

Dual Conditional Cross-Entropy Filtering of Noisy Parallel Corpora

Marcin Junczys-Dowmunt

Microsoft

1 Microsoft Way

Redmond, WA 98121, USA

Abstract

In this work we introduce dual conditional cross-entropy filtering for noisy parallel data. For each sentence pair of the noisy parallel corpus we compute cross-entropy scores according to two inverse translation models trained on clean data. We penalize divergent cross-entropies and weigh the penalty by the cross-entropy average of both models. Sorting or thresholding according to these scores results in better subsets of parallel data. We achieve higher BLEU scores with models trained on parallel data filtered only from Paracrawl than with models trained on clean WMT data. We further evaluate our method in the context of the WMT2018 shared task on parallel corpus filtering and achieve the overall highest ranking scores of the shared task, scoring top in three out of four subtasks.

1 Introduction

Recently, large web-crawled parallel corpora which are meant to rival non-public resources held by popular machine translation providers have been made publicly available to the research community in form of the Paracrawl corpus.¹ At the same time, it has been shown that neural translation models are far more sensitive to noisy parallel training data than phrase-based statistical machine translation methods (Khayrallah and Koehn, 2018; Belinkov and Bisk, 2017). This creates the need for data selection methods that can filter harmful sentence pairs from these large resources.

In this paper, we introduce dual conditional cross-entropy filtering, a simple but effective data selection method for noisy parallel corpora. We think of it as the missing adequacy component to the fluency aspects of cross-entropy difference filtering by Moore and Lewis (2010). Similar to Moore-Lewis filtering for monolingual data, we

directly select samples that have the potential to improve perplexity (and in our case translation performance) of models trained with the filtered data.

This is different from Axelrod et al. (2011) who simply expand Moore and Lewis filtering to both sides of the parallel corpus. We use conditional probability distributions and enforce agreement between inverse translation directions.

In most cases, neural translation models are trained to minimize perplexity (or cross-entropy) on a training set. Our selection criterion includes the optimization criterion of neural machine translation which we approximate by using neural translation models pre-trained on clean seed data.

We evaluated our method in the context of the WMT2018 Shared Task on Parallel Corpus Filtering (Koehn et al., 2018) and submitted our best method to the task. Although we only optimized for one of the four subtasks of the shared task, our submission scored highest for three out of four subtasks and third for the fourth subtask; there were 48 submissions to each subtask in total.

2 WMT 2018 shared task on parallel corpus filtering

We quote the shared task description provided by the organizers on the task website² and add citations where appropriate: The organizers “provide a very noisy 1 billion word (English token count) German-English corpus crawled from the web as part of the Paracrawl project” and “ask participants to subselect sentence pairs that amount to (a) 100 million words, and (b) 10 million words. The quality of the resulting subsets is determined by the quality of a statistical machine translation — Moses, phrase-based (Koehn et al., 2007) — and a neural machine translation system — Mar-

¹<https://paracrawl.eu>

²<http://www.statmt.org/wmt18/parallel-corpus-filtering.html>

ian (Junczys-Dowmunt et al., 2018) — trained on this data.” The organizers note that the task is meant to address “the challenge of data quality and not domain-relatedness of the data for a particular use case.” They discourage participants from sub-sampling the corpus for relevance to the news domain and announce that more emphasis will be put on undisclosed test sets rather than the WMT2018 test set.

Furthermore the organizers remark that “the provided raw parallel corpus is the outcome of a processing pipeline that aimed from high recall at the cost of precision, so it is very noisy. It exhibits noise of all kinds (wrong language in source and target, sentence pairs that are not translations of each other, bad language, incomplete or bad translations, etc.)” It is allowed to use the 2018 news translation task data for German-English (without the Paracrawl parallel corpus) to train components of our methods.

2.1 Sub-sampling based on submitted scores

Participants submit files with numerical scores, one score per line of the original unfiltered parallel corpus. A tool provided by the organizers takes as input the scores and the German and English corpus halves in form of raw text. Higher scores are better. The tool first determines at which best thresholds 10M and 100M words can be collected and next creates two data sets containing all sentences with scores above the two selected respective thresholds. Systems trained on these data sets are used for evaluation by the organizers (4 systems per submission) and for development purposes by task participants.

We focus on the 100M sub-task for neural machine translation systems as this is closest to our interests of finding as much relevant data as possible in large noisy parallel corpora. We only develop systems for this scenario.

