On the Complementarity of Data Selection and Fine Tuning for Domain Adaptation

Dan Iter
Stanford University
daniter@stanford.edu

David Grangier
Google Brain
grangier@google.com

Abstract

Domain adaptation of neural networks commonly relies on three training phases: pretraining, selected data training and then fine tuning. Data selection improves target domain generalization by training further on pretraining data identified by relying on a small sample of target domain data. This work examines the benefit of data selection for language modeling and machine translation. Our experiments assess the complementarity of selection with fine tuning and result in practical recommendations: (i) selected data must be similar to the fine-tuning domain but not so much as to erode the complementary effect of fine-tuning; (ii) there is a trade-off between selecting little data for fast but limited progress or much data for slow but long lasting progress; (iii) data selection can be applied early during pretraining, with performance gains comparable to long pretraining session; (iv) data selection from domain classifiers is often more effective than the popular contrastive data selection method.

1 Introduction

Machine learning models, and neural networks in particular, benefit from large training sets. However, for many application domains, the amount of training data representative of the inference conditions is limited. It is therefore common to train a model over a large amount of generic, out-of-domain data while relying on a small amount of target domain data to adapt such a model. In the recent years, a large body of work has focused on leveraging large amount of web data to train neural networks for language modeling (Peters et al., 2018; Devlin et al., 2019) or translation systems (Bañón et al., 2020; Koehn et al., 2020). Such systems are then adapted to the target distribution, typically via fine tuning (Liu et al., 2019; Raffel et al., 2020). This work studies data selection, an intermediate training phase that visits a subset of the out-of-domain data that is deemed closer to the target domain.

Previous work has proposed conducting a data selection step after pretraining (van der Wees et al., 2017a; Wang et al., 2018; Gururangan et al., 2020; Aharoni and Goldberg, 2020), either as a final training stage or before regular fine tuning. Data selection is meant to identify a subset of the out-of-domain pretraining set which might be the most helpful to improve generalization on the target distribution. This selection is typically conducted by estimating the probability that each data point belongs to the target domain (Moore and Lewis, 2010; Axelrod et al., 2011). Recently, (Aharoni and Goldberg, 2020) introduced the use of domain classifiers for data selection.

This work examines the benefit of data selection for language modeling and machine translation. We compare different selection methods and examine their effect for short and long pretraining sessions. We also examine the benefit of selecting varying amount of training data and the impact of selection on the subsequent benefit of fine-tuning. In addition to this novel analysis, our machine translation experiments compare the benefit of selecting data with a classifier based on source language, target language or both.

The effectiveness of data selection is dependent on (i) the similarity of the pretraining data to the target domain data, (ii) the precision of the selection method to identify in-domain examples from the pretraining set, (iii) the extent to which training on the selected data is complementary to fine-tuning. This work focuses on selecting data from the pretraining set so (i) is fixed. We show that (ii) benefits from the use of domain classifiers, in particular, fine-tuned pretrained language models, outperforming the more popular constrastive methods (eg. Wang et al. (2018)) in all settings that we tested. We present the first analysis of (iii), which we refer to as the complimentarity of selected data

to finetuning data. We show that some data selection methods can actually erode the effectiveness of subsequent fine-tuning. In some settings, we even report that a poor complementarity of selection and fine tuning can result in their combination reaching worse results than fine tuning alone.

Effective application of data selection requires careful selection of when to switch from pretraining to selection, how much selected data to train on and how long to train on selected data before switching to finetuning. Much of the previous work on data selection either evaluates small models that converge quickly (Moore and Lewis, 2010; Axelrod et al., 2011) or does not describe the extent of grid search over selection size, number of steps of pretraining and number of steps of training on selected data. We are the first to analyze the hyperparameter selection tradeoffs for data selection on large neural models, where models may be undertrained (Liu et al., 2019) and evaluating many selection sizes may be prohibitively expensive. We evaluate data selection on checkpoints with variable numbers of pretraining steps and show that data selection provides consistent results between minimally and extensively pretrained models. We also show the challenges of searching over selection sizes because smaller selection sizes always converge more quickly but are outperformed by larger selection sizes trained for more steps.

Our findings are the following: (i) the data selection mechanism must select data that is similar, but complementary to the fine tuning dataset (ii) the amount of selected data introduces a trade-off between quick but limited improvements when limiting selection to the best data, and long lasting but slow progress when selecting more data with an overall worse quality, (iii) data selection techniques are not created equal and domain classifiers often outperform contrastive scoring, the most common data selection method, (iv) we propose three simple variants of domain classifiers for machine translation that can conditions the classifier on either source, target or both. We demonstrate these findings on language modeling and two language pairs for neural machine translation.

2 Related Work

In Natural Language Processing (NLP), adaptation methods have been applied to language modeling (Moore and Lewis, 2010), machine translation (Axelrod et al., 2011; Daumé III and Jagarlamudi, 2011), dependency parsing (Finkel and Manning, 2009) or sentiment analysis (Tan et al., 2009; Glorot et al., 2011). With the growing popularity of neural methods (Collobert et al., 2011; Bahdanau et al., 2015; Goldberg, 2017), the adaptation of neural models via fine tuning has become wide-spread for various NLP applications (Devlin et al., 2019; Liu et al., 2019; Raffel et al., 2020). Data selection is another popular technique (van der Wees et al., 2017b; Wang et al., 2018) which can be used on its own or in combination to fine tuning.

Data selection is a common domain adaptation method. It has been been introduced before neural methods were popular (Moore and Lewis, 2010; Axelrod et al., 2011) and has later been adapted to neural networks (Duh et al., 2013; van der Wees et al., 2017b; Wang et al., 2018). Data selection relies on an intermediate classifier which discriminate between in-domain and out-of-domain data. This classifier is trained relying on the small in-domain dataset and the large out-of-domain dataset and is then applied to the out-of-domain set to identify the examples closest to the targeted domain. Choosing a selection model and the amount of out-of-domain data to select have a strong impact on the effectiveness of the selection methods (Aharoni and Goldberg, 2020; Gururangan et al., 2020). Our experiments explore these aspects, in addition to the complementarity of selection with fine tuning.

Data selection can be performed in multiple rounds, either to gradually restrict the out-of-domain dataset to less and less data (van der Wees et al., 2017b) or to re-evaluate out-of-domain data as pretraining progresses (Wang et al., 2018). Data selection can also be performed as a continuous online process (Wang et al., 2018, 2021; Dou et al., 2020). Our work focus on single round data selection, the most common setting. The benefit of dynamic selection effectiveness has shown to be variable (Wang et al., 2018) and its use involves defining a complex schedule which is a research topic in itself (Kumar et al., 2019).

Data selection for domain adaptation is also related to data selection for multitask learning. In that case, the out-of-domain dataset is composed of heterogeneous data from different tasks/domains and the training algorithm favor data from some tasks at the expense of others (Graves et al., 2017; Wu et al., 2020; Standley et al., 2020). Contrary to our setting, selection operates only at the task level and the association of training examples to tasks

is already known. Multitask learning is an active area of research. This area has explored dynamic selection with reinforcement learning (Graves et al., 2017; Guo et al., 2019) as well as update projections to align out-of-domain gradients to in-domain gradients (Yu et al., 2020; Dery et al., 2021). Some of these ideas have later been investigated in the context of data selection for domain adaptation (Wu et al., 2018; Kumar et al., 2019; Wang et al., 2021).

3 Data Selection Methods

This section presents the selection method our experiments will focus on and introduce the trade-offs involved in choosing data selection hyperparameters.

3.1 In-Domain Data Selection

Domain adaptation has been introduced for application domains where data reflecting the inference conditions is only available in limited quantity. This setting considers that two training sets are available, a large generic out-of-domain dataset and a small specialized in-domain dataset from the targeted domain (Søgaard, 2013). Classical machine learning assumes that training and test data originate from the same distribution. At the same time, statistical modeling reaches better generalization performance with large training sets (Vapnik, 1998). Domain adaptation therefore faces a tension between using a large data set with a distribution possibly far from the test conditions and using a small training set matching the test condition.

Data selection tries to address this dilemma. It examines the out-of-domain data and identifies training examples likely to be most effective at improving the in-domain training loss. For neural methods, data selection is often used in conjunction with fine tuning in a three phases process, as shown in Algorithm 1. In a first phase, the model is pretrained on all the out-of-domain data. In a second phase, an intermediate classifier is trained to distinguish in-domain from out-of-domain data, using both training sets. The classifier is applied to the out-of-domain set to identify examples considered close to in-domain data. The intermediate classifier is then no longer required and the main model is trained on the selected data starting from the pretrained parameters. Finally, the main model is fine tuned, i.e. it is trained on the small in-domain training dataset starting from the parameters after the selection phase.

Algorithm 1: Data Selection & Fine Tuning for Neural Models

Input: D, T out and in domain train sets.
Output: θ trained model parameters.

Function `Select` (D, T, n) **:**
  $w \leftarrow$ trainClassifier($D \cup T$)
  $Y \leftarrow$ classify(w, D)
  return $\text{argtop}_n(Y)$

Function `Main` (D, T) **:**
  $\theta_0 \leftarrow$ initParam()
  $\theta_{\text{pre}} \leftarrow$ train(θ_0, D) `#pretraining`
  $D_{\text{sel}} \leftarrow$ select(D, T, n)
  $\theta_{\text{sel}} \leftarrow$ train($\theta_{\text{pre}}, D_{\text{sel}}$)
  $\theta_{\text{ft}} \leftarrow$ train(θ_{sel}, T) `#fine-tuning`
  return θ_{ft}

Contrastive Data Selection: Commonly, classification is done by estimating the probability that a given out-of-domain example x belongs to the target domain, $P(\mathcal{T}|x)$. Such an estimation can be done by contrasting the likelihood estimated by in-domain LM, $P(\cdot|\mathcal{T})$ and an out-of-domain LM, $P(\cdot|\mathcal{D})$, i.e.

$$\log P(\mathcal{T}|x) = \log P(x|\mathcal{T}|) - \log P(x|\mathcal{D}) + C \quad (1)$$

where C is a constant (log prior ratio). This method was introduced as *intelligent selection* (Moore and Lewis, 2010) and was later renamed *contrastive data selection* (CDS) (Wang et al., 2018). Initially, it relied on independent n-gram LMs for estimating $P(\cdot|\mathcal{T})$ and $P(\cdot|\mathcal{D})$, trained respectively on the (small) in-domain training set T and the (large) out-of-domain training set D (Moore and Lewis, 2010; Axelrod et al., 2011). With neural LMs, $P(\cdot|\mathcal{T})$ can be estimated by fine-tuning $P(\cdot|\mathcal{D})$ as suggested by (van der Wees et al., 2017b; Wang et al., 2018).

The fine tuning strategy is particularly efficient when one performs data selection to adapt a language model. In that case, there is no need for an intermediate model. The pretrained language model to adapt is itself fine-tuned in a few steps on T and is itself used to score the out-of-domain set.

Classifier Selection: Discriminative classification (DC), introduced by Aharoni and Goldberg (2020); Jacovi et al. (2021), trains a binary classifier to distinguish T and D examples. This classifier is either trained from scratch or fine tuned from a pre-

trained model (Devlin et al., 2019; Liu et al., 2019). Aharoni and Goldberg (2020) train the domain classifier, which they refer to as "Domain-Finetune", only on the source (English) side of the parallel corpus. We propose two alternative domain classifiers, that instead condition the classifier on either the target language or both source and target concatenated. To finetune language models on the target language data, we use BERT models that are pretrained on German (deepset.ai), Russian (Kuratov and Arkhipov, 2019) and multilingual BERT (Devlin et al., 2018).

The motivation for these alternative classifiers are two fold: (1) noisy web crawled translation datasets often have incorrect translations (or even languages) which could be missed by the domain classifier if only conditioning on the English source data, (2) the multilingual domain classifier is able to model the interaction between the source and target and is more analogous to the *bilingual cross-entropy difference* proposed by Axelrod et al. (2011)

Compared to CDS, DC trains a different model which adds training overhead. On the other hand, a distinct intermediate model offers more flexibility. The classifier might be pretrained on a different task (e.g. masked LM to select translation data) and its capacity can be selected independently from the hyperparameter of the model to be adapted. Both aspects are important since intermediate models can easily overfit given the small size of the target domain set T.

Nearest Neighbor Selection: A lesser used methods is sentence embedding nearest neighbors (Gururangan et al., 2020; Aharoni and Goldberg, 2020). Embedding nearest neighbors relies on a pretrained model (Devlin et al., 2019; Liu et al., 2019) to represent sentences as vectors and then measure a domain-score by comparing the distance between a candidate sentence vector $v(x)$ and the average in-domain sentence vector $\frac{1}{|T|}\sum_{x \in T} x$.

In our experiments, we evaluate both constrastive data selection, the most common method by far, and selection with discriminitative classifiers as it has been shown more effective in subsequent work (Aharoni and Goldberg, 2020). Previous work and our preliminary experiments indicated that nearest neighbor selection was not competitive with other baselines so we do not include it in our analysis.

3.2 Hyperparameter Trade-offs

Data selection for domain adaptation requires selecting several hyperparameters: the *number of pretraining steps*, i.e. when to transition from training on the full out-of-domain set to the selected subset; the *number of selection steps*, i.e. how long to train the model on the selected data; the *fraction of selected data*, i.e. the size of the selected subset.

These parameters are important as they impact the computational cost of training and the target domain generalization performance. To examine these trade-offs, the difference between pretraining and fine-tuning is important. Pretraining on a large dataset starts with an initial strong generalization improvement, followed by a long session where the rate of generalization improvement is still positive but ever diminishing. Fine tuning gives a strong generalization improvement in a few steps before overfitting quickly. The fraction of selected data allows trading off between these two extremes: a large fraction of selected data results in a large training set with a distribution close to the out-of-domain distribution while a small fraction results in small training set with a distribution close to the in-domain distribution. This means that settings with large fractions can perform more steps with generalization improvement albeit at a slower pace compared to lower fraction settings. Thus the number of selection steps and the selected fraction parameter interact. Our experiments investigate this interaction.

We characterize the effects of overfitting of the intermediate selection classifier, which uniquely affects data selection in conjunction with finetuning. The intermediate classifier is trained on the small target domain set T. As any machine learning model, it is biased toward its training set and the data it selects can reflect this bias. The selected out-of-domain examples might resemble the examples of T more than other in-domain examples unseen during training. This bias transferred to the selected data is itself inherited by the model trained on the selected data. This indirect overfitting is crucial for later fine tuning: we report that, in some cases, the selected data is too similar to T. There, the complementary value of selection and fine tuning vanishes as data selection fails to identify data providing updates complementary to those provided later by fine tuning on T.

4 Experiments

We evaluate domain adaptation with data selection on two tasks, language modeling (LM) and machine translation (MT). For both tasks, we have a large out-of-domain dataset and a small number of examples from the target domain. Both sets of data fulfil two functions each. The out-of-domain data is used to pretrain the model and all the selected data come from the out-of-domain set. The small target domain set is used to train the intermediate model that scores examples for data selection and, critically, this same set is used for finetuning the final model. For evaluation, we also have a validation set and test set from the target domain. The validation set is used to select hyperparameters and early stopping points and the test set is only used for the final model evaluation.

For language modeling, we use the 4.5 million sentences from the One Billion Word corpus (Chelba et al., 2013) as the out-of-domain set and 5k sentences from the Yelp corpus as the target domain. This dataset was used for domain adaptation by (Oren et al., 2019) and we use their filtered and preprocessed version of the data, including the 1k Yelp validation set and 10k Yelp test set. We train 2 language models; a 2-layer LSTM recurrent network (Zaremba et al., 2014) and a base-size transformer (Vaswani et al., 2017).

Our machine translation experiments focus on English-to-German and English-to-Russian. For the out-of-domain set, we use 4.5 million English-to-German pairs and and 5.2 million English-to-Russian pairs taken from filtered Paracrawl (Esplà et al., 2019). Paracrawl is composed of translations crawled from the web. Even though we use the filtered version of the dataset, Paracrawl is still noisy including examples of entirely mismatched sentences and occasionally incorrect languages. As in domain data, we rely on news data from the News Commentary Dataset (Tiedemann, 2012), which are high quality translations from the news domain. Our in-domain set is limited to 6k sentence pairs. We use an additional 3k for validation and 10k as the test set. As a neural MT model, we train a base transformer (Vaswani et al., 2017). Code to reproduce our experiments is available[1]. Models are implemented with Flax (Heek et al., 2020).

We finetune on the small in-domain set by grid searching for a learning rate and using the validation set for early stopping.

[1]https://git.io/JuAAL

4.1 Selection Methods

Contrastive Data Selection The base pretrained (PT) model is fine-tuned (FT) on the small target domain dataset. This model acts as the "intermediate" model in this setting. Each example in the out-of-domain dataset is scored by the difference between the log likelihoods of the fine-tuned model and the pretrained model. The full dataset can be ranked by this score and a threshold is selected to train on a uniform distribution of only the top examples.

Discriminative Classifier The target domain dataset is used as positive examples and random samples from the out-of-domain dataset are used as negative examples to train a discriminative domain classifier. The classifier can be a new model trained from random weights, the base model with a binary classification head or a pretrained model from another task (such as a generic masked language model). Unlike CDS, the base model is not necessarily reused. The input features to the classifier may either be representations learned from the pretrained base model, other embeddings or the raw text data. In the case of machine translation, the classifier can be trained on the source, target or both.

In our transformer experiments, we evaluate CDS and two classifiers, (i) a logistic regression model on bytepair encodings (Sennrich et al., 2016) and (ii) a fine-tuned BERT classifier (deepset.ai; Kuratov and Arkhipov, 2019; Devlin et al., 2018). We use four settings for the BERT classifier, training on the source, target, mean of the former two, and concatenated language pairs, using the respective language specific pretrained BERT. For the concatenated case, we use a multilingual BERT.

	En-De		En-Ru	
	logPPL	BLEU	logPPL	BLEU
PT	1.666	*23.71*	1.815	*23.20*
+FT	1.612	*26.89*	1.708	*24.92*
PT + CDS	1.626	*26.77*	1.757	*24.08*
+FT	1.608	*27.27*	1.707	*25.08*
PT + DC (LogReg)	1.624	*26.22*	1.762	*23.43*
+FT	1.575	*27.54*	1.666	*25.35*
PT + DC (BERT)	1.599	*26.33*	1.752	*23.66*
+FT	**1.550**	***27.78***	**1.645**	***25.52***

Table 1: Data selection for machine translation of English to German and English to Russian. BLEU in italics next to log-perplexity (log PPL). For both datasets, models were trained to 200K steps of pretraining and 15k steps of data selection.

	En-De		En-Ru		LM
	lgPPL	BLEU	lgPPL	BLEU	lgPPL
PT	1.00	1.00	1.00	1.00	1.00
+FT	1.00	1.00	1.00	0.992	1.00
CDS	1.00	1.00	1.00	1.00	1.00
+FT	1.00	0.998	1.00	0.975	1.00
DC-LR	1.00	1.00	1.00	1.00	1.00
+FT	0.951	0.890	0.840	0.742	0.998
DC-BERT	1.00	1.00	1.00	1.00	1.00
+FT	-	-	-	-	-

Table 2: Paired bootstrap comparison: each value reports the fraction of samples with worse mean performance than PT + DC-BERT + FT for 1k samples of 10k sentences sampled from a 10k sample test set.

4.2 Training on Selected Data

Machine Translation Table 1 reports the log-perplexity and BLEU scores on two language pairs for each of the selection methods described above. Data selection always outperforms the baseline without selection, with the BERT domain classifier producing the best log-probability and BLEU on both datasets. The effectiveness of DC compared to CDS is a surprising result given the popularity of CDS. We fix the number of training steps on the selected data to 15K and pretrain the baseline model for an additional 15k steps so there is the same number of pretraining + finetuning steps for all settings. We search the optimal selection size for this cutoff of training steps, which we found to be 1 million for En-Ru and 500k for En-De. We report results before and after finetuning to highlight the variation in effectiveness of finetuning after the alternative selection methods. This is particularly noticeable for En-Ru where CDS outperforms the logistic regression classifier before finetuning but is worse after finetuning. In all settings, finetuning is more effective after data selection with a discriminative classifier rather than with CDS. Section 4.3 provides insight as to why this is the case.

Table 2 reports the paired bootstrap resampling (Koehn, 2004) where the PT + DC (BERT) + FT model is compared to the baseline models, in terms of loss and BLEU, corresponding to Table 1. Each value is computed from the 10,000 example test set. We draw 1,000 bootstrap samples of 10,000 points each, with replacement. This test shows that the classifier method of data selection outperforms CDS with over 99% statistical significance on log-perplexity.

Figure 1 shows the log-probabilities at different checkpoints ranging from 50k to 1 million steps of training. The relative benefit of FT and DC+FT over PT is diminishing as training progresses. However, there are consistent benefits from data selection, so longer pretraining on large models is not sufficient to replace data selection. Even pretraining up to 1m steps and finetuning (log ppl = 1.530) does not reach the loss from DC + FT at 400k (log ppl = 1.519). The relative improvement between methods is surprisingly constant across pretraining steps with a slight decline in the complementary benefit of combining fine tuning with selection. This means that comparing the adaptation methods early in the pretraining process is indicative of their relative loss at a later stage.

Further evaluation of performance at different checkpoints throughout pretraining can be found in the Appendix.

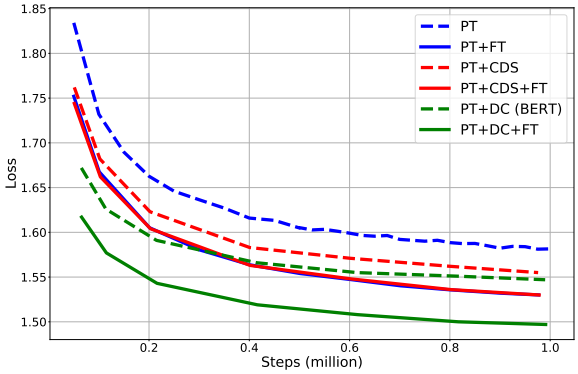

Figure 1: The validation loss curves for pretraining, data selection and finetuning (MT En-De). The pretraining loss (PT) is a single training run, whereas all the other points are checkpoints from the base run that were trained on selected data and/or finetuned.

Domain Classifier Variants Table 3 reports the log-perplexities and BLEU scores for the four variants of the BERT domain classifier. Aharoni and Goldberg (2020) propose the Source DC method. We propose also exploring target-language-conditioned domain classifiers, and in fact, find that the Target DC selection method outperforms Source DC on En-DE. Concatenation DC does not yield the best results despite having access to the most data (ie. both source and target). This may be because of the pretraining mismatch, in that Multilingual BERT was not trained on pairs of segments from different languages. We also take evaluate using the mean score of the source and target models as a simple alternative to the multilingual BERT approach. Future work may explore alterna-

tive methods for fusing source and target language representations for training a domain classifier.

	En-De		En-Ru	
	log PPL	BLEU	log PPL	BLEU
Target DC	**1.550**	***27.78***	1.653	*25.21*
Source DC	1.557	*27.52*	**1.645**	***25.52***
Concat DC	1.560	*27.68*	1.657	*25.20*
Mean DC	1.555	*27.71*	1.647	*25.29*

Table 3: Different types of BERT classifiers, target uses the target language (De/Ru), the source is English and *Concat* concatenates source and target and trains classifier on multilingual BERT. Mean takes the mean scores from source and target classifiers. All models are evaluated at 200k pretraining steps, similar to Table 1.

	LSTM	Transformer
PT	4.978	4.582
+FT	4.284	4.145
PT + CDS	4.548	4.392
+FT	4.183	4.151
PT + DC (LogReg)	4.644	4.456
+FT	4.183	4.108
PT + DC (LM Hidden)	4.603	-
+FT	**4.179**	-
PT + DC (BERT)	-	4.385
+FT	-	**4.069**

Table 4: Language modeling results (log-perplexity) across selection methods for an LSTM and a base-transformer. The LSTM was pretrained for 115k steps and the transformer was trained for 20k steps.

Language Modeling For language modeling we evaluate on both a modestly sized LSTM and a base-size transformer. For the LSTM domain classifier, we reuse the pretrained language model as the feature representation for a simple linear domain classifier (LM Hidden), as a smaller domain classifier seems appropriate given the smaller language model. We see similar results for the two models despite the large differences in number of parameters, training steps and proximity to convergence. The LM results in Table 4 show that fine tuning (PT+FT) and data selection (CDS, DC) are improving the pretrained model on target domain validation data. The benefit of FT alone is generally greater than selection alone but both approaches are complementary with the best result obtained with combined approaches (CDS+FT, DC+FT). When comparing methods we observe that DC is worse than CDS on its own but it is equivalent or better in combination with fine tuning (DC+FT vs CDS+FT). This indicates that the methods differ in their complementarity with FT and evaluating selection approaches before fine tuning is not sufficient.

4.3 Overfitting and Complementarity

Our work compares two common data selection techniques, contrastive data selection (CDS) and a discriminative domain classifier (DC). As discussed in the previous section, we found the combination of DC+FT to be the most effective combination both for our LM and MT settings. One reason of this success is the complementarity of DC with FT. CDS did not benefit as much from subsequent fine tuning as DC selection.

In Figure 2 (left), we show the learning curves for both CDS and DC (BERT) with the same selection size of 1m for MT with 200k steps of pretraining. The red dotted curve show that the CDS model reaches excellent performance on the target-domain training set, but fail to perform as well on the target-domain validation set. This means that the MT model trained on CDS selected data suffers more from overfitting than the MT model trained on DC selected data. This is particularly surprising given the large selection size of nearly 1/4th of pretraining data. The data selected by CDS is too specific to the target-domain training set. This bias also certainly explains the worse complementary of CDS and FT, i.e. if CDS selects a training set whose effect is similar to the target-domain training set T, the updates from T at fine-tuning are less beneficial.

Lastly, we examine important pitfalls to avoid when comparing selection methods and validating their parameters. Figure 2 (middle) shows that when considering selection sets of different sizes, training curves converges at different rates. Small selected subsets progress at the fastest rate but reaches their best generalization quickly, and subsequently overfit, while large subsets progress at a slower rate but their best generalization later. This means that short diagnostics to pick the subset size will under estimate the value of large subsets. This is problematic for efficiently defining curriculum with data selection (Kumar et al., 2019). Similarly, the generalization loss of model which went through a data selection phase but prior to fine tuning is also misleading to predict its loss after fine tuning as illustrated in Figure 2 (right).

4.4 Effectiveness of Data Selection

The purpose of the intermediate data selection model is to rank all the out-of-domain data from

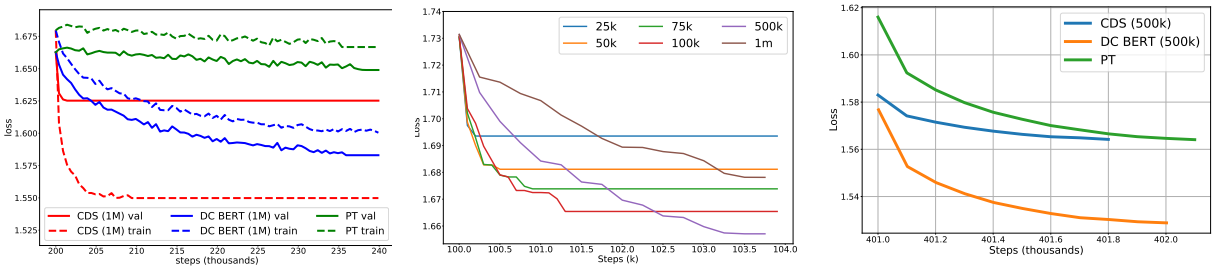

Figure 2: Effects of overfitting and complementarity: **Left:** Validation and training loss on the target domain during training on selected data (MT En-De). The dotted line falling below the solid line indicates the model is overfitting to the small target domain dataset despite never seeing this data in training. **Middle:** Loss curves for 6 different data selection sizes for DC (BERT) at the 100k checkpoint (MT En-De). Larger sizes improve loss more slowly but can be trained for longer to eventually outperform the smaller sets. For readability, we display the best checkpoint up to each step. **Right:** Validation loss on MT En-De during finetuning. Both data selection methods start at a loss that is better than pretraining but CDS does not benefit much from finetuning, reaching a loss similar to finetuning without data selection. Classifier selection has large a improvement from finetuning.

most to least similar with respect to the in-domain data. We evaluate and report the performance of CDS and DC for both LM and MT tasks. The data selection model is never used explicitly as a binary classifier but rather as a scorer. However, as a proxy for the quality of scoring, we evaluate the binary classification accuracy on an unseen set of in-domain and out-of-domain data. We also report the average quantile of the in-domain validation data which simulates where in the ranking true in-domain examples would appear. We split the out-of-domain data into 100 equal bins and take the average of the bin index that each in-domain example would fall into by its data selection score.

Table 5 shows good performance of CDS and DC for language modeling but clear underperformance of CDS as a binary classifier in the MT setting. Also, it is noteworthy that logistic regression on byte-pair unigrams outperforms CDS and approaches the performance of BERT while having many fewer parameters and a much lower training cost.

	Classifier	Accuracy	Avg Quant.
LM	CDS	91.65%	3.6
	MLP	89.02%	4.9
MT (En-De)	CDS	66.94%	26.0
	LogReg	87.52%	3.9
	BERT	93.51%	2.0

Table 5: Binary classification accