Curriculum Learning for Domain Adaptation in Neural Machine Translation

Xuan Zhang[1], Pamela Shapiro[1], Gaurav Kumar[1], Paul McNamee[1], Marine Carpuat[2], Kevin Duh[1]

[1]Johns Hopkins University
[2]University of Maryland

{xuanzhang, pshapiro, mcnamee}@jhu.edu,
marine@cs.umd.edu, {gkumar, kevinduh}@cs.jhu.edu

Abstract

We introduce a curriculum learning approach to adapt generic neural machine translation models to a specific domain. Samples are grouped by their similarities to the domain of interest and each group is fed to the training algorithm with a particular schedule. This approach is simple to implement on top of any neural framework or architecture, and consistently outperforms both unadapted and adapted baselines in experiments with two distinct domains and two language pairs.

1 Introduction

Neural machine translation (NMT) performance often drops when training and test domains do not match and when in-domain training data is scarce (Koehn and Knowles, 2017). Tailoring the NMT system to each domain could improve performance, but unfortunately high-quality parallel data does not exist for all domains. Domain adaptation techniques address this problem by exploiting diverse data sources to improve in-domain translation, including *general domain* data that does not match the domain of interest, and *unlabeled domain* data whose domain is unknown (e.g. webcrawl like Paracrawl).

One approach to exploit unlabeled-domain bitext is to apply *data selection* techniques (Moore and Lewis, 2010; Axelrod et al., 2011; Duh et al., 2013) to find bitext that are similar to in-domain data. This selected data can additionally be combined with in-domain bitext and trained in a continued training framework, as shown in Figure 1. Continued training or fine-tuning (Luong et al., 2015; Freitag and Al-Onaizan, 2016; Chu et al., 2017) is an adaptation technique where a model is first trained on the large general domain data, then used as initialization of a new model which is further trained on in-domain bitext. In our framework, the selected samples are concatenated with in-domain data, then used for continued training. This effectively increases the in-domain training size with "pseudo" in-domain samples, and is helpful in continued training (Koehn et al., 2018).

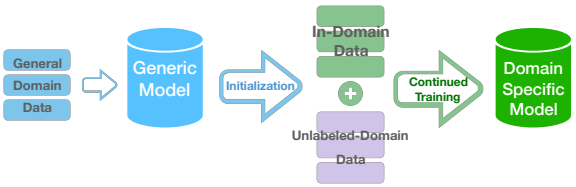

Figure 1: Workflow of our domain adaptation system.

A challenge with employing data selection in continued training is that there exists no clear-cut way to define whether a sample is sufficiently similar to in-domain data to be included. In practice, one has to define a threshold based on similarity scores, and even so the continued training algorithm may be faced with samples of diverse similarities. We introduce a new domain adaptation technique that addresses this challenge.

Inspired by curriculum learning (Bengio et al., 2009), we use the similarity scores given by data selection to rearrange the order of training samples, such that *more similar examples are seen earlier and more frequently during training*. To the best of our knowledge, this is the first work applying curriculum learning to domain adaptation.

We demonstrate the effectiveness of our approach on TED Talks and patent abstracts for German-English and Russian-English pairs, using two distinct data selection methods, Moore-Lewis method (Moore and Lewis, 2010) and cynical data selection (Axelrod, 2017). Results show that our approach consistently outperforms standard continued training, with up to 3.22 BLEU improvement. Our S^4 error analysis (Irvine et al., 2013) reveal that this approach reduces a reasonable number of SENSE and SCORE errors.

Proceedings of NAACL-HLT 2019, pages 1903–1915
Minneapolis, Minnesota, June 2 - June 7, 2019.

2 Curriculum Learning for Adaptation

Weinshall and Cohen (2018) provide guidelines for curriculum learning: "A practical curriculum learning method should address two main questions: how to rank the training examples, and how to modify the sampling procedure based on this ranking." For domain adaptation we choose to estimate the difficulty of a training sample based on its distance to the in-domain data, which can be quantified by existing data selection methods (Section 2.1). For the sampling procedure, we adopt a probabilistic curriculum training (CL) strategy that takes advantage of the spirit of curriculum learning in a nondeterministic fashion without discarding the good practice of original standard training policy, like bucketing and mini-batching.

2.1 Domain Similarity Scoring

We adopt similarity metrics from prior work on data selection to score examples for curriculum learning. Let I be an in-domain corpus, and N be a unlabeled-domain data set. Data selection models rank sentences in N according to a domain similarity measure with respect to I, and choose top n samples from N by a cut-off threshold for further training purpose. We examine two data selection methods, Moore-Lewis method (Moore and Lewis, 2010) and cynical data selection (Axelrod, 2017).

Moore-Lewis Method Each sentence s in N is assigned a *cross-entropy difference score*,

$$H_I(s) - H_N(s), \tag{1}$$

where $H_I(s)$ is the per-word cross-entropy of s according to a language model trained on I, and $H_N(s)$ is the per-word cross-entropy of s according to a language model trained on a random sample of N with roughly the same size as I. A lower cross-entropy difference indicates that s is more like the in-domain data and less like the unlabeled-domain data.

Cynical Data Selection Iteratively select sentence s from N to construct a training corpus that would approximately model I. At each iteration, each sentence is scored by the expected cross-entropy change from adding it to the already selected subset of N. The selected sentence is the one which most decreases H_n, the cross-entropy between previously selected n-sentence corpus and I.

2.2 Curriculum Learning Training Strategy

We identify two general types of curriculum learning strategy. The *deterministic* curriculum (c.f. Kocmi and Bojar (2017)) trains on a fixed order of samples based on their scores (e.g. "easy-to-hard" or "more similar to less"). While simple to motivate, this may not always perform well because neural methods benefit from randomization in the minibatches and multiple epochs. In contrast, the *probabilistic* curriculum (Bengio et al., 2009) works by dividing the training procedure into distinct phases. Each phase creates a random sample from the entire pool of data, but earlier phases sample the "easier" or "more similar" sentence with higher probability.. Since each phase can be viewed as creating a new training dataset, all the well-tested tricks of the trade for neural network optimization can be employed.

In this paper, we use the same probabilistic curriculum strategy and code base[1] as Zhang et al. (2018). The main difference here is the application to domain adaptation. The proposed strategy is summarized as follows:

- Sentences are first ranked by similarity scores and then distributed evenly into shards, such that each shard contains samples with similar similarity criteria values.
- The training process is segmented into consecutive *phases*, where only a subset of shards are available for training.
- During the first phase, only the easiest shard is presented. When moving to the next phase, the training set will be increased by adding the second easiest shard into it, and so on. Easy shards are those that are more similar to the in-domain data, as quantified by either Moore-Lewis or Cynical Data Selection.
- The presentation order of samples is not deterministic. (1) Shards within one curriculum phase are shuffled, so they are not necessarily visited by the order of similarity level during this phase. (2) Samples within one shard are bucketed by length and batches are drawn randomly from buckets.

[1] https://github.com/kevinduh/sockeye-recipes/tree/master/egs/curriculum

3 Experiments and Results

We evaluate on four domain adaptation tasks. The code base is provided to ensure reproducibility.[2]

3.1 Data and Setup

General Domain Data We have two general domain datasets, Russian-English (ru) and German-English (de). Both are a concatenation of OpenSubtitles2018 (Lison and Tiedemann, 2016) and WMT 2017 (Bojar et al., 2017), which contains data from several domains, e.g. parliamentary proceedings (Europarl, UN Parallel Corpus), political/economic news (news commentary, Rapid corpus), and web-crawled parallel corpus (Common Crawl, Yandex, Wikipedia titles). We performed sentence length filtering (up to 80 words) after tokenization, ending up with 28 million sentence pairs for German and 51 million sentence pairs for Russian.

In-domain Data We evaluate our proposed methods on two distinct domains per language pair:

- TED talks: data-split from Duh (2018).
- Patents: from the World International Property Organization COPPA-V2 dataset (Junczys-Dowmunt et al., 2016).

We randomly sample 15k parallel sentences from the original corpora as our in-domain bitext.[3] We also have around 2k sentences of development and test data for TED and 3k for patent.

Unlabeled-domain Data For additional unlabeled-domain data, we use web-crawled bitext from the Paracrawl project.[4] We filter the data using the Zipporah cleaning tool (Xu and Koehn, 2017), with a threshold score of 1. After filtering, we have around 13.6 million Paracrawl sentences available for German-English and 3.7 million Paracrawl sentences available for Russian-English. Using different data selection methods, we include up to the 4096k and 2048k sentence-pairs for our German and Russian experiments, respectively.

Data Preprocessing All datasets are tokenized using the Moses (Koehn et al., 2007) tokenizer. We learn byte pair encoding (BPE) segmentation models (Sennrich et al., 2016) from general domain data. The BPE models are trained separately for each language, and the number of BPE symbols is set to 30k. We then apply the BPE models to in-domain and Paracrawl data, so that the parameters of the generic model can be applied as an initialization for continued training. Once we have a converged generic NMT model, which is very expensive to train, we can adapt it to different domains, without building up a new vocabulary and retraining the model.

NMT Setup Our NMT models are developed in Sockeye[5] (Hieber et al., 2017). The generic model and continued training model are trained with the same hyperparameters. We use the seq2seq attention architecture (Bahdanau et al., 2015) with 2 LSTM layers for both encoder and decoder, and 512 hidden nodes in each layer. The word embedding size is also set to 512. Our models apply Adam (Kingma and Ba, 2014) as the optimizer, with an initial learning rate 0.0003. The learning rate is multiplied by 0.7 whenever validation perplexity does not surpass the previous best in 8 checkpoints.[6] We use minibatches of 4096 words. Training stops when the perplexity on the development set has not improved for 20 checkpoints (1000 updates/batches per checkpoint).

Domain Similarity Scoring Setup To get similarity scores, we build 5-gram language models on the source side[7] with modified Kneser-Ney smoothing using KenLM (Heafield, 2011).

Curriculum Learning Setup The number of batches in each curriculum phase is set to 1000. We split the training data into 40 shards[8], with all the 15k in-domain data in the first shard, and Paracrawl data split into the remaining 39 shards.

3.2 Experimental Comparison

Our goal is to empirically test whether the proposed curriculum learning method improves translation quality in the continued training setup of

[2] https://github.com/kevinduh/sockeye-recipes/tree/master/egs/curriculum

[3] Appendix A explains our choice of 15k in detail.

[4] https://www.paracrawl.eu/

[5] github.com/awslabs/sockeye

[6] The Adam optimizer for continued training model is initialized without reloading from the trained generic model.

[7] Appendix D also shows the effect of using language models built from target side and both sides.

[8] After experimenting with various values from 5 to 100 (Appendix B), we found best performance can be achieved at 40 shards.

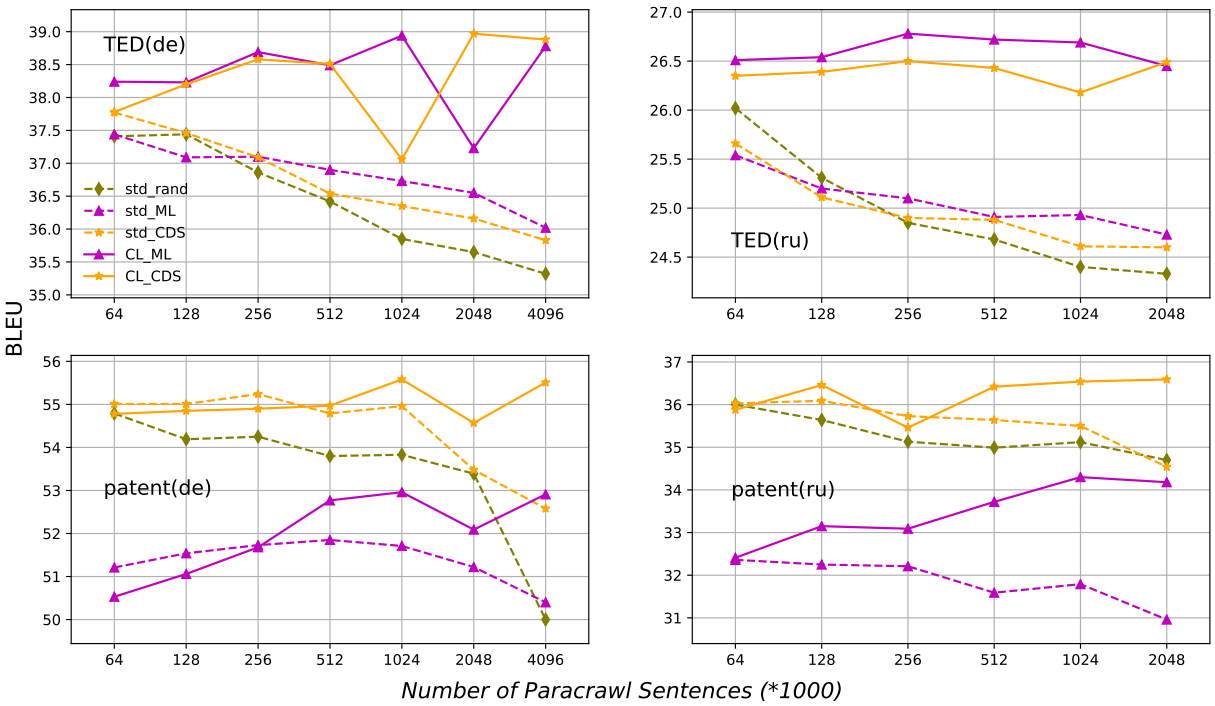

Figure 2: BLEU of adapted models using a concatenation of in-domain and varying amounts of Paracrawl data.

Figure 1. We compare two approaches to continued training: (1) the standard approach reads batches of in-domain and selected Paracrawl in random order; (2) the proposed curriculum learning approach reads these batches according to a schedule. We run the comparison with two data selection methods, leading to four systems:

- **std_ML**: standard continued training with Moore-Lewis scores
- **CL_ML**: curriculum learning approach to continued training with Moore-Lewis scores
- **std_CDS**: standard continued training with scores from Cynical Data Selection
- **CL_CDS**: curriculum learning approach to continued training with scores from Cynical Data Selection

For reference, we show results of the generic model (**GEN**), the model trained from scratch with in-domain data (**IN**), the model continued trained on in-domain data only (**IN_CT**), and a standard continued training model using a random subset (rather than ML or CDS scores) of the concatenated in-domain and Paracrawl data (**std_rand**).

3.3 Results

Table 1 summarizes the key results, where we continue train on 15k in-domain samples and 4096k Paracrawl samples (for de) or 2048k Paracrawl samples (for ru):

- The baseline BLEU scores confirm the need for domain adaptation. Using only the 15k in-domain samples alone (IN) is not sufficient to train a strong domain specific model, yielding BLEU scores as low as 2.53 on TED(de) and 1.76 on TED(ru). The model trained with a large amount of general domain data (GEN) is a stronger baseline, with BLEU scores of 34.59 and 23.40.
- Standard continued training is not robust to samples that are noisy and less similar to in-domain. As expected, continued training on in-domain data (IN_CT) improves BLEU significantly, by up to 18.74 BLEU on patent(de). However, when adding Paracrawl data, the standard continued training strategy (std_rand, std_ML, std_CDS) consistently performs worse than IN_CT.
- Curriculum learning consistently improves BLEU score. Ranking examples using

	TED(de)	TED(ru)	patent(de)	patent(ru)
GEN	34.59	23.40	35.95	23.41
IN	2.53	1.76	12.09	16.81
IN_CT	36.16	25.04	54.70	35.61
std_rand	35.32	24.33	50.00	34.70
std_ML	36.02	24.73	50.40	30.96
CL_ML	38.78	26.45	52.91	34.18
Δ_ML	2.76	1.72	2.51	3.22
std_CDS	35.83	24.60	52.58	34.54
CL_CDS	**38.88**	**26.49**	**55.51**	**36.59**
Δ_CDS	3.05	1.89	2.93	2.05

Table 1: BLEU of unadapted & adapted models. Δ shows improvement of **CL** over **std**.

Moore-Lewis (CL_ML) and Cynical Data Selection (CL_CDS) improve BLEU over their baselines (std_ML and std_CDS) by up to 3.22 BLEU points.

As an additional experiment, we report results on different amounts of Paracrawl data. Figure 2 shows how the curriculum uses increasing amounts of Paracrawl better than standard continued training. Standard continued training model hurts BLEU when too much Paracrawl data is added: for TED(de), there's a 1.94 BLEU drop when increasing CDS data from 64k to 4096k, and for patent(de), the decrease is 2.43 BLEU. By contrast, the curriculum learning models achieve a BLEU score that is as good or better as the initial model, even after being trained on the most dissimilar examples. This trend is clearest on the patent(ru) CL_ML model, where the BLEU score consistently rises from 32.41 to 34.18.

The method used to score domain relevance has a different impact on the TED domain (top plots) and on the patent domain (bottom plots). On the patent domain, which is more distant from Paracrawl, CDS significantly outperforms ML. Replacing ML with CDS improve BLEU from 2.18 to 4.05 BLEU points for standard models and 2.20 to 4.25 BLEU points for curriculum learning models. Interestingly, for patents, the Moore-Lewis method does not beat the random selection, even when curriculum learning is applied. For example, at 64k selected sentences for patent(de), std_rand gets 4.26 higher BLEU scores than CL_ML. By contrast on the TED domain, which is closer to Paracrawl, the Moore-Lewis method slightly outperforms cynical data selection. Due to these differences, we suggest trying different data selection methods with curriculum learning on new tasks; a potential direction for future work may be a curriculum that considers multiple similarity scores jointly.

4 Analysis

4.1 Comparison of Curriculum Strategies

We compare our approach to other curriculum strategies. *CL_reverse* reverses the presenting order of the shards, so that shards containing less similar examples will be visited first, *CL_scrambled* is a model that adopts the same training schedule as *CL*, but no data selection method and ranking is involved here — Paracrawl data are evenly split and randomly assigned to shards; *CL_noshuffle* is another curriculum learning model that does not shuffle shards in each curriculum phase.

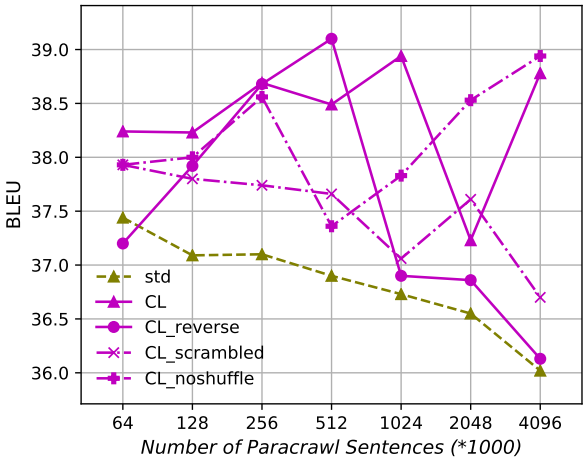

Figure 3: Comparison of various curriculum strategies on German-English TED corpora, where Moore-Lewis method is applied. [9]

Results from Figure 3 show that CL outperforms CL_reverse and CL_noshuffle for 5 out of 7 cases and outperforms CL_scrambled in 6 out of 7 cases. This suggests that it is beneficial to train on examples that are closest to in-domain first and to use a probabilistic curriculum. Analyzing the detailed difference between CL and CL_reverse would be interesting future work. One potential hypothesis why CL might help is that it first trains on a low-entropy subset of the data before moving on to the whole training set, which may have regularization effects.

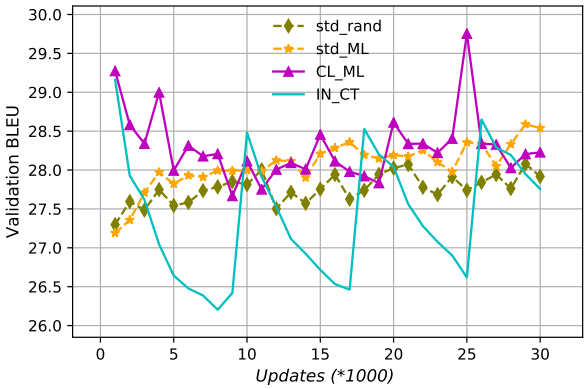

Figure 4: Learning curves for German-English TED NMTs. Except for IN_CT, the other three models all continued trained on the concatenation of in-domain and 1024k Paracrawl data.

[9] Each point represents a model trained to convergence on the fixed amount of in-domain and ParaCrawl data whose amount is specified by the x-axis.

4.2 Learning Curves

Learning curves (Figure 4) further illustrate the advantage of our method. Continued training on in-domain data only starts from a strong initialization (thanks to pre-training on large general domain data) but heavily oscillates over training without reaching the initial performance. This behavior may be due to the sparsity of the TED data: the small randomly sampled training set may not represent the development and test data well. Std_ML shows opposite behavior to IN_CT: it starts from a lower initial performance, and then gradually improves to a level comparable to IN_CT. Std_rand behaves similarly to std_ML—in other words, uniformly sampling from Paracrawl drags the initial performance down without helping with the final performance.

Compared to all above, the curriculum learning models start from a high initial performance, suffer much less oscillation than IN_CT, and gradually achieve the highest performance.[10]

4.3 Impact of Curriculum Learning on Lexical Choice: S^4 Analysis

How do translations improve when using curriculum learning? We characterize the impact of curriculum learning on lexical translation errors using the S^4 taxonomy of domain change errors introduced by Irvine et al. (2013) for phrase-based machine translation: (1) SEEN: incorrect translation for a source word that has never been seen in the training corpus; (2) SENSE: incorrect translation for a previously seen source word, whose correct translation (sense) has never been seen in the training corpus; (3) SCORE: a score error is made when the source word and its correct translation are both observed in training data, but the incorrect translation is scored higher than the correct alternative ; and (4) SEARCH: an error caused by pruning in beam search[11].

We extend this taxonomy to neural machine translation. As the unit of S^4 analysis is word alignment between a source word and a reference target word, we first run fast-align (Dyer et al., 2013) to get the source-target word alignments. After this, we follow the algorithm shown in Appendix C to give a summary of S^4 errors on the model's translation of test set.

[10]When converged, IN_CT does not outperform CL_ML.

[11]We will only focus on the first three error categories in this paper for the purpose of model comparison.

Figure 5 shows the word translation results for the test set of German-English TED. Most of the errors are SCORE errors, while SEEN and SENSE errors are relatively rare. Curriculum learning significantly improves the adapted NMT systems at the word level — with 4096k Paracrawl data selected by CDS, curriculum continued training model can translate 554 more words correctly than the standard continued training model. This improvement mainly happens in SCORE errors: 1.75% of SCORE errors are corrected. SEEN and SENSE errors are also reduced by 0.02% and 0.026%, respectively. But overall, CL does not help much on SEEN errors.

4.4 Characteristics of Selected Data

We characterize the sentences chosen by different data selection methods, to understand their effect on adaptation as observed in Section 3.3.

Selected Sentences Overlap For each domain in German-English, we compute the overlap between the top n ML and CDS Paracrawl sentences. The overlap is as low as 3.69% for the top 64k sentences in the TED domain, and 8.43% for the patent domain. Even in the top 4096k sentences, there are still 46.25% and 65.40% different ones in TED and patent domain respectively. See Table 2 for examples of selected sentences.

Average Sentence Length The ML score prefers longer sentences and is more correlated with sentence length (See Figure 6) — the curve TED_ML is near linear, which might be a side-effect of sentence-length normalization. CDS produces sentences that better match the average sentence length in the in-domain corpus, which was also observed in Santamaría and Axelrod (2017).

Out-of-Vocabulary Words We count out-of-vocabulary (OOV) tokens in in-domain corpus based on the vocabulary of selected unlabeled-domain data (Figure 7). The CDS subsets cover in-domain vocabulary better than ML subsets as expected, since CDS is based on vocabulary coverage.

Unigram Distribution Distance How do unigram relative frequencies compare in the in-domain and selected Paracrawl data?

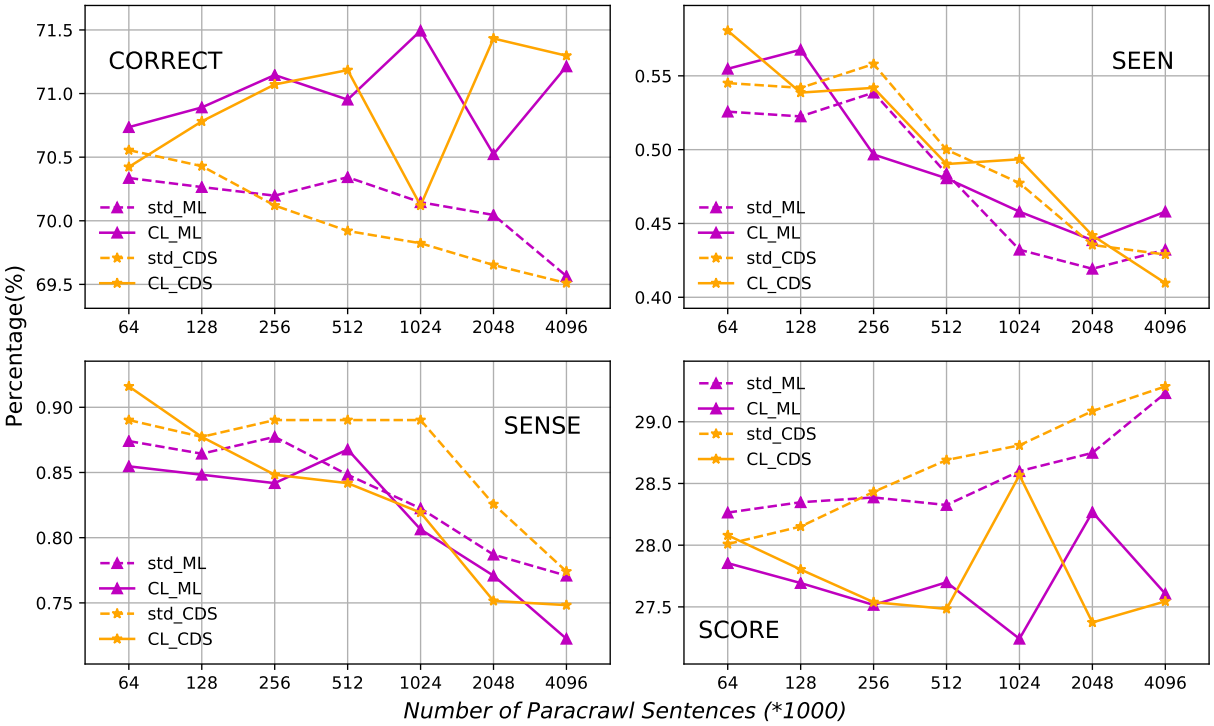

Figure 5: S^4 error analysis on German-English TED.

TED_ML	It changes the way we think; it changes the way we walk in the world; it changes our responses; it changes our attitudes towards our current situations; it changes the way we dress; it changes the way we do things; it changes the way we interact with people.
TED_CDS	But, on the other hand, this signifies that the right of self-determination, as a part of the proletarian peace program, possesses not a "Utopian" but a revolutionary character.
patent_ML	The sites x, y and z can accommodate a large variety of cations with x=na+, k+, ca2+, vacancy; y=mg2+, fe2+, al3+, fe3+, li+, mn2+ and z=al3+, mg2+ , fe3+, v3+, cr3+; while the t site is predominantly occupied by si4+.
patent_CDS	To select alternative viewing methods, such as for 3d-tv.

Table 2: The top ranked sentences selected from German-English Paracrawl corpus.

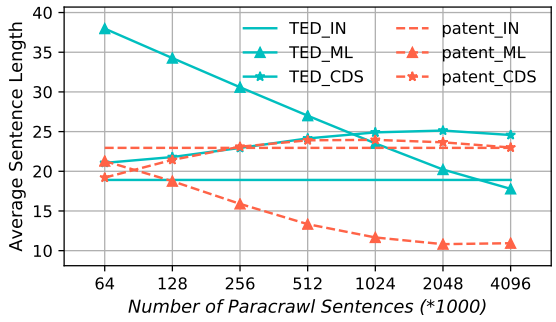

Figure 6: Average sentence length for increasing size of Paracrawl data. This is calculated on the source side of German-English pairs. TED_IN stands for TED corpus. TED_ML and TED_CDS represent the Paracrawl samples selected by ML and CDS methods.

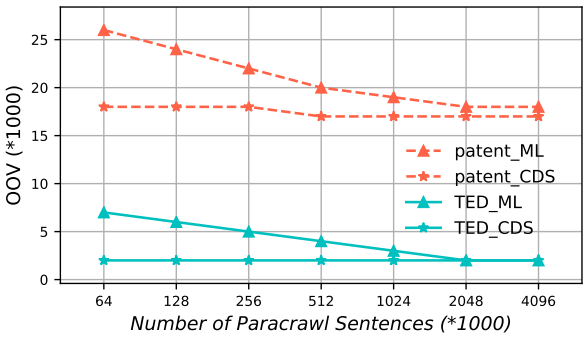

Figure 7: Number of OOV words in the source side of German-English target domain corpora.

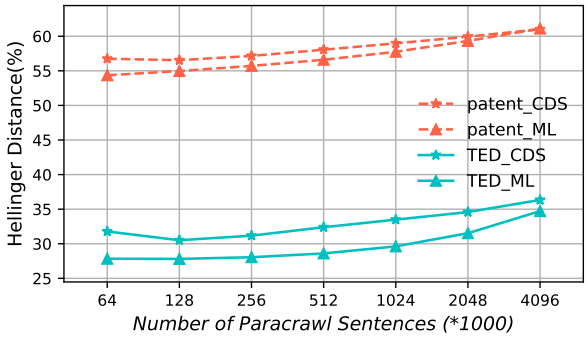

Figure 8: Hellinger distance for source side unigram distributions of German-English corpora between in-domain data and ML/CDS selected data.

We measure the difference of unigram distributions from two corpora by *Hellinger distance*, which is defined as Equation 2 when the probability distribution is discrete, where P and Q are the unigram distributions for the source side of in-domain and Paracrawl. V is the vocabulary size.[12]

$$H_{HD}(P,Q) = \frac{1}{\sqrt{2}}\sqrt{\sum_{i=1}^{V}(\sqrt{p_i} - \sqrt{q_i})^2}. \quad (2)$$

From Figure 8, we can see ML can better match the in-domain vocabulary distribution than CDS.

With respect to the OOV rate and unigram distribution, patent is more distant from the Paracrawl data than TED is. Figure 2 suggests that CDS dominates ML for distant domains such as Patent, while ML can do slightly better than CDS for domains that are not that distant such as TED.

5 Related Work

Curriculum learning has shown its potential to improve sample efficiency for neural models (Graves et al., 2017; Weinshall and Cohen, 2018) by guiding the order of presented samples, usually from easier-to-learn samples to difficult samples. Although there is no single criterion to measure difficulty for general neural machine translation tasks (Kocmi and Bojar, 2017; Wang et al., 2018; Zhang et al., 2018; Kumar et al., 2019; Platanios et al., 2019), for the domain adaptation scenario, we measure difficulty based on the distance from in-domain data. Compared to previous work, our application of curriculum learning mainly focuses on improvements on translation quality without consideration of convergence speed.

[12] In Figure 8, for the purpose of fair comparison, each distribution is defined on the same vocabulary, consisting of the source side vocabulary of TED, patent and Paracrawl data.

Chu and Wang (2018) surveyed recent domain adaptation methods for NMT. In their taxonomy, our workflow in Figure 1 can be considered a hybrid that uses both data-centric and model-centric techniques due to the use of additional unlabeled-domain data, with a modified training procedure based for continued training.

For data-centric domain adaptation methods, our curriculum learning approach has connections to instance weighting. In our work, the presentation of certain examples at specific training phases is equivalent to up-weighting those examples and down-weight the others at that time. Weights of similar samples and less similar ones are adjusted dynamically during the training of NMT models based on the curriculum training strategy. In NMT, instance weighting is usually implemented by modifying the objective function (Chen and Huang, 2016; Wang et al., 2017; Chen et al., 2017). In statistical machine translation, Matsoukas et al. (2009) extract features from sentences to capture their domains and then use a classifier to map features to sentence weights. Foster et al. extend this method by weighting at the level of phrase pairs. Shah et al. (2010) use resampling to weight corpora and alignments. Mansour and Ney (2012) focus on sentence-level weighting for phrase extraction. Zhou et al. (2015) weight examples based on their word distributions.

For model-centric domain adaptation methods, our work is related to van der Wees et al. (2017). They adopt gradual fine-tuning, which does the opposite of our method: training starts from the whole dataset, and the training set gradually decreases by removing less similar sentences. Wang et al. (2018) use a similar approach, where the NMT model is trained on progressively noise-reduced data batches. However, such schedules have the risk of wasting computation on non-relevant data, especially when most of the Paracrawl data is not similar to the target domain.

6 Conclusion

We introduced a curriculum learning approach to adapt neural machine translation models to new domains. Our approach first ranks unlabeled-domain training samples based on their similarity to in-domain data, and then adopts a probabilistic curriculum learning strategy so that more similar samples are used earlier and more frequently during training.

We show the effectiveness of our method on four tasks. Results show that curriculum learning models can improve over the standard continued training model by up to 3.22 BLEU points and can take better advantage of distant and noisy data. According to our S^4 analysis of lexical choice errors, this improvement is mainly due to better scoring of words that acquire a new SENSE or have a different SCORE distribution in the new domain. Our extensive empirical analysis suggests that this approach is effective for several reasons: (1) It provides a robust way to augment the training data with samples that have different levels of similarity to the in-domain data. Unlabeled-domain data such as webcrawls naturally have a diverse set of sentences, and the probabilistic curriculum allows us to exploit as much diversity as possible. (2) It implements the intuition that samples more similar to in-domain data are seen earlier and more frequently; when adding a new shard into the training set, the previously visited shards are still used, so the model will not forget what it just learned. (3) It builds on a strong continued training baseline, which continues on in-domain data. (4) The method implements best practices that have shown to be helpful in NMT, e.g. bucketing, mini-batching, and data shuffling.

For future work, it would be interesting to measure how curriculum learning models perform on the general domain test set (rather than the in-domain test set we focus on in this work); do they suffer more or less from catastrophic forgetting (Goodfellow et al., 2014; Kirkpatrick et al., 2017; Khayrallah et al., 2018; Thompson et al., 2019)?

Acknowledgments

This work is supported in part by a AWS Machine Learning Research Award and a grant from the Office of the Director of National Intelligence, Intelligence Advanced Research Projects Activity (IARPA), via contract FA8650-17-C-9115. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the sponsors.

We thank the organizers and participants of the 2018 Machine Translation Marathon for providing a productive environment to start this project. We also thank Amittai Axelrod, Hongyuan Mei and all the team members of the JHU SCALE 2018 workshop for helpful discussions.

References

Amittai Axelrod. 2017. Cynical selection of language model training data. *arXiv preprint arXiv: 1709.02279*.

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the conference on empirical methods in natural language processing*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the 3rd International Conference on Learning Representations*.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, et al. 2017. Findings of the 2017 conference on machine translation (wmt17). In *Proceedings of the Second Conference on Machine Translation*.

Boxing Chen, Colin Cherry, George Foster, and Samuel Larkin. 2017. Cost weighting for neural machine translation domain adaptation. In *Proceedings of the First Workshop on Neural Machine Translation*.

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*.

Chenhui Chu, Raj Dabre, and Sadao Kurohashi. 2017. An empirical comparison of domain adaptation methods for neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*.

Chenhui Chu and Rui Wang. 2018. A survey of domain adaptation for neural machine translation. *arXiv preprint arXiv:1806.00258*.

Kevin Duh. 2018. The multitarget ted talks task. http://www.cs.jhu.edu/~kevinduh/a/multitarget-tedtalks/.

Kevin Duh, Graham Neubig, Katsuhito Sudoh, and Hajime Tsukada. 2013. Adaptation data selection using neural language models: Experiments in machine translation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*.

Chris Dyer, Victor Chahuneau, and Noah A Smith. 2013. A simple, fast, and effective reparameterization of ibm model 2. In *Proceedings of the 2013*

Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.

George Foster, Cyril Goutte, and Roland Kuhn. Discriminative instance weighting for domain adaptation in statistical machine translation. In *Proceedings of the 2010 conference on empirical methods in natural language processing*.

Markus Freitag and Yaser Al-Onaizan. 2016. Fast domain adaptation for neural machine translation. *arXiv preprint arXiv:1612.06897*.

Ian J Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. 2014. An empirical investigation of catastrophic forgetting in gradient-based neural networks. In *Proceedings of the 2nd International Conference on Learning Representations*.

Alex Graves, Marc G Bellemare, Jacob Menick, Remi Munos, and Koray Kavukcuoglu. 2017. Automated curriculum learning for neural networks. In *Proceedings of the 34th International Conference on Machine Learning*.

Kenneth Heafield. 2011. Kenlm: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*.

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

Ann Irvine, John Morgan, Marine Carpuat, Hal Daumé III, and Dragos Munteanu. 2013. Measuring machine translation errors in new domains. *Transactions of the Association for Computational Linguistics*.

Marcin. Junczys-Dowmunt, Bruno. Pouliquen, and Christophe. Mazenc. 2016. Coppa v 2 . 0 : Corpus of parallel patent applications building large parallel corpora with gnu make.

Huda Khayrallah, Brian Thompson, Kevin Duh, and Philipp Koehn. 2018. Regularized training objective for continued training for domain adaptation in neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations*.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*.

Tom Kocmi and Ondřej Bojar. 2017. Curriculum learning and minibatch bucketing in neural machine translation. In *Proceedings of the International Conference Recent Advances in Natural Language Processing*.

Philipp Koehn, Kevin Duh, and Brian Thompson. 2018. The jhu machine translation systems for wmt 2018. In *Proceedings of the Third Conference on Machine Translation*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*.

Gaurav Kumar, George Foster, Colin Cherry, and Maxim Krikun. 2019. Reinforcement learning based curriculum optimization for neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*.

Pierre Lison and Jörg Tiedemann. 2016. Opensubtitles2016: Extracting large parallel corpora from movie and tv subtitles.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*.

Saab Mansour and Hermann Ney. 2012. A simple and effective weighted phrase extraction for machine translation adaptation. In *International Workshop on Spoken Language Translation*.

Spyros Matsoukas, Antti-Veikko I Rosti, and Bing Zhang. 2009. Discriminative corpus weight estimation for machine translation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*.

Robert C Moore and William Lewis. 2010. Intelligent selection of language model training data. In *Proceedings of the ACL 2010 conference short papers*.

Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabas Poczos, and Tom M. Mitchell. 2019. Competence-based curriculum learning for neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Lucía Santamaría and Amittai Axelrod. 2017. Data selection with cluster-based language difference models and cynical selection. In *International Workshop on Spoken Language Translation*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*.

Kashif Shah, Loïc Barrault, and Holger Schwenk. 2010. Translation model adaptation by resampling. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*.

Brian Thompson, Jeremy Gwinnup, Huda Khayrallah, Kevin Duh, and Philipp Koehn. 2019. Overcoming catastrophic forgetting during domain adaptation of neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*.

Rui Wang, Masao Utiyama, Lemao Liu, Kehai Chen, and Eiichiro Sumita. 2017. Instance weighting for neural machine translation domain adaptation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*.

Wei Wang, Taro Watanabe, Macduff Hughes, Tetsuji Nakagawa, and Ciprian Chelba. 2018. Denoising neural machine translation training with trusted data and online data selection. In *Proceedings of the Third Conference on Machine Translation: Research Papers*.

Marlies van der Wees, Arianna Bisazza, and Christof Monz. 2017. Dynamic data selection for neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*.

Daphna Weinshall and Gad Cohen. 2018. Curriculum learning by transfer learning: Theory and experiments with deep networks. In *Proceedings of the 35th International Conference on Machine Learning*.

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*.

Xuan Zhang, Gaurav Kumar, Huda Khayrallah, Kenton Murray, Jeremy Gwinnup, Marianna J Martindale, Paul McNamee, Kevin Duh, and Marine Carpuat. 2018. An empirical exploration of curriculum learning for neural machine translation. *arXiv preprint arXiv:1811.00739*.

Xinpeng Zhou, Hailong Cao, and Tiejun Zhao. 2015. Domain adaptation for smt using sentence weight. In *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*.

A In-domain Data Details

Dataset	TED(de)	TED(ru)	patent(de)	patent(ru)
#samples	151,627	180,316	821,267	28,536

Table 3: In-domain data statistics.

The total amount of the in-domain data in each domain is summarized in Table 3. In this paper, we uniformly sample 15k in-domain data from each dataset. We choose the amount of 15k, which makes up a relatively small percentage of the original corpora, in order to evaluate the extreme case of low-resource domain adaptation settings. Under this setting, the positive effect of adding more selected unlabeled-domain data into training corpus is more obvious in terms of the performance improvement of NMT models. Our pilot experiments show that curriculum learning can scale with more in-domain data—it consistently outperforms the standard training policy, but with less improvement. This is not surprising, as when there is enough in-domain data, continued training on only the in-domain data can already achieve a pretty good performance, and we do not need to use extra unlabeled-domain data to augment it any more, neither does curriculum learning.

B Data Sharding

We experimented with different number of shards for curriculum learning models as shown in Figure 9. Overall, the performance shows the tendency to first improve and then degrade as the number of shards increases. Consider the extreme case, where the data are all put into one shard, or there are as many shards as samples, then it will actually be the same as the standard continued training.

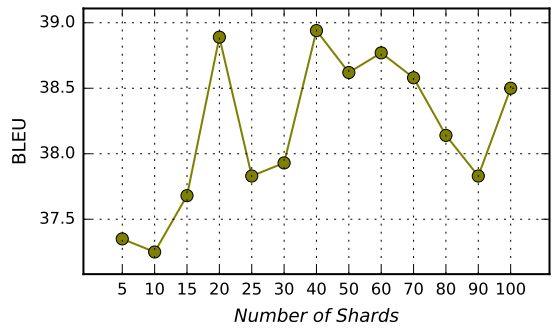

Figure 9: Tuning number of shards on a curriculum learning model trained with German-English corpus augmented by 1024k Paracrawl data.

C S^4 Error Analysis Algorithm

The algorithm for getting S^4 word translation error statistics is shown in 1.

Algorithm 1 S^4 Error Analysis

▷S: The source-side sentences in test set
▷f_i: The ith unique word in a sentence
▷$E^r(f_i)$: Words aligned to f_i in the reference translation for test set
▷$E^h(f_i)$: Words aligned to f_i in the output translation for test set
▷$E^t(f_i)$: Words aligned to f_i in the reference translation for training set
procedure S4ERRORCOUNTER(S, E^r, E^h, E^t)
    correct←0; seen←0; sense←0; score←0
    for $s \in S$ **do**
        for $f_i \in s$ **do**
            for $e_j \in E^r(f_i)$ **do**
                if $e_j \in E^h(f_i)$ **then**
                    correct ← correct+1
                else
                    if $f_i \notin E^t$ **then**
                        seen ← seen+1
                    else
                        if $e_j \notin E^t(f_i)$ **then**
                            sense ← sense+1
                        else
                            score ← score+1
                        end if
                    end if
                end if
            end for
        end for
    end for
    return correct, seen, sense, score
end procedure

D Bilingual Criterion

In previous experiments, we only considered the data selection scores obtained from the source side of the corpora. It is very likely that curriculum learning would benefit from also taking into account the features of target side. For Moore-Lewis score, following Axelrod et al. (2011), we sum the scores over each side of the corpus:

$$[H_{I-src}(s) - H_{N-src}(s)] + [H_{I-tr}(s) - H_{N-trg}(s)]. \quad (3)$$

In addition, we also conduct comparison experiments using scores obtained from only the target

side for both of the two data selection methods. Results are shown in Figure 10.

For Moore-Lewis method, in terms of the standard models, scores collected from target side can lead to better translation than source side scores, and bilingual criteria is somewhere in between, for all the sizes of Paracrawl data we experimented with. But this does not map to the curriculum learning models perfectly. Although CL_en achieves several impressive BLEU scores (39.35 BLEU at 512k, 39.26 BLEU at 2048k), CL_de can sometimes outperform it. And the performance of their bilingual counterparts are unpredictable: it can be either worse or better than both of them. At 4096k ML selected sentences, CL_bi improves the BLEU score to 39.37 BLEU, which is the best test score among all the results we have for German-English TED. For cynical data selection, it is obvious that curriculum learning models prefer the scores obtained from the source side of the corpus.

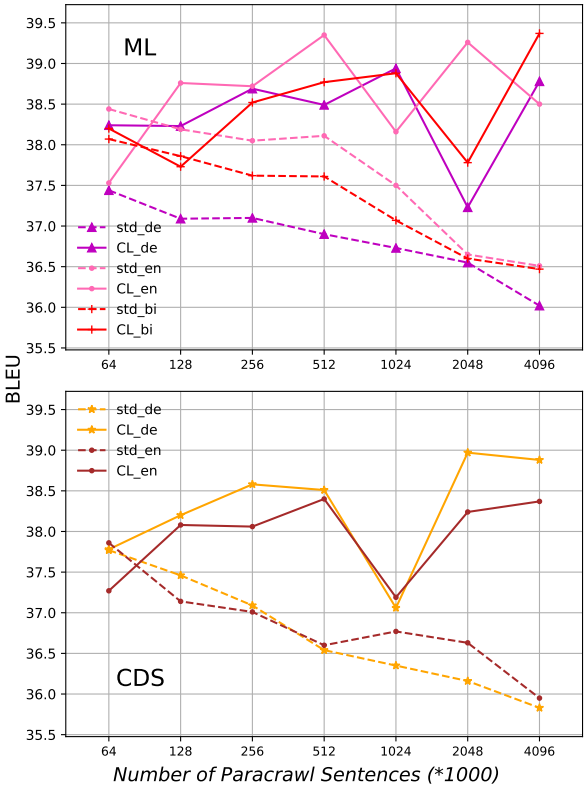

Figure 10: Performance of models continued trained with TED data and Paracrawl data, ranked by their similarity scores collected from the source side (de) or the target side (en) or both sides (bi) of the sentence pairs.

E Perplexity Selection

In Section 3.3, we train models with Paracrawl data of different sizes, only after the training is finished and we get the decoding results of those NMT systems, as shown in Figure 2, can we know which size should be the best choice. Moore and Lewis (2010) proposed a method that can determine the count cut-offs of the selected data beforehand, so that a lot of time and computation will be saved. In their work, the optimal selection threshold is determined by the perplexity of in-domain set evaluated on the language models trained on the different-size selected subsets. We name this method as perplexity selection and we are curious whether it is effective in the NMT settings. Unfortunately, the best thresholds elected by this method (Figure 11) are inconsistent with the cutoffs that achieve high BLEU scores in Figure 2. We can then conclude that perplexity selection may not be an appropriate way to determine the optimal amount of unlabeled-domain data to use for NMT models.

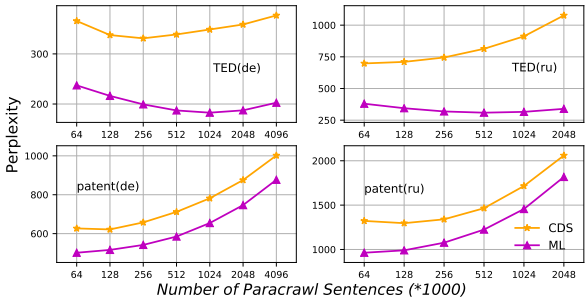

Figure 11: Perplexity of selected data evaluated on the language model learned from in-domain corpus.

However, if computational resources are limited, according to the experiment results (Figure 2) in our work, we recommend 1024k as the first choice for cutoffs on ranked unlabeled-domain data, for NMT domain adaptation models trained with curriculum learning strategy.