2.2 Neural machine translation evaluation

As required by the shared task, we use Marian (Junczys-Dowmunt et al., 2018) to train our development systems. We follow the recommended settings quite closely in terms of model architecture, but change training settings, favoring hyper-parameters that lead to quicker convergence during our own development phase. We switched off synchronous ADAM in favor of asynchronous ADAM, increased the evaluation frequency to once per 5000 updates and increased work-space size to 5000MB per GPU. We also set the initial learning

rate to 0.0003 instead of 0.0001 and used an inverse square-root decaying scheme for the learning rate (Vaswani et al., 2017) that started after 16,000 updates. We removed dropout of source and target words and decreased variational dropout from 0.2 to 0.1 (Gal and Ghahramani, 2016). With these settings, our models usually converged within 10 to 15 hours of training on four NVidia Titan Xp GPUs. Convergence was assumed if perplexity did not improve for 5 consecutive evaluation steps. We evaluated on the provided WMT2016 and WMT2017 test sets.

3 Scores and experiments

We produce a single score $f(x, y)$ per sentence pair (x, y) as the product of partial scores $f_i(x, y)$:

$$f(x, y) = \prod_i f_i(x, y). \quad (1)$$

Partial scores take values between 0 and 1, as does the total score f . Partial scores that might generate values outside that range are clipped. We assume that sentence pairs with a score of 0 are excluded from the training data.³

In this section, we describe the scores explored in this work and present results on the development data.

3.1 Experimental baselines

Following the training recipe in Section 2.2, we first trained a model (“WMT18-full” in Table 2) on the admissible parallel WMT18 data for German-English (excluding Paracrawl). This model is only used for the computation of reference BLEU scores.

Next, we trained a German-English model on randomly scored Paracrawl data only (“random” in Table 2). The random scores – uniformly sampled values between 0 and 1 – were used to select representative data consisting of 100M words from unprocessed Paracrawl while using the threshold-based selection tool provided by the shared task organizers. Results for WMT16 and WMT17 test sets for both systems are shown in Table 2. The Paracrawl-trained systems (random) has dramatically worse BLEU scores than the WMT18-trained system. Upon manual inspection, we see many

³This is only guaranteed by the selection algorithm of the shared task if more than 100M words appear in sentence pairs scored with non-zero scores. However, we did not encounter situations where we got close or below that boundary.

Model	Description
W_{en}	RNN language model trained on 1M sentences from English WMT monolingual news data 2015-2017
P_{en}	RNN language model trained on 1M sentences from target (English) side of Paracrawl
$W_{\text{de} \rightarrow \text{en}}$	German-English translation model trained on WMT parallel data
$W_{\text{en} \rightarrow \text{de}}$	English-German translation model trained on WMT parallel data
$W_{\text{de} \leftrightarrow \text{en}}$	Translation model trained on union of German-English and English-German WMT parallel data

Table 1: Helper models trained for various scorers. All models are neural models, we do not use n-gram or phrase-based models. WMT parallel data excludes Paracrawl data.

untranslated and partially copied sentences in the case of the randomly-selected Paracrawl system.

3.2 Language identification

We noticed that the provided sentence pairs do not seem to have been subjected to language identification and simply used the Python `langid` package to assign a language code to each sentence in a sentence pair. We did not restrict the inventory of languages beforehand as we wanted the tool to propose a language if that language wins against all other candidates. We only accepted sentence pairs where both elements of a pair had been assigned the desired languages (German for source, English for target). The result is our first non-trivial score:

$$\text{lang}(x, l) = \begin{cases} 1 & \text{if LANGID}(x) = l \\ 0 & \text{otherwise} \end{cases}$$

$$\text{de-en}(x, y) = \text{lang}(x, \text{“de”}) \cdot \text{lang}(y, \text{“en”}) \quad (2)$$

This is a very harsh but also very effective filter that removes nearly 70% of the parallel sentence candidates. As a beneficial side-effect of language identification many language-ambiguous fragments which contain only little textual information are discarded, e.g. sentences with lots of numbers, punctuation marks or other non-letter characters. The

identification tool gets confused by the non-textual content and selects a random language.

We combined the $\text{de-en}(x, y)$ filter with the random scores and trained a corresponding system (de-en-random). As we see in Table 2, this strongly improved the results on both dev sets. When reviewing the translated development sets, we did not see any copied/untranslated sentences in the output.

3.3 Dual conditional cross-entropy filtering

The scoring method introduced in this section is our main contribution. While inspired by cross-entropy difference filtering for monolingual data (Moore and Lewis, 2010), our method does not aim for monolingual domain-selection effects. Instead we try to model a bilingual adequacy score.

Moore and Lewis (see next section) quantify the directed disagreement (signed difference) of similar distributions (two language models over the same language) trained on dissimilar data (different monolingual corpora). A stronger degree of separation between the two models indicates more interesting data.

In contrast, we try to find maximal symmetric agreement (minimal absolute difference) of dissimilar distributions (two translation models over inverse translation directions) trained on the same data (same parallel corpus). Concretely, for a sentence pair (x, y) we calculate a score:

$$\begin{aligned} & |H_A(y|x) - H_B(x|y)| \\ & + \frac{1}{2} (H_A(y|x) + H_B(x|y)) \end{aligned} \quad (3)$$

where A and B are translation models trained on the same data but in inverse directions, and $H_M(\cdot|\cdot)$ is the word-normalized conditional cross-entropy of the probability distribution $P_M(\cdot|\cdot)$ for a model M :

$$\begin{aligned} H_M(y|x) &= -\frac{1}{|y|} \log P_M(y|x) \\ &= -\frac{1}{|y|} \sum_{t=1}^{|y|} \log P_M(y_t|y_{<t}, x). \end{aligned}$$

The score (denoted as dual conditional cross-entropy) has two components with different functions: the absolute difference $|H_A(y|x) - H_B(x|y)|$ measures the agreement between the two conditional probability distributions, assuming that (word-normalized) translation

probabilities of sentence pairs in both directions should be roughly equal. We want disagreement to be low, hence this value should be close to 0.

However, a translation pair that is judged to be equally improbable by both models will also have a low disagreement score. Therefore we weight the agreement score by the average word-normalized cross-entropy from both models. Improbable sentence pairs will have higher average cross-entropy values.

This score is also quite similar to the dual learning training criterion from He et al. (2016) and Hassan et al. (2018). The dual learning criterion is formulated in terms of joint probabilities, later decomposed into translation model and language model probabilities. In practice, the influence of the language models is strongly scaled down which results in a form more similar to our score.

While Moore and Lewis filtering requires an in-domain data set and a non-domain-specific data set to create helper models, we require a clean, relative high-quality parallel corpus to train the two dual translation models. We sample 1M sentences from WMT parallel data excluding Paracrawl and train Nematus-style translation models $W_{de \rightarrow en}$ and $W_{en \rightarrow de}$ (see Table 1).

Formula (3) produces only positive values with 0 being the best possible score. We turn it into a partial score with values between 0 and 1 (1 being best) by negating and exponentiating, setting $A = W_{de \rightarrow en}$ and $B = W_{en \rightarrow de}$:

$$\text{adq}(x, y) = \exp(-(|H_A(y|x) - H_B(x|y)| + \frac{1}{2}(H_A(y|x) + H_B(x|y)))).$$

Combining the adq filter with the de-en filter results in a promising NMT system (de-en · adq in Table 2) trained on Paracrawl alone that beats the BLEU scores of the pure-WMT baseline.

We further evaluated three ablative systems:

- we omitted the language id filter (no de-en) which resulted in a system worse than randomly selected. This is not too surprising as we would expect many identical strings to be selected as highly adequate;
- we dropped the absolute difference from formula (3) which decreased BLEU by about 1 point;
- we removed the weighting by the averaged cross-entropies from formula (3), losing about 3 BLEU points.

This seems to indicate that the two components of the dual conditional cross-entropy filter are indeed useful and that we have a practical scoring method for parallel data.

3.4 Cross-entropy difference filtering

When inspecting the training data generated with the above methods we saw many fragments that looked like noisy or not particularly useful data. This included concatenated lists of dates, series of punctuation marks or simply not well-formed text. Due to the adequacy filtering, the noise was at least adequate, i.e. similar or identical on both sides and mostly correctly translated if applicable. The language filter had made sure that only few fully identical pairs of fragments had remained.

However, we preferred to have a training corpus that also looked like clean data. To achieve this we treated cross-entropy filtering proposed by Moore and Lewis (2010) as another score. Cross-entropy filtering or Moore-Lewis filtering uses the quantity

$$H_I(x) - H_N(x) \quad (4)$$

where I is an in-domain model, N is a non-domain-specific model and H_M is the word-normalized cross-entropy of a probability distribution P_M defined by a model M :

$$\begin{aligned} H_M(x) &= -\frac{1}{|x|} \log P_M(x) \\ &= -\frac{1}{|x|} \sum_{t=1}^{|x|} \log P_M(x_t | x_{<t}). \end{aligned}$$

Sentences scored with this method and selected when their score is below a chosen threshold are likely to be more in-domain according to model I and less similar to data used to train N than sentences above that threshold.

We chose WMT English news data from the years 2015-2017 as our in-domain, clean language model data and sampled 1M sentences to train model $I = W_{en}$. We sampled 1M sentences from Paracrawl without any previously applied filtering to produce $N = P_{en}$. The shared task organizers encourage submitting teams to not optimize for a specific domain, but it has been our experience that news data is quite general and clean data beats noisy data on many domains.

To create a partial score for which the best sentence pairs produce a 1 and the worst at 0, we apply a number of transformations. First, we negate and

exponentiate cross-entropy difference arriving at a quotient of perplexities of the target sentence y (x is ignored):

$$\begin{aligned} \text{dom}'(x, y) &= \exp(-(H_I(y) - H_N(y))) \\ &= \frac{\text{PP}_N(y)}{\text{PP}_I(y)}. \end{aligned}$$

This score has the nice intuitive interpretation of how many times sentence y is less perplexing to the in-domain model W_{en} than to the out-of-domain model P_{en} .

We further clip the maximum value of the score to 1 (the minimum value is already 0) as:

$$\text{dom}(x, y) = \min(\text{dom}'(x, y), 1). \quad (5)$$

This seems counterintuitive at first, but is done to avoid that a high monolingual in-domain score strongly overrides bilingual adequacy; we are fine with low in-domain scores penalizing sentence pairs. This is a precision-recall trade-off for adequacy and we prefer precision.

Finally, we also propose a cut-off value c as a parameter:

$$\text{cut}(x, c) = \begin{cases} x & \text{if } x \geq c \\ 0 & \text{otherwise} \end{cases}$$

$$\text{dom}_c(x, y) = \text{cut}(\text{dom}(x, y), c). \quad (6)$$

Parameter c can be used to completely eliminate sentence pairs, regardless of other scores, if y is less than c times more perplexing to the out-of-domain model than to the in-domain model, or inversely $1/c$ times more perplexing to the in-domain model than the out-of-domain model. This seems useful if we want a hard noise-filter similar to the language-id filter described above.

We used the domain filter only in combination with the previously introduced filters. In Table 2, we can observe that any variant leads to small improvements of the model over variants without the dom filters. This is expected as we optimized for WMT news development sets. We experimented with three cut-off values: 0.00 (no cut-off), 0.25 and 0.50, reaching the highest BLEU scores for a cut-off value $c = 0.25$. This best result (bold in Table 2) was submitted to the shared task organizers as our only submission.

Future work should consider bilingual cross-entropy difference filtering as proposed by Axelrod et al. (2011) where both sides of the corpus undergo

Filter	test16	test17
WMT18-full	33.9	29.0
random	16.2	14.1
de-en-random	26.6	23.3
de-en-adq	35.1	30.2
- no de-en	15.4	12.7
- no absolute difference	33.8	29.3
- no CE weighting	31.7	27.4
de-en-adq·dom _{0.00}	35.5	30.5
de-en-adq·dom_{0.25}	36.0	31.0
de-en-adq·dom _{0.50}	35.4	30.6
de-en·sim	34.5	29.6
de-en·sim·dom _{0.25}	35.5	30.6
de-en·adq·sim·dom _{0.25}	35.5	30.7

Table 2: Results on development data. We only train neural models for the 100M sub-task. We did not optimize for any of the other three sub-tasks.

the selection process or experiment with conditional probability distributions (translation models) for domain filtering.

3.5 Cosine similarity of sentence embeddings

We further experimented with sentence embedding similarity to contrast this method with our cross-entropy based approach. Recently, Hassan et al. (2018) and Schwenk (2018) used cosine similarities of sentence embeddings in a common multi-lingual space to select translation pairs for neural machine translation. Both these approaches rely on creating a multi-lingual translation model across all available translation directions and then using the accumulated encoder representations (after summing or max-pooling contextual word-level embeddings across the time dimension) of sentences in a pair to compute similarity scores.

Following Hassan et al. (2018), we train a new multi-lingual translation model on WMT18 parallel data (excluding Paracrawl) by joining German-English and English-German training data into a mixed-direction training set (see model $W_{\text{de} \leftrightarrow \text{en}}$ in Table 1). For a given sentence x , we create its sentence embedding vector \mathbf{s}_x according to translation model $W_{\text{de} \leftrightarrow \text{en}}$ by collecting encoder representation vectors \mathbf{h}_1 to $\mathbf{h}_{|x|}$

$$\mathbf{h}_{1:|x|} = \text{Encoder}_{W_{\text{de} \leftrightarrow \text{en}}}(x) \quad (7)$$

which are then averaged to form a single vector

System	Avg-BLEU	System	Avg-BLEU
RWTH Neural Redund.	24.58	Our submission	26.50
RWTH Neural Indep.	24.53	AliMT Mix	26.44
Our submission	24.45	AliMT Mix-div	26.42
AliMT Mix	24.11	Prompsit Active	26.41
AliMT Mix-div	24.11	NRC yisi-bicov	26.40

(a) SMT 10M

System	Avg-BLEU	System	Avg-BLEU
Our submission	28.62	Our submission	32.05
RWTH Neural Redund.	28.01	AliMT Mix	31.93
RWTH Neural Indep.	28.00	AliMT Mix-div	31.92
Speechmatics best	27.97	NRC yisi-bicov	31.88
Speechmatics prime	27.88	NRC yisi	31.76

(b) SMT 100M

System	Avg-BLEU	System	Avg-BLEU
Our submission	111.63	RWTH Neural Redundancy	110.09
RWTH Neural Redundancy	110.09	AliMT Mix	110.07
AliMT Mix	110.07	AliMT Mix-div	110.05
AliMT Mix-div	110.05	RWTH Neural Independent	109.91
RWTH Neural Independent	109.91		

(c) NMT 10M

System	Avg-BLEU	System	Avg-BLEU
Our submission	32.05		
AliMT Mix	31.93		
AliMT Mix-div	31.92		
NRC yisi-bicov	31.88		
NRC yisi	31.76		

(d) NMT 100M

System	Avg-BLEU
Our submission	111.63
RWTH Neural Redundancy	110.09
AliMT Mix	110.07
AliMT Mix-div	110.05
RWTH Neural Independent	109.91

(e) Sum of all sub-tasks

Table 3: Top-5 out of 48 submissions for each of the four sub-tasks and total sum

representation

$$\mathbf{s}_x = \frac{1}{|x|} \sum_{t=1}^{|x|} \mathbf{h}_t. \quad (8)$$

For a given sentence pair (x, y) we compute the cosine similarity of \mathbf{s}_x and \mathbf{s}_y as

$$\text{sim}(x, y) = \cos(\angle \mathbf{s}_x \mathbf{s}_y) = \frac{\mathbf{s}_x \cdot \mathbf{s}_y}{|\mathbf{s}_x| |\mathbf{s}_y|}. \quad (9)$$

Since the model has seen both languages, English and German, as source data it can produce useful sentence representations of both sentences in a translation pair. Unlike Hassan et al. (2018), we did not define a cut-off value for the similarity score as the threshold-based selection method of shared-task tool computes its own cut-off thresholds.

We ran two experiments with the similarity based scores, evaluating configurations de-en-sim and de-en-adq·sim·dom_{0.25}. The first one corresponds to de-en-adq and we compare the effectiveness of the adq and sim filters after the application of the language-id-based filter de-en. We

see in Table 2 that while de-en-sim leads to improvements over the language-filtered randomly selected Paracrawl data, it is significantly worse than de-en-adq on both development sets. Interestingly, even when combined with our best scoring scheme (de-en-adq·dom_{0.25}) resulting in de-en-adq·sim·dom_{0.25} we see a slight degradation. Based on these results, we do not use the similarity scores for our submission.

In future experiments we want to use the multilingual model $W_{\text{de} \leftrightarrow \text{en}}$ instead of the two models $W_{\text{en} \rightarrow \text{de}}$ and $W_{\text{de} \rightarrow \text{en}}$ for our dual conditional cross-entropy method from Section 3.3. A multilingual model does not only have a common encoder, but also a common probability distribution for both languages which might lead to better agreement of the conditional cross-entropies.

4 Shared task results

As mentioned before, we submitted only our single-best set of scores de-en·adq·dom_{0.25} to the shared task. The shared task organizers trained four sys-

tems with each set of submitted scores, two Moses SMT (Koehn et al., 2007) systems on the best 10M and 100M words corpora and two neural Marian NMT systems on the same sets.

Based on the spread-sheet made available by the organizers, 48 sets of scores were submitted. Each set of scores was evaluated using the four mentioned models on 6 different test sets (newstest 2018, iwslt 2017, Acquis, EMEA, Global Voices, KDE). This required the organizers to train nearly 200 separate models; an effort that should be applauded.

It seems that systems are ranked by their average score across these test sets and sub-tasks. In Table 3 we selected the top-5 system across each sub-task for the purpose of this paper. The shared task overview will likely include a more thorough analysis. We place highest out of 48 submissions in three out of four tasks (SMT 100M, NMT 10M and NMT 100M) and third out of 48 for sub-task SMT 10M. The systems are packed quite closely, but the overall total across all four tasks shows, that we accumulate a slightly larger margin over the next best systems while the next four systems barely differ. This result is better than we expected as we only optimized for the NMT 100M task.

For more details on the evaluation process and conclusions see the shared task overview paper Koehn et al. (2018).

5 Future work and discussion

We introduced dual conditional cross-entropy filtering for noisy parallel data and combined this filtering with multiple other noise filtering methods. Our submission to the WMT 2018 shared task on parallel corpus filtering achieved the highest overall rank and scored best in three out of four subtasks while scoring third in the fourth subtask. Each subtask had 48 participants.

We believe this positive effect is rooted in the idea of directly asking a model that is very similar to the to-be-trained model which data it prefers (weighting by cross-entropy) while also constraining its answer with the introduced disagreement penalty. Our selection criterion is also very close to the optimization criterion used during NMT training, especially the dual learning training criterion. Other methods, for instance the evaluated similarity-based methods, do not have this direct connection to the training process.

Future work should concentrate on further for-

malizing this method. We should analyze the connection to the dual learning training criterion on experiments whether models that were trained with this criterion are also better candidates for sentences scoring. Furthermore, the models we used for scoring were trained on small subsamples of clean data, we should investigate if stronger translation and language models are better discriminators.

References

- Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 355–362. Association for Computational Linguistics.
- Yonatan Belinkov and Yonatan Bisk. 2017. Synthetic and natural noise both break neural machine translation. *CoRR*, abs/1711.02173.
- Yarin Gal and Zoubin Ghahramani. 2016. A theoretically grounded application of dropout in recurrent neural networks. In *Advances in neural information processing systems*, pages 1019–1027.
- Hany Hassan, Anthony Aue, Chang Chen, Vishal Chowdhary, Jonathan Clark, Christian Federmann, Xuedong Huang, Marcin Junczys-Dowmunt, William Lewis, Mu Li, Shujie Liu, Tie-Yan Liu, Renqian Luo, Arul Menezes, Tao Qin, Frank Seide, Xu Tan, Fei Tian, Lijun Wu, Shuangzhi Wu, Yingce Xia, Dongdong Zhang, Zhirui Zhang, and Ming Zhou. 2018. Achieving human parity on automatic chinese to english news translation. *CoRR*, abs/1803.05567.
- Di He, Yingce Xia, Tao Qin, Liwei Wang, Nenghai Yu, Tiejian Liu, and Wei-Ying Ma. 2016. Dual learning for machine translation. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 820–828. Curran Associates, Inc.
- Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121. Association for Computational Linguistics.
- Huda Khayrallah and Philipp Koehn. 2018. On the impact of various types of noise on neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 74–83, Melbourne, Australia. Association for Computational Linguistics.
- Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi,

- Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In ACL. The Association for Computer Linguistics.
- Philipp Koehn, Huda Khayrallah, Kenneth Heafield, and Mikel Forcada. 2018. 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.
- Robert C. Moore and William Lewis. 2010. Intelligent selection of language model training data. In Proceedings of the ACL 2010 Conference Short Papers, pages 220–224, Uppsala, Sweden. Association for Computational Linguistics.
- Holger Schwenk. 2018. Filtering and mining parallel data in a joint multilingual space. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 228–234. Association for Computational Linguistics.
- 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 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.