Dynamic Data Selection for Neural Machine Translation

Marlies van der Wees
Informatics Institute
University of Amsterdam

Arianna Bisazza[*]
LIACS
Leiden University

Christof Monz
Informatics Institute
University of Amsterdam

Abstract

Intelligent selection of training data has proven a successful technique to simultaneously increase training efficiency and translation performance for phrase-based machine translation (PBMT). With the recent increase in popularity of neural machine translation (NMT), we explore in this paper *to what extent* and *how* NMT can also benefit from data selection. While state-of-the-art data selection (Axelrod et al., 2011) consistently performs well for PBMT, we show that gains are substantially lower for NMT. Next, we introduce *dynamic data selection* for NMT, a method in which we vary the selected subset of training data between different training epochs. Our experiments show that the best results are achieved when applying a technique we call *gradual fine-tuning*, with improvements up to +2.6 BLEU over the original data selection approach and up to +3.1 BLEU over a general baseline.

1 Introduction

Recent years have shown a rapid shift from phrase-based (PBMT) to neural machine translation (NMT) (Sutskever et al., 2014; Cho et al., 2014; Bahdanau et al., 2014) as the most common machine translation paradigm. With large quantities of parallel data, NMT outperforms PBMT for an increasing number of language pairs (Bojar et al., 2016). Unfortunately, training an NMT model is often a time-consuming task, with training times of several weeks not being unusual.

Despite its training inefficiency, most work in NMT greedily uses all available training data for a given language pair. However, it is unlikely that all data is equally helpful to create the best-performing system. In PBMT, this issue has been addressed by applying intelligent data selection, and it has consistently been shown that using more data does not always improve translation quality (Moore and Lewis, 2010; Axelrod et al., 2011; Gascó et al., 2012). Instead, for a given translation task, the training bitext likely contains sentences that are irrelevant or even harmful, making it beneficial to keep only the most relevant subset of the data while discarding the rest, with the additional benefit of smaller models and faster training.

Motivated by the success of data selection in PBMT, we investigate in this paper *to what extent* and *how* NMT can benefit from data selection as well. While data selection has been applied to NMT to reduce the size of the data (Cho et al., 2014; Luong et al., 2015b), the effects on translation quality have not been investigated. Intuitively, and confirmed by our exploratory experiments in Section 5.1, this is a challenging task; NMT systems are known to under-perform when trained on limited parallel data (Zoph et al., 2016; Fadaee et al., 2017), and do not have a separate large-scale target-side language model to compensate for smaller parallel training data.

To alleviate the negative effect of small training data on NMT, we introduce *dynamic data selection*. Following conventional data selection, we still dramatically reduce the training data size, favoring parts of the data which are most relevant to the translation task at hand. However, we exploit the fact that the NMT training process iterates over the training corpus in multiple epochs, and we alter the quantity or the composition of the training data *between epochs*. The proposed method requires no modifications to the NMT architecture or parameters, and substantially speeds up training times while simultaneously improving translation quality with respect to a complete-bitext baseline.

[*] Work done while at University of Amsterdam

Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 1400–1410
Copenhagen, Denmark, September 7–11, 2017. ©2017 Association for Computational Linguistics

In summary, our contributions are as follows:

(i) We compare the effects of a commonly used data selection approach (Axelrod et al., 2011) on PBMT and NMT using four different test sets. We find that this method is much less effective for NMT than for PBMT, while using the exact same training data subsets.

(ii) We introduce dynamic data selection as a way to make data selection profitable for NMT. We explore two techniques to alter the selected data subsets, and find that our method called *gradual fine-tuning* improves over conventional static data selection (up to +2.6 BLEU) and over a high-resource general baseline (up to +3.1 BLEU). Moreover, gradual fine-tuning approximates in-domain fine-tuning in ∼20% of the training time, even when no parallel in-domain data is available.

2 Static data selection

As a first step towards dynamic data selection for NMT, we compare the effects of a commonly used, state-of-the-art data selection method (Axelrod et al., 2011) on both neural and phrase-based MT. Briefly, this approach ranks sentence pairs in a large training bitext according to their difference in cross-entropy with respect to an in-domain corpus (i.e., a corpus representing the test data) and a general corpus. Next, the top n sentence pairs with the highest rank—thus lowest cross-entropy—are selected and used for training an MT system.

Formally, given an in-domain corpus I, we first create language models from the source side f of I ($LM_{I,f}$) and the target side e of I ($LM_{I,e}$). We then draw a random sample (similar in size to I) of the large general corpus G and create language models from the source and target sides of G: $LM_{G,f}$ and $LM_{G,e}$, respectively. Note that the data for creating these LMs need not be parallel but can be independent corpora in both languages.

Next, we compute for each sentence pair s in G four cross-entropy scores, defined as:

$$H_{C,s_b} = -\sum p\left(s_b\right) \log \left(LM_{C,b}\left(s_b\right)\right), \qquad (1)$$

where $C \in \{I, G\}$ is the corpus, $b \in \{f, e\}$ refers to the bitext side, and s_b is the bitext side b of sentence pair s in the parallel training corpus.

To find sentences that are similar to the in-domain corpus, i.e., have low H_I, and at the same time dissimilar to the general corpus, i.e., have high H_G, we compute for each sentence pair s the bilingual cross-entropy difference CED_s following Axelrod et al. (2011):

$$CED_s = (H_{I,s_f} - H_{G,s_f}) + (H_{I,s_e} - H_{G,s_e}). \qquad (2)$$

Finally, we rank all sentence pairs $s \in G$ according to their CED_s, and then select only the top n sentence pairs with the lowest CED_s.

