in the same language of the word unit u; α is the ratio of precision and recall in the final YiSi score. In this experiment, we set α to 0.5 for a balanced ratio of precision and recall.

This year, we experimented with two methods to build the bilingual word embeddings for evaluating cross-lingual lexical semantic similarity in YiSi-2. The *supervised* bilingual word embeddings are trained on the parallel data provided using `bivec` (Luong et al., 2015). The *unsupervised* (weakly supervised, to be precise) bilingual word embeddings are built by transforming monolingual `w2v` (Mikolov et al., 2013) embeddings of each language into the same vector space using `vecmap` (Artetxe et al., 2016). Table 1 shows the statistics of the data used to train the two bilingual word embedding models. Common Crawl data was not used to train the bilingual word embeddings.

2.2.2 Deep Transformer Network (XLM)

The other approach we tested to score sentence pairs exploits self-supervised cross-lingual language model pre-training (Lample and Conneau, 2019) of a deep Transformer network, followed by a fine-tuning stage where we teach the network to distinguish real (good) parallel sentences from bad ones. We thereby transfer over knowledge acquired from a token-level (cross-lingual) language modelling task to a sentence-level (cross-lingual) discourse modelling task, i.e. predicting whether two sentences are translations of each other. This approach allows us to exploit both monolingual and parallel text during the unsupervised pre-training phase, therefore allowing us to profit from the greater availability of monolingual data.

Our use of XLM for sentence pair scoring is similar to the Zipporah system (Xu and Koehn, 2017), in that we train a model to discriminate between positive examples of actual translations and procedurally generated negative examples, then use the predicted probability of the positive class to score sentence pairs. The way we generate negative examples, which we will explain below, is also similar, but the model itself is very different.

Lample and Conneau (2019) introduced self-supervised cross-lingual language model pre-training of deep Transformer networks, and released a system called XLM (for cross-lingual language model).[3] The cross-lingual LM pre-training task is similar to the masked language model (MLM) pre-training used in BERT (Devlin et al., 2018), but the model can exploit cross-lingual context, as we will explain below. The architecture of XLM is a Transformer network like BERT, but it incorporates language embeddings in the input representation layer.

We used XLM to train a model using almost all the available data, except for the monolingual English Common Crawl data. This includes both monolingual and parallel data, and includes the Hindi datasets. All the data was preprocessed[4] using XLM's preprocessing tools, which include the Moses tokenizer (which defaults to English for both Nepali and Sinhala) and a script to remove accents and convert to lower case.

We then applied byte pair encoding[5] (BPE; Sen-

[3]https://github.com/facebookresearch/XLM

[4]The Nepali-English dictionary was first converted to the same format as the rest of the parallel data (two separate, line-aligned files).

[5]We used fastBPE (https://github.com/glample/fastBPE).

nrich et al., 2016a,b) to the training and test data, after learning 80K BPE codes on the concatenation of 3 monolingual corpora (of similar sizes) representing the 3 languages present in the test set (selected from sources similar to the benchmark, comprised of Wikipedia translations):

- Sinhala: all of Sinhala Wikipedia and all of Sinhala Common Crawl, for a total of 5.3 million sentences
- Nepali: all of Nepali Wikipedia, all of Nepali Common Crawl, and 1.6M sentences sampled from the monolingual Hindi corpus, for a total of 5.3 million sentences
- English: 5.3 million sentences sampled from English Wikipedia

Once BPE was applied to all the training data, for each language and language pair, we kept 5K sentence pairs for validation, and used the rest for training. Statistics on the data used to train the deep Transformer network are shown in Table 2. In all, this data contains around 1 billion tokens, with a vocabulary[6] size of 95056 (including BPE codes and single characters).

The size differences between the training sets of the 4 languages (between 3.6 and 10 million sentences) and 3 language pairs (between 577K and 642K sentence pairs) were assumed to be unimportant, as XLM samples the languages during training, such that under-represented languages are sampled more frequently.[7]

To teach the Transformer network to distinguish good translations from bad ones, we generated negative examples based on the positive examples in the (clean) parallel training data, in a manner similar to that of Xu and Koehn (2017), but adapted to address one of the types of noise in the (noisy) test data, that is sentence pairs where either side (or both) are not in the right language. Note that corpus filtering systems often use language identification to heuristically filter out this type of noise, but we found it important to provide our system this type of negative example to help it learn to assign them low scores.

[6]We compute the vocabulary on the data used to learn the BPE codes, after applying BPE to it.

[7]We still recommend minding the size differences between languages, as the sampling function currently implemented in XLM will not behave as intended if the differences are too great.

For each of the positive examples in the (clean) parallel training data, we generate negative examples that are either inadequate or lack fluency (or both), the idea being that such sentences are not useful for training MT systems. Specifically, we generate 4 negative examples using the following 4 procedures:

1. Swap sentence in source or target with a confounding sentence randomly drawn from the test corpora (from either source or target, regardless of which side is being swapped).
2. Shuffle words in source or target. Make sure the one we shuffle contains at least 2 words.
3. Do both 1 and 2. Do these separately, so we may corrupt the same side twice, or both sides, but in different ways.
4. Either copy source as target, copy target as source or swap source and target. This is meant to learn to detect noise due to the source and/or target being in the wrong language.

Sampling negative examples from the test corpus (in method 1) was meant to teach the model something about the language used in the test data. We feared that this might teach the model that sentences like those in the test corpora are always negative, so we also tested an alternative source of confounding sentences, that is to draw them from the positive examples instead (in any language).

Note that sentence pairs where the target was identical to the source were removed before generating the data for fine-tuning. Not translating certain words does happen in practice (e.g. names, loan words) but if the whole text is a copy of the source, it is not very informative on the task of translating, and in the case of corpus filtering, it may be confounding, as we know some of the noise in the test data is comprised of identical or very similar text segments in the same language. We also removed pairs where both source and target contained a single word.

The model was trained using a fork of XLM which we modified to allow for fine-tuning on pre-labeled sentence pairs (rather than positive examples only, from which negative examples are generated on-the-fly by XLM).

We start by pre-training the model on both monolingual and cross-lingual (masked) language

Lang(s)	Training data sources	Nb training sentences	Nb validation sentences
hi	IITB (mono)	10M (sampled)	5000
si	Wiki, CC	5.2M	2500 each Wiki and CC
ne	Wiki, CC	3.6M	2500 each Wiki and CC
en	Wiki	10M (sampled)	5000
hi–en	IITB (para)	600K (sampled)	5000
si–en	Open Subtitles, GNOME/KDE/Ubuntu	642K	2500 each
ne–en	Bible, Global Voices, Penn Treebank, GNOME/KDE/Ubuntu, ne–en dictionary	577K	500 Treebank and 4500 Bible

Table 2: Data used to train the deep Transformer network. CC means Common Crawl. For more information on the data sources, see the overview paper on the corpus filtering task.

model tasks. The task can be defined as follows: given a sequence of words in which certain words have been masked, predict the masked words based on the observable ones. The sequence of words can be one or more sentences in a single language, or a parallel pair of sentences in two different languages. In the bilingual case, the model can learn to use cross-lingual context in order to predict words, that is to use not only the context in the language the word belongs to, but also the translation of that context (and the word itself) in another language. Note that the input representation layer in XLM includes language embeddings, which are added to the input representation of each token. We thus specify the language of the texts being fed to the (multilingual) encoder.

Then we fine-tune the model on a sentence pair classification (SPC) task, which can be defined as follows: given two sentences, a source and a target, is the target a valid (i.e. adequate and fluent) translation of the source. This is done only on parallel data, and instead of using only real examples of translations, as during pre-training, we train on both positive and negative examples (in a ratio of 1:4).

During fine-tuning, we can choose to keep training the model on the language modeling tasks, to avoid overfitting the new data or forgetting too much about the old. We tested this approach, using only monolingual data for the language model task during fine-tuning – this was done for practical reasons, to avoid having the model update its language model on the negative examples in the parallel training sets used for fine-tuning.[8]

To set the hyperparameters, we used the default values or those used by Lample and Conneau (2019), with a few exceptions. We reduced the number of layers from 12 to 6 and the embedding size from 1024 to 512. We reduced the maximum batch size for pre-training from 64 to 32 (because of limited GPU memory), with around 4000 tokens per batch, and used a learning rate of 2e-4 for pre-training. For fine-tuning, we used a batch size of 8 and a learning rate of 1e-5.

[8]Our fork of XLM was created simply to accommodate fine-tuning on pre-labeled examples, and was not fool-proof in this respect.

It is worth noting that this model was supposed to be pre-trained for a week or more, but we discovered an issue with our data and had to restart pre-training the day before the deadline, so we were only able to pre-train it for 16 hours or so. Our preliminary experiments suggest we could have reduced the perplexity of the (monolingual and cross-lingual) LM by two thirds or more if we had pre-trained fully, but we do not know what effect this would have had on the sentence pair scoring task. We also had to foreshorten fine-tuning, as we only had time to do a few epochs. It is also worth noting that we only had time to evaluate MT quality on a 1M-word sample of Sinhala before the deadline, which may have made our model selection suboptimal.

2.3 Re-ranking to improve vocabulary coverage

Our scoring mechanisms process each sentence pair independently, therefore we sometimes observe redundancy in the top-ranking sentences, as well as a somewhat limited coverage of the words of the source language. To mitigate this issue, we applied a form of re-ranking to improve source token coverage. Going down the ranked list of (previously scored) sentence pairs, we applied a penalty to the pair's score if it did not contain at least one "new" source-language word bigram, i.e., a pair of consecutive source-language tokens not observed in previous (higher-scoring) sentence pairs. The penalty was simply a 20% score discount. This had the effect of down-ranking sen-

Source of confounders	Fine-tuning tasks	Acc (hi–en)	Acc (ne–en)	Acc (si–en)	Acc (avg)
Test set	SPC only	96.3	99.3	94.8	96.8
Test set	SPC+MLM	92.8	98.2	88.3	93.1
Train set	SPC only	95.6	95.7	93.2	94.8
Train set	SPC+MLM	93.7	93.4	91.1	92.8

Table 3: Sentence pair classification accuracy of XLM model on dev sets. *Confounders* are sentences that we draw at random to create inadequate translations.

	ne-en		si-en	
system	1M-word	5M-word	1M-word	5M-word
Zipporah*	3.40	4.22	4.16	4.77
random	1.30	3.01	1.43	3.33
Zipporah	4.14	**4.42**	4.12	**4.96**
YiSi-2-sup	3.86	3.76	4.85	4.71
YiSi-2-unsup	**4.42**	3.91	3.97	4.56
XLM-v2-spc	4.14	4.09	4.52	4.72
XLM-v2-spc-mlm	3.96	3.69	4.37	4.68
XLM-v3-spc-mlm	3.89	3.91	4.12	4.66
ensemble	3.94	3.95	**4.89**	4.85

Table 4: Uncased BLEU scores on the official dev ("dev-test") sets achieved by the SMT systems trained on the 1M- and 5M-word corpora subselected by the scoring systems. For XLM, v2 is the version that selects confounders from the test corpora, whereas v3 selects them from the training data, and spc-mlm means that both SPC and MLM were used for fine-tuning. *These results for the Zipporah baseline were reported by the task organizers, and the SMT architecture was different from our systems. We obtained Zipporah's score lists and trained our own SMT systems using the data selected from those lists, and results are shown in the third row.

tences that were too similar to a previously selected sentence.

2.4 Ensembling

To combine the output of different sentence pair scoring methods, we use the following, rank-based function:

$$s^*(e, f) = 1 - \frac{1}{|S| \times N} \sum_{s \in S} r(s(e, f))$$

where N is the number of sentence pairs, S is the set of scoring functions, and $r(s(e, f))$ returns the rank of the pair of sentences (e, f) according to score s.

3 Experiments and results

3.1 Intrinsic evaluation of XLM

To evaluate the deep Transformer model intrinsically, we can look at its accuracy on the sentence pair classification task used to fine-tune it. Table 3 shows the accuracy on the dev sets for all three language pairs. The table shows the results obtained using 4 different configurations for training, with the confounding sentences being drawn either from the training data or test data, and using either sentence pair classification (SPC) only or both SPC and the (monolingual) masked language model (MLM) for fine-tuning. First, we see that the accuracy scores are high,[9] so the model is good at discriminating real translations from procedurally generated bad ones.

The results also suggest that including the (monolingual) MLM task during fine-tuning is a hindrance, since the model achieves lower accuracy. However, it is important to note that we did no hyperparameter tuning, had to use a smaller model because of time and resource limitations, and did not have time to fully train any of the models tested. More extensive testing would be required to assess the usefulness of multi-task fine-tuning.

If we analyze the scores output by the model on the test data (i.e. the predicted probability of

[9]Picking the most frequent class would achieve 80% accuracy, as 80% of the examples are negative.

		SMT		NMT	
langs	system	1M-word	5M-word	1M-word	5M-word
ne–en	YiSi-2-sup	3.55 (10)	4.07 (T-14)	3.06 (12)	1.34 (10)
	YiSi-2-unsup	4.04 (T-4)	4.14 (T-12)	3.74 (8)	0.98 (16)
	XLM-v2-spc	3.92 (7)	**4.51 (4)**	4.03 (7)	**1.40 (9)**
	ensemble	**4.10 (3)**	4.30 (8)	**4.58 (5)**	1.10 (14)
si–en	YiSi-2-sup	3.87 (6)	4.39 (9)	**4.97 (3)**	**1.58 (9)**
	YiSi-2-unsup	3.14 (13)	4.29 (10)	2.41 (12)	0.68 (15)
	XLM-v2-spc	3.80 (T-8)	4.42 (T-7)	1.63 (15)	0.91 (13)
	ensemble	**4.19 (3)**	**4.54 (4)**	4.06 (4)	1.39 (11)

Table 5: BLEU scores (and ranking, out of 21 submissions for ne–en and 23 for si–en) of NRC's submissions on the test sets. The best of our submissions in each test setting is bolded.

the positive class), we see that the model predicts that a vast majority of sentences pairs are not valid translations of each other, their score being below 0.5. We briefly inspected the top-scoring sentences in the test set,[10] and in the case of ne–en, these seem to contain a lot of biblical texts, which suggests a domain bias, as the ne–en fine-tuning data included biblical texts.

3.2 MT quality check

We used the software provided by the task organizers to extract the 1M-word and 5M-word samples from the original test corpora, using the scores of each of our 4 systems in turn. We then trained SMT systems using the extracted data. The SMT systems were trained using Portage (Larkin et al., 2010) with components and parameters similar to the German-English SMT system in Williams et al. (2016). The MT systems were then evaluated on the official dev set ("dev-test"). Table 4 shows their BLEU scores. We have also included the results of a random scoring baseline (with initial filtering and token coverage re-ranking), as well as those of Zipporah.

These results show that all our BLEU scores are above the random baseline, and some of our systems outperform Zipporah when using a 1M-word sample (for both ne–en and si–en), but not when using a larger, 5M-word sample. We also see that our ensembling method produced good results on si–en, but not on ne–en, where individual systems fared better.

It is also interesting to note that in some cases, the 5M-word sample produced poorer MT results than the 1M-word sample. In fact, we see that the 1M-word samples selected by our best systems produce similar MT quality than the 5M-word samples selected by Zipporah.

Based on these results, we decided to submit the Transformer model that was fine-tuned on v2 of the fine-tuning data (where confounders were drawn from the test corpora), using SPC only, as well as both YiSi models and an ensemble of these three models.

[10]We used the XLM model's scores directly for this, and did not apply re-ranking.

4 Official Results

Table 5 presents the BLEU scores of our 4 systems on the test sets, using either 1M-word or 5M-word samples. Our best systems were ranked 3rd or 4th out of over 20 submissions in most test settings, except when using NMT on a 5M-word sample. It is worth noting that we were not able to conduct any NMT tests during development due to resource limitations, and were thus unable to tune any of our systems for this test setting.

If we compare the results of our 4 systems, the ensemble system performed best in 4 of 8 test settings, whereas XLM and YiSi (supervised) were best in 2 settings each. The ensemble system was most robust with an average score of 3.53 over all 8 test settings.

In the case of NMT, BLEU scores are much lower when using 5M-word rather than 1M-word samples, and this was true for other top systems, which suggests there is less that 5M words worth of parallel data in the test corpora that are useful (i.e. not too noisy) for NMT training. In the case of SMT, BLEU scores are slightly higher when using the larger samples, which suggests SMT is more robust to noise in the training data. Finally, it is worth noting that our best scores and rankings are similar for both language pairs.

5 Conclusion

In this paper, we presented the NRC's submissions to the WMT19 parallel corpus filtering task. Official results indicate our best systems were ranked 3rd or 4th out of over 20 submissions in most test settings, except when using NMT on a 5M-word sample, and that the ensemble system provided the most robust results. Further experimentation is required to understand why the sentence pair rankings produced by our systems work well for NMT if we take a small sample of top-ranked pairs, but less well if we take larger samples. A better way of re-ranking the pairs to optimize vocabulary coverage may lead to improved MT performance. Future work could also include using self-training to adapt the Transformer network to the test data, by iteratively selecting the most likely good examples in the test data and updating the language model and/or sentence pair classification model using these examples.

Acknowledgments

We thank the anonymous reviewers for their helpful suggestions on this paper.

References

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2016. Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2289–2294.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Cyril Goutte, Marine Carpuat, and George Foster. 2012. The impact of sentence alignment errors on phrase-based machine translation performance. In *Proceedings of the Tenth Conference of the Association for Machine Translation in the Americas*.

Francisco Guzmán, Peng-Jen Chen, Myle Ott, Juan Pino, Guillaume Lample, Philipp Koehn, Vishrav Chaudhary, and Marc'Aurelio Ranzato. 2019. Two new evaluation datasets for low-resource machine translation: Nepali-english and sinhala-english. *CoRR*, abs/1902.01382.

Philipp Koehn, Kenneth Heafield, Mikel L. Forcada, Miquel Esplà-Gomis, Sergio Ortiz-Rojas, Gema Ramírez Sánchez, Víctor M. Sánchez Cartagena, Barry Haddow, Marta Bañón, Marek Střelec, Anna Samiotou, and Amir Kamran. 2018a. ParaCrawl corpus version 1.0. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Philipp Koehn, Huda Khayrallah, Kenneth Heafield, and Mikel Forcada. 2018b. Findings of the wmt 2018 shared task on parallel corpus filtering. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, Brussels, Belgium. Association for Computational Linguistics.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *CoRR*, abs/1901.07291.

Samuel Larkin, Boxing Chen, George Foster, Uli Germann, Eric Joanis, J. Howard Johnson, and Roland Kuhn. 2010. Lessons from NRC's Portage System at WMT 2010. In *5th Workshop on Statistical Machine Translation (WMT 2010)*, pages 127–132.

Chi-kiu Lo. 2019. YiSi - a unified semantic mt quality evaluation and estimation metric for languages with different levels of available resources. In *Proceedings of the Fourth Conference on Machine Translation: Shared Task Papers*, Florence, Italy. Association for Computational Linguistics.

Chi-kiu Lo, Cyril Goutte, and Michel Simard. 2016. CNRC at Semeval-2016 task 1: Experiments in crosslingual semantic textual similarity. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 668–673.

Chi-kiu Lo, Michel Simard, Darlene Stewart, Samuel Larkin, Cyril Goutte, and Patrick Littell. 2018. Accurate semantic textual similarity for cleaning noisy parallel corpora using semantic machine translation evaluation metric: The NRC supervised submissions to the parallel corpus filtering task. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 908–916, Belgium, Brussels. Association for Computational Linguistics.

Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Bilingual word representations with monolingual quality in mind. In *NAACL Workshop on Vector Space Modeling for NLP*, Denver, United States.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems*, NIPS'13, pages 3111–3119, USA. Curran Associates Inc.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *40th Annual Meeting of the Association for Computational Linguistics (ACL-02)*, pages 311–318, Philadelphia, Pennsylvania.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Edinburgh neural machine translation systems for WMT 16. In *Proceedings of the First Conference on Machine Translation*, pages 371–376, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Michel Simard. 2014. Clean data for training statistical MT: the case of MT contamination. In *Proceedings of the Eleventh Conference of the Association for Machine Translation in the Americas*, pages 69–82, Vancouver, BC, Canada.

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.

Philip Williams, Rico Sennrich, Maria Nadejde, Matthias Huck, Barry Haddow, and Ondřej Bojar. 2016. Edinburgh's statistical machine translation systems for wmt16. In *Proceedings of the First Conference on Machine Translation*, pages 399–410, Berlin, Germany. Association for Computational Linguistics.

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, Copenhagen, Denmark. Association for Computational Linguistics.