Following related work by Moore and Lewis (2010), we restrict the vocabulary of the LMs to the words occurring at least twice in the in-domain corpus. To analyze the quality of the selected data subsets, we also run experiments on random selections, all performed in threefold. Finally, we always use the exact same selection of sentence pairs in equivalent PBMT and NMT experiments.

LSTM versus n-gram The described data selection method uses n-gram LMs to determine the domain-relevance of sentence pairs. We adhere to this setting for our comparative experiments on PBMT and NMT (Section 5.1). However, when applying data selection to NMT, we examine the potential benefit of replacing the conventional n-gram LMs with LSTMs[1]. These have the advantage to remember longer histories, and do not have to back off to shorter histories when encountering out-of-vocabulary words. In this neural variant to rank sentences, the score for each sentence pair in G is still computed as the bilingual cross-entropy difference in Equation (2). In addition, we use the same in-domain and general corpora as with the n-gram method, and we again restrict the vocabulary to the most frequent words.

3 Dynamic data selection

While data selection aims to discard irrelevant data, it can also exacerbate the problem of low vocabulary coverage and unreliable statistics for rarer words in the 'long tail', which are major issues in NMT (Luong et al., 2015b; Sennrich et al., 2016b). In addition, it has been shown that NMT performance drops tremendously in low-resource scenarios (Zoph et al., 2016; Fadaee et al., 2017; Koehn and Knowles, 2017).

To overcome this problem, we introduce *dynamic data selection*, in which we vary the selected data subsets *during* training. Unlike other MT paradigms, which require training data to be fixed during the entire training process, NMT iterates over the training corpus in several epochs,

[1] We use four-layer LSTMs with embedding and hidden sizes of 1,024, which we train for 30 epochs.

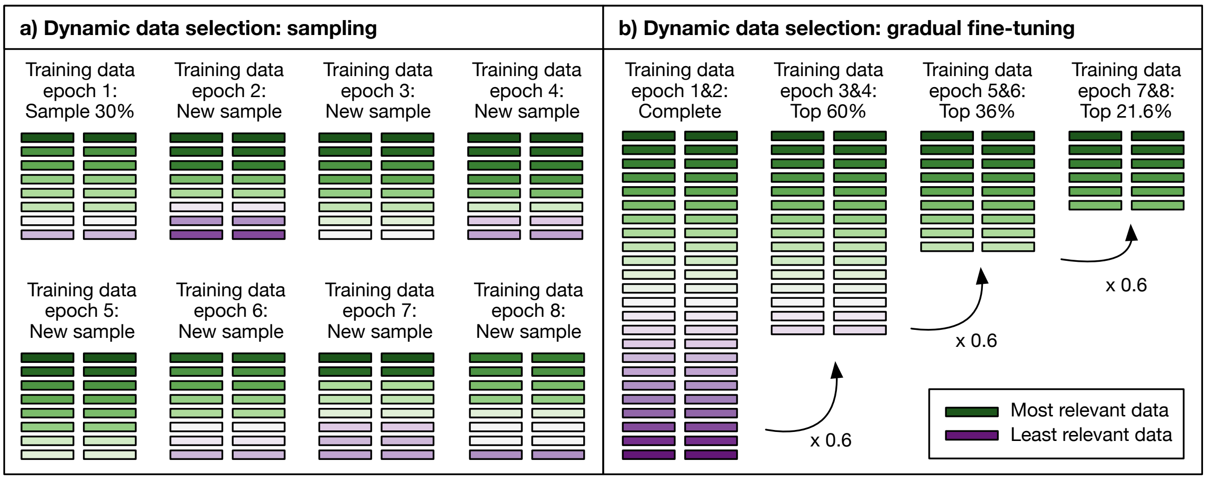
a) Dynamic data selection: sampling
Training data epoch 1: Sample 30%
Training data epoch 2: New sample
Training data epoch 3: New sample
Training data epoch 4: New sample
Training data epoch 5: New sample
Training data epoch 6: New sample
Training data epoch 7: New sample
Training data epoch 8: New sample
b) Dynamic data selection: gradual fine-tuning
Training data epoch 1&2: Complete
Training data epoch 3&4: Top 60%
Training data epoch 5&6: Top 36%
Training data epoch 7&8: Top 21.6%
x 0.6
x 0.6
x 0.6
Most relevant data
Least relevant data

Figure 1: Illustration of two dynamic bitext selection techniques for NMT: *sampling* (left) and *gradual fine-tuning* (right). Measured over 16 training epochs (which is used in this work), the total training time of both examples would be ∼30% of the training time needed when using the complete bitext.

allowing to use a different subset of the training data in every epoch.

Dynamic data selection starts from a relevance-ranked bitext, which we create using CED scores as computed in Equation (2). Given this ranking, we investigate two dynamic data selection techniques[2] that vary per epoch the composition or the size of the selected training data. Both techniques aim to favor highly relevant sentences over less relevant sentences while not completely discarding the latter. In all experiments, we use a fixed vocabulary created from the complete bitext.

While we use in this work a domain-relevance ranking of the bitext following Axelrod et al. (2011), dynamic data selection can also be applied using other ranking criteria, for example limiting redundancy in the training data (Lewis and Eetemadi, 2013) or complementing similarity with diversity (Ruder and Plank, 2017).

Sampling sentence pairs In the first technique, illustrated in Figure 1a, we sample for every epoch n sentence pairs from G, using a distribution computed from the domain-specific CED_s scores. Concretely, this is done as follows:

First, since higher ranked sentence pairs have lower CED_s scores, and they can be either negative or positive, we scale and invert CED_s scores such that $0 \leq CED'_s \leq 1$ for each sentence pair $s \in G$:

$$CED'_s = 1 - \frac{CED_s - \min(CED_G)}{\max(CED_G) - \min(CED_G)}, \qquad (3)$$

where CED_G refers to the set of CED_s scores for bitext G.

Next, we convert CED'_s scores to relative weights, such that $\sum_{s \in G} w(s) = 1$:

$$w(s) = \frac{CED'_s}{\sum_{s_i \in G} CED'_{s_i}}. \qquad (4)$$

We then use $\{w(s) : s \in G\}$ to perform weighted sampling, drawing for each epoch n sentence pairs without replacement. While all selection weights are very close to zero, higher ranked sentences have a noticeably higher probability of being selected than lower-ranked sentences; in practice we find that top-ranked sentences get selected in nearly each epoch, while bottom-ranked sentence pairs get selected at most once. Note that the sampled selection for any epoch is independent of selections for all other epochs.

Gradual fine-tuning The second dynamic data selection technique, see Figure 1b, is inspired by the success of domain-specific fine-tuning (Luong and Manning, 2015; Zoph et al., 2016; Sennrich et al., 2016a; Freitag and Al-Onaizan, 2016), in which a model trained on a large general-domain bitext is trained for a few additional epochs only on small in-domain data. However, rather than training a full model on the complete bitext G, we gradually decrease the training data size, starting from G and keeping only the top n sentence pairs for the duration of η epochs, where the top n pairs are defined by their CED_s scores. Given its resemblance to fine-tuning, we refer to this variant as *gradual fine-tuning*.

[2] Code for bitext ranking and both selection techniques: github.com/marliesvanderwees/dds-nmt.

During gradual fine-tuning, the selection size n is a function of epoch i:

$$n(i) = \alpha \cdot |G| \cdot \beta^{\lfloor (i-1)/\eta \rfloor}. \quad (5)$$

Here $0 \leq \alpha \leq 1$ is the *relative start size*, i.e., the fraction of general bitext G used for the first selection, $0 \leq \beta \leq 1$ is the *retention rate*, i.e., the fraction of data to be kept in each new selection, and $\eta \geq 1$ is the number of consecutive epochs each selected subset is used. Note that $\lfloor i/\eta + 1 \rfloor$ indicates rounding down $i/\eta + 1$ to the nearest integer. For example, if we start with the complete bitext ($\alpha = 1$), select the top 60% ($\beta = 0.6$) every second epoch ($\eta = 2$), then we run epochs 1 and 2 with a subset of size $|G|$, epochs 3 and 4 with a subset of size $0.6 \cdot |G|$, epochs 5 and 6 with a subset of size $0.36 \cdot |G|$, and so on. For every size n, the actual selection contains the top n sentences pairs of G.

4 Experimental settings

We evaluate static and dynamic data selection on a German→English translation task comprising four test sets. Below we describe the MT systems and data specifications.

4.1 Machine translation systems

While the main aim of this paper is to improve data selection for NMT, we also perform comparative experiments using PBMT. Our PBMT system is an in-house system similar to Moses (Koehn et al., 2007). To create optimal PBMT systems given the available resources, we apply test-set-specific parameter tuning using PRO (Hopkins and May, 2011). In addition, we use a linearly interpolated target-side language model trained with Kneser-Ney smoothing on 480M tokens of data in various domains. LM interpolation weights are also optimized per test set. Consistent with Axelrod et al. (2011), we do not vary the target-side LM between different experiments on the same test set. All n-gram models in our work are 5-gram.

For our NMT experiments we use an in-house encoder-decoder[3] model with global attention as described in Luong et al. (2015a). This choice comes at the cost of optimal translation quality but allows for a relatively fast realization of large-scale experiments given our available resources. Both the encoder and decoder are four-layer unidirectional LSTMs, with embedding and layer sizes of 1,000. We uniformly initialize all parameters, and use SGD with a mini-batch size of 64 and an initial learning rate of 1, which is decayed by a factor two every epoch after the fifth epoch. We use dropout with probability 0.3, and a beam size of 12. We train for 16 epochs and test on the model from the last epoch. All NMT experiments are run on a single NVIDIA Titan X GPU.

	Train		Dev/valid		Test	
Corpus	Lines	Tokens	Lines	Tokens	Lines	Tokens
EMEA	206K	3.3M	3.9K	59K	5.8K	93K
Movies	101K	1.2M	4.5K	54K	7.1K	87K
TED	189K	3.3M	2.5K	50K	5.4K	99K
WMT	3.8M	84M	3.0K	64K	3.0K	65K
Mix	4.3M	92M	3.5K	61K	–	–

Table 1: Data specifications with tokens counted on the German side. The WMT training corpus contains Commoncrawl, Europarl, and News Commentary but no in-domain news data.

4.2 Training and evaluation data

We evaluate all experiments on four domains: (i) EMEA medical guidelines (Tiedemann, 2009), (ii) movie dialogues (van der Wees et al., 2016) constructed from OpenSubtitles (Lison and Tiedemann, 2016), (iii) TED talks (Cettolo et al., 2012), and (iv) WMT news. For TED, we use IWSLT2010 as development set and IWSLT2011-2014 as test set, and for WMT we use newstest2013 as development set and newstest2016 as test set. We train our systems on a mixture of domains, comprising Commoncrawl, Europarl, News Commentary, EMEA, Movies, and TED. Corpus specifications are listed in Table 1.

The in-domain LMs used to rank training sentences for data selection are trained on small portions of in-domain parallel data whenever available (3.3M, 1.2M and 3.3M German tokens for EMEA, Movies and TED, respectively). Since no sizeable in-domain parallel text is available for WMT, we independently sample 200K sentences from the WMT monolingual News Crawl corpora (3.3M German tokens or 3.5M English tokens). This demonstrates the applicability of data selection techniques even in cases where one lacks parallel in-domain data.

Before running data selection, we preprocess our data by tokenizing, lowercasing and remov-

[3] github.com/ketranm/tardis

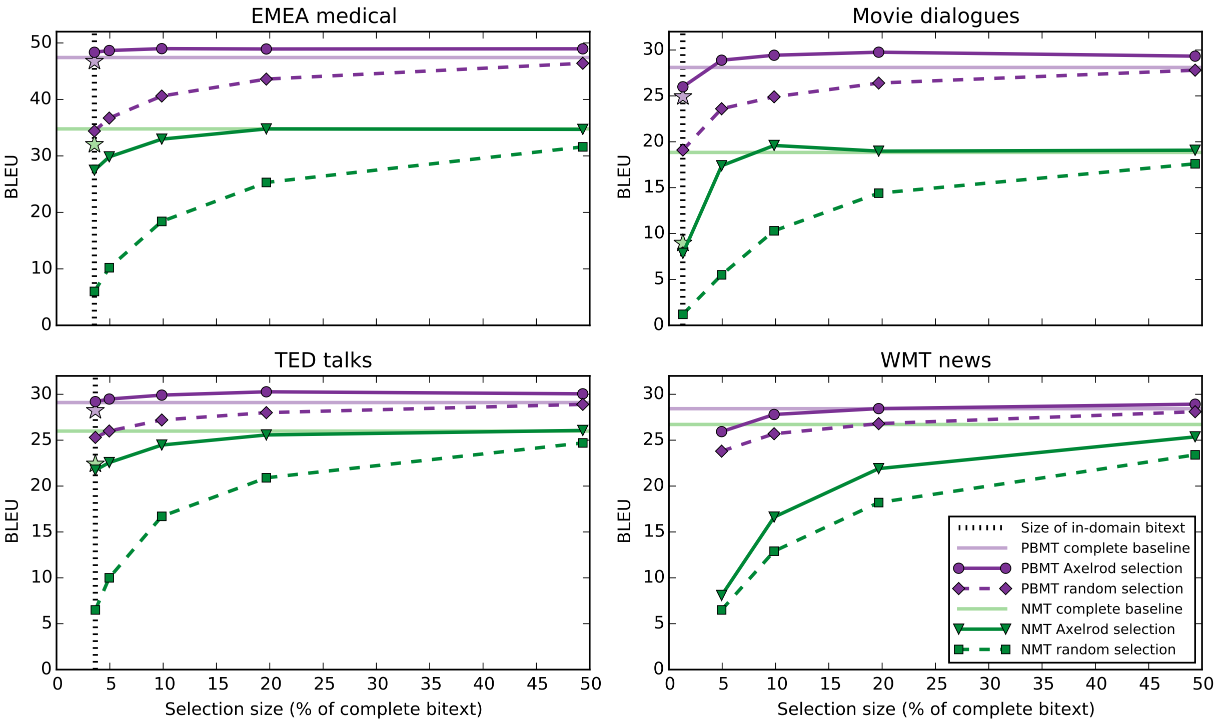

Figure 2: PBMT (purple) and NMT (green) German→English results of Axelrod data selection and random data selection (average of three runs) for four domains. Purple and green stars indicate BLEU scores when only the available in-domain data is used. We use selections of the in-domain size $|I|$, and 5%, 10%, 20%, and 50% of the complete bitext, which are exactly the same for PBMT and NMT.

ing sentences that are longer than 50 tokens or that are identified as a different language. After selection, we apply Byte-pair encoding (BPE, Sennrich et al. (2016b)) with 40K merge operations on either side of the complete mix-of-domains training bitext. For our NMT experiments we use BPE-processed corpora on both bitext sides, while for PBMT we only apply BPE to the German side. Our NMT systems use a vocabulary size of 40K on both the source and target side.

5 Results

Below we discuss the results of our translation experiments using static and dynamic data selection, measuring translation quality with case-insensitive untokenized BLEU (Papineni et al., 2002).

5.1 Static data selection for PBMT and NMT

We first compare the effects of static data selection with n-gram LMs on both NMT and PBMT using various selection sizes. Concretely, we select the top n sentence pairs such that the number of selected tokens $t \in \{5\%, 10\%, 20\%, 50\%\}$ of G, or $t = |I|$ (the in-domain corpus size). Figure 2 shows German→English translation performance in BLEU for our four test sets. The benefits of n-gram-based data selection for PBMT (purple circles) are confirmed: In all test sets, the selection of size $|I|$ (dotted vertical line) yields better performance than using only the in-domain data of the exact same size (purple star), and at least one of the selected subsets—often using only 5% of the complete bitext—outperforms using the complete bitext (light purple line). We also show that the informed selections are superior to random selections of the same size (purple diamonds).

In NMT, results of n-gram-based data selection (green triangles) vary: While for Movies a selection of only 10% outperforms the complete bitext (light green line), none of the selected subsets for other test sets is noticeably better than the full bitext.[4] Interestingly, the same selections of size $|I|$ that proved useful in PBMT, never beat the system that uses exactly the available in-domain data (green star), indicating that the current selections can be further improved for NMT. In all scenarios we see that NMT suffers much more from small-data settings than PBMT. Finally, the random se-

[4] Validation cross-entropy converges after 10–12 epochs, never reaching the scores of the complete bitext.

lections (green squares) show that NMT not only needs large quantities of data, but it is also affected when the selected data is of low quality. In PBMT, both low-quantity and low-quality scenarios appear to be compensated for by the large monolingual LM on the target side.

When comparing the different test sets, we observe that the impact of domain mismatch in NMT with respect to PBMT is largest for the two domains that are most distinct from the general bitext, EMEA and Movies. For WMT, both MT systems achieve very similar baseline results, but translation quality deteriorates considerably in data selection experiments, which is likely caused by the lack of in-domain data in the general bitext.

LSTM versus n-gram Before proceeding with dynamic data selection for NMT, we test whether bitext ranking for NMT can be improved using LSTMs rather than conventional n-gram LMs. Table 2 shows NMT BLEU scores of a few different sizes of selected subsets created using n-gram LMs or LSTMs. While results vary among test sets and selection sizes, we observe an average improvement of 0.4 BLEU when using LSTMs instead of n-gram LMs. For PBMT, similar results have been reported when replacing n-gram LMs with recurrent neural LMs (Duh et al., 2013). In all subsequent experiments we use relevance rankings computed with LSTMs instead of n-gram LMs.

Selection	LM type	EMEA	Movies	TED	WMT
5%	n-gram	29.8	17.4	22.6	8.1
	LSTM	**30.0**	**17.8**	22.6	**9.6**
10%	n-gram	33.0	19.6	24.5	16.6
	LSTM	33.0	**19.7**	**24.7**	**17.4**
20%	n-gram	**34.8**	19.0	25.6	21.9
	LSTM	34.5	**19.6**	**26.6**	21.9

Table 2: NMT BLEU comparison between using n-gram LMs and LSTMs for bitext ranking. Selection sizes concern the selected bitext subsets; LMs are created from the exact same in-domain data.

5.2 Dynamic data selection for NMT

Equipped with a relevance ranking of sentence pairs in bitext G, we now examine two variants of dynamic data selection as described in Section 3.

We are interested in reducing training time while limiting the negative effect on BLEU for various domains. Therefore we report BLEU as well as the *relative training time* of each experiment. Since wall-clock times depend on other factors such as the NMT architecture and memory speed, we define training time as the total number of tokens observed while training the NMT system, i.e., the sum of tokens in the selected subsets of all epochs. We report all training times relative to the training time of our complete-bitext baseline (i.e., 4.3M tokens × 16 epochs). Note that this measure of training time corresponds closely but not exactly to the number of model updates, as the latter relies on the number of sentences, which vary in length, rather than the number of tokens in the training data. For completeness: Training the 100% baseline takes 106 hours, while our fastest dynamic selection variant takes 19–21 hours. Computing CED scores takes ∼15 minutes when using n-gram LMs and 5–6 hours when using LSTMs.

Figure 3 shows BLEU scores of some selected experiments as a function of relative training time. Compared to static data selection (blue lines), our weighted sampling technique (orange triangles) yields variable results. When sampling a subset of 20% of $|G|$ from the top 50% of the ranked bitext, we obtain small improvements for TED and WMT, but small drops for EMEA and Movies. Other selection sizes (30% and 40%, not shown) give similar results lacking a consistent pattern.

By contrast, our gradual fine-tuning method performs consistently better than static selection, and even beats the general baseline in three out of four test sets. The displayed version uses settings ($\alpha = 0.5, \beta = 0.7, \eta = 2$) and is at least as fast as static selection using 20% of the bitext, yielding up to +2.6 BLEU improvement (for WMT news) over this static version. Compared to the complete baseline, this gradual fine-tuning method improves up to +3.1 BLEU (for TED talks).

Table 3 provides detailed information on additional experiments using other settings. For all three test domains which are covered in the parallel data—EMEA, Movies and TED—improvements are highest when starting gradual fine-tuning with only the top 50% of the ranked bitext, which are also the fastest approaches. For WMT, which is not covered in the general bitext, adding more data clearly benefits translation quality. These findings are consistent with the static data selection patterns; Using low-ranked sentences on top of the most relevant selection

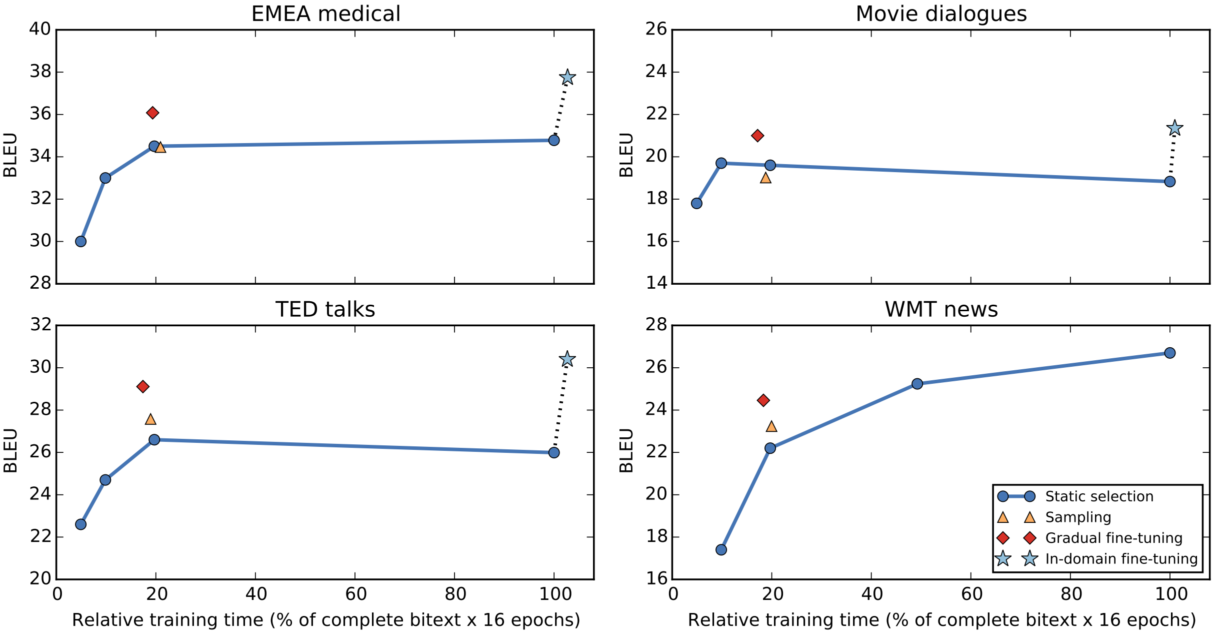

Figure 3: Selected German→English translation results of dynamic data selection methods (orange and red markers) compared to conventional static data selection (blue circles). *Relative training time* equals the total number of training tokens relative to the complete baseline, which takes 106 hours to train and is represented by the rightmost blue circle. Note that no parallel in-domain data is available for WMT news. All y-axes are scaled equally for easy comparison of BLEU differences across domains.

Experiment			Relative training time	BLEU			
Start size	Retention rate β	Decrease every		EMEA	Movies	TED	WMT
Static selection top 20%			20%	34.5	19.6	26.6	21.9
50% ($\alpha = 0.5$)	0.7	$\eta = 2$ epochs	18–20%	**36.1 (+1.6)**	21.0 (+1.4)	**29.1 (+2.5)**	24.5 (+2.6)
50% ($\alpha = 0.5$)	0.5	$\eta = 4$ epochs	21–23%	36.0 (+1.5)	**21.2 (+1.6)**	29.0 (+2.4)	25.0 (+3.1)
50% ($\alpha = 0.5$)	0.6	$\eta = 4$ epochs	25–27%	35.6 (+1.1)	21.0 (+1.4)	28.5 (+1.9)	25.1 (+3.2)
100% ($\alpha = 1$)	0.6	$\eta = 2$ epochs	29–31%	35.5 (+1.0)	21.1 (+1.5)	29.0 (+2.4)	25.6 (+3.7)
100% ($\alpha = 1$)	0.7	$\eta = 2$ epochs	37–39%	35.9 (+1.4)	20.4 (+0.8)	28.2 (+1.6)	25.8 (+3.9)
100% ($\alpha = 1$)	0.9	$\eta = 1$ epoch	50–52%	35.4 (+0.9)	19.6 (±0.0)	27.4 (+0.8)	**26.1 (+4.2)**
Complete bitext baseline			100%	34.8	18.8	26.0	26.7
Gold: fine-tuning on in-domain data			101–103%	37.7	21.3	30.4	–

Table 3: German→English BLEU results of various gradual fine-tuning experiments sorted by relative training time. Indicated improvements are with respect to static selection using 20% of the bitext, and highest scores per test set are bold-faced. Results from the first experiment are also shown in Figure 3.

does not improve translation performance for any domain except WMT news.

Finally, we compare our data selection experiments to domain-specific fine-tuning (light blue stars in Figure 3), which is the current state-of-the-art for domain adaptation in NMT. To this end, we first train a model on the complete bitext, and then train for twelve additional epochs on available in-domain data, using an initial learning rate of 1 which halves every epoch. Depending on the test set, this approach yields +2.5–4.4 BLEU improvements over our baselines, however it does not speed up training and requires a parallel in-domain text which may not be available (e.g., for WMT). While none of our data selection experiments outperforms domain-specific fine-tuning, we obtain competitive translation quality in only 20% of the training time. In additional experiments we found that in-domain fine-tuning on top of our selection approaches does not yield improvements.

6 Further analysis

In this section we conduct a few additional experiments and analyses. We restrict to one parameter setting per selection approach: Static selection and sampling with 20% of the data, and gradual fine-tuning using ($\alpha = 0.5, \beta = 0.7, \eta = 2$). All have very similar training times.

First, we hypothesize that dynamic data selection works well because more different sentence pairs are observed during training, and it therefore increases coverage with respect to static data selection. To verify this, we measure for each test set the number of unseen source word types in the training data of different selection methods. Figure 4 shows indeed that the average number of unseen word types is reduced noticeably in both of our dynamic selection techniques, being much closer to the complete bitext baseline than to static selection. Note that all methods use the same vocabulary during training.

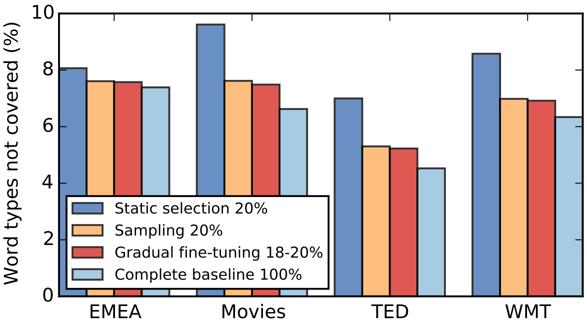

Figure 4: Test set source words not covered in the training data of different data selection methods.

Next, following the static data selection experiments in Section 5.1, we examine how well dynamic data selection performs using random selections. To this end, we repeat all techniques using a bitext which is ranked randomly rather than by its relevance to the test sets. The results in Table 4 show that the bitext ranking plays a crucial role in the success of data selection. However, the results also show that *even* in the absence of an appropriate bitext ranking, dynamic data selection—and in particular gradual fine-tuning—is still superior to static data selection. We explain this result as follows: Compared to static selection, both sampling and gradual fine-tuning have better coverage due to their improved exploration of the data. However, sampling also suffers from a surprise effect of observing new data in every epoch. Gradual fine-tuning on the other hand gradually improves learning on a subset of the selected data, suggesting that repetition across epochs has a positive effect on translation quality.

Ranking	Method	EMEA	Movies	TED	WMT
Relevance	Gradual FT	**36.1**	**21.0**	**29.1**	**24.5**
	Sampling 20%	34.5	19.0	27.6	23.2
	Static 20%	34.5	19.6	26.6	21.9
Random	Gradual FT	**29.2**	**16.1**	**23.2**	**21.3**
	Sampling 20%	26.7	14.4	22.0	19.8
	Static 20%	25.3	14.4	20.9	18.2

Table 4: BLEU scores of data selection using relevance versus random ranking of the bitext. Gradual fine-tuning uses ($\alpha = 0.5, \beta = 0.7, \eta = 2$), with relative training times of 18–20%.

One could expect that changing the data during training results in volatile training behavior. To test this, we inspect cross-entropy of our development sets after every training epoch. Figure 5 shows these results for TED. Clearly, static data selection converges most steadily. However, both dynamic selection techniques eventually converge to a lower cross-entropy value which is reflected by higher translation quality of the test set. We observe very similar behavior for the other test sets.

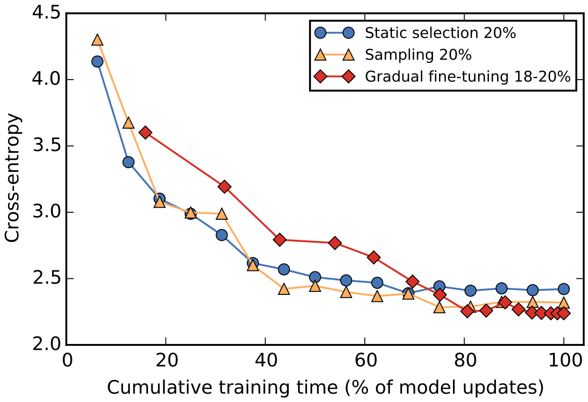

Figure 5: German→English cross-entropy of the TED dev set as a function of training time. Each data point represents a completed training epoch.

By its nature, our gradual fine-tuning technique uses training epochs of different sizes, and therefore also implicitly differs from other methods in its parameter optimization behavior. Since we decrease both the training data size and the SGD learning rate after finishing complete training epochs, we automatically decay the learning rate at decreasing time intervals. We therefore study how this approach is affected when we (i) decay the learning rate after a fixed number of updates (i.e., the same as in static data selection) rather than per epoch, or (ii) keep the learning rate fixed. In the first scenario, we observe that translation performance drops with –1.1–2.0 BLEU. When keeping a fixed learning rate, BLEU scores hardly change or even improve, indicating that the implicit change in search behavior may contribute to the success of gradual fine-tuning.

7 Related work

A few research topics are related to our work. Regarding data selection for SMT, previous work has targeted two goals; to reduce model sizes and training times, or to adapt to new domains. Data selection methods for domain adaptation mostly employ information theory metrics to rank training sentences by their relevance to the domain at hand. This has been applied monolingually (Gao et al., 2002) as well as bilingually (Yasuda et al., 2008). In more recent work, training sentences are typically ranked according to their cross-entropy *difference* between in-domain and general-domain data (Moore and Lewis, 2010; Axelrod et al., 2011, 2015), favoring sentences that are similar to the test domain and at the same time dissimilar from the general domain. Duh et al. (2013) and Chen and Huang (2016) present similar methods in which n-gram LMs are replaced by neural LMs or neural classifiers, respectively.

Data selection with the aim of model size and training time reduction has the objective to use the minimum amount of data while still maintaining high vocabulary coverage (Eck et al., 2005; Gascó et al., 2012; Lewis and Eetemadi, 2013). In a comparative study, Mirkin and Besacier (2014) find that similarity-objected methods perform best if the test domain and general corpus are very different, while a coverage-objected method is superior if test and general corpus are relatively similar. A comprehensive survey on data selection for SMT is provided by Eetemadi et al. (2015). While in this work we have used a similarity objective to rank our bitext, one could also apply dynamic data selection using a coverage objective.

In NMT, data selection can serve similar goals as in PBMT; increasing training efficiency or domain adaptation. Domain adaptation in NMT typically involves training a model on the complete bitext, followed by fine-tuning the parameters on a smaller in-domain corpus (Luong and Manning, 2015; Zoph et al., 2016). Other work combines fine-tuning with model ensembles (Freitag and Al-Onaizan, 2016) or with domain-specific tags in the training corpus (Chu et al., 2017). Finally, Sennrich et al. (2016a) adapt their systems by back-translating in-domain data, which is then added to the training data and used for fine-tuning.

Some other previous work has addressed training efficiency for NMT, for example by parallelizing models or data (Wu et al., 2016), modifying the NMT network structure (Kalchbrenner et al., 2016), decreasing the number of parameters through knowledge distillation (Crego et al., 2016; Kim and Rush, 2016), or by boosting parts of the data that are 'challenging' to the NMT system (Zhang et al., 2016). The latter is most related to our work since training data is also adjusted during training, however we reduce the training data size much more aggressively and study different techniques of data selection.

Finally, recent work comparing various aspects for PBMT and NMT includes (Bentivogli et al., 2016; Farajian et al., 2017; Toral and Sánchez-Cartagena, 2017; Koehn and Knowles, 2017).

8 Conclusions

With the recent increase in popularity of neural machine translation (NMT), we explored in this paper *to what extent* and *how* NMT can benefit from data selection. We first showed that a state-of-the-art data selection method yields unreliable results for NMT while consistently performing well for PBMT. Next, we have introduced *dynamic data selection* for NMT, which entails varying the selected subset of training data between different training epochs. We explored two techniques of dynamic data selection and found that our *gradual fine-tuning* technique, in which we gradually reduce training size, improves consistently over conventional static data selection (up to +2.6 BLEU) and over a high-resource general baseline (up to +3.1 BLEU). Moreover, gradual fine-tuning approximates in-domain fine-tuning using only $\sim$20% of the training time, even when no parallel in-domain data is available.

Acknowledgments

This research was funded in part by NWO under project numbers 639.022.213 and 639.021.646. We thank Ke Tran for providing the NMT system, and the reviewers for their valuable comments.

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.

Amittai Axelrod, Philip Resnik, Xiaodong He, and Mari Ostendorf. 2015. Data selection with fewer words. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 58–65.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Luisa Bentivogli, Arianna Bisazza, Mauro Cettolo, and Marcello Federico. 2016. Neural versus phrase-based machine translation quality: a case study. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 257–267.

Ondrej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, et al. 2016. Findings of the 2016 conference on machine translation (WMT16). In *Proceedings of the first conference on machine translation (WMT16)*.

Mauro Cettolo, Christian Girardi, and Marcello Federico. 2012. Wit^3: Web inventory of transcribed and translated talks. In *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT)*, pages 261–268.

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

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734.

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

Josep Crego, Jungi Kim, Guillaume Klein, Anabel Rebollo, Kathy Yang, Jean Senellart, Egor Akhanov, Patrice Brunelle, Aurelien Coquard, Yongchao Deng, et al. 2016. SYSTRAN's pure neural machine translation systems. *arXiv preprint arXiv:1610.05540*.

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*, pages 678–683.

Matthias Eck, Stephan Vogel, and Alex Waibel. 2005. Low cost portability for statistical machine translation based on n-gram frequency and TF-IDF. In *Proceedings of the 2005 International Workshop on Spoken Language Translation*, pages 61–67.

Sauleh Eetemadi, William Lewis, Kristina Toutanova, and Hayder Radha. 2015. Survey of data-selection methods in statistical machine translation. *Machine Translation*, 29(3-4):189–223.

Marzieh Fadaee, Arianna Bisazza, and Christof Monz. 2017. Data augmentation for low-resource neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*.

M. Amin Farajian, Marco Turchi, Matteo Negri, Nicola Bertoldi, and Marcello Federico. 2017. Neural vs. phrase-based machine translation in a multi-domain scenario. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 280–284.

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

Jianfeng Gao, Joshua Goodman, Mingjing Li, and Kai-Fu Lee. 2002. Toward a unified approach to statistical language modeling for chinese. *ACM Transactions on Asian Language Information Processing (TALIP)*, 1(1):3–33.

Guillem Gascó, Martha-Alicia Rocha, Germán Sanchis-Trilles, Jesús Andrés-Ferrer, and Francisco Casacuberta. 2012. Does more data always yield better translations? In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 152–161.

Mark Hopkins and Jonathan May. 2011. Tuning as ranking. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1352–1362.

Nal Kalchbrenner, Lasse Espeholt, Karen Simonyan, Aaron van den Oord, Alex Graves, and Koray Kavukcuoglu. 2016. Neural machine translation in linear time. *arXiv preprint arXiv:1610.10099*.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327.

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 *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Demo and Poster Sessions*, pages 177–180.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. *arXiv preprint arXiv:1706.03872*.

William D. Lewis and Sauleh Eetemadi. 2013. Dramatically reducing training data size through vocabulary saturation. In *Proceedings of the 8th Workshop on Statistical Machine Translation*, pages 281–291.

Pierre Lison and Jörg Tiedemann. 2016. Opensubtitles2016: Extracting large parallel corpora from movie and tv subtitles. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*.

Minh-Thang Luong and Christopher D Manning. 2015. Stanford neural machine translation systems for spoken language domains. In *Proceedings of the 12th International Workshop on Spoken Language Translation*, pages 76–79.

Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015a. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421.

Minh-Thang Luong, Ilya Sutskever, Quoc Le, Oriol Vinyals, and Wojciech Zaremba. 2015b. Addressing the rare word problem in neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 11–19.

Shachar Mirkin and Laurent Besacier. 2014. Data selection for compact adapted SMT models. In *Proceedings of the 11th Conference of the Association for Machine Translation in the Americas*, pages 301–314.

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.

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

Sebastian Ruder and Barbara Plank. 2017. Learning to select data for transfer learning with bayesian optimization. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 86–96.

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*, pages 1715–1725.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Jörg Tiedemann. 2009. News from OPUS-a collection of multilingual parallel corpora with tools and interfaces. In *Recent advances in natural language processing*, volume 5, pages 237–248.

Antonio Toral and Víctor M. Sánchez-Cartagena. 2017. A multifaceted evaluation of neural versus phrase-based machine translation for 9 language directions. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1063–1073.

Marlies van der Wees, Arianna Bisazza, and Christof Monz. 2016. Measuring the effect of conversational aspects on machine translation quality. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics*, pages 2571–2581.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Keiji Yasuda, Ruiqiang Zhang, Hirofumi Yamamoto, and Eiichiro Sumit. 2008. Method of selecting training data to build a compact and efficient translation model. In *International Joint Conference on Natural Language Processing (IJCNLP)*, pages 655–660.

Dakun Zhang, Jungi Kim, Joseph Crego, and Jean Senellart. 2016. Boosting neural machine translation. *arXiv preprint arXiv:1612.06138*.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. *arXiv preprint arXiv:1604.02201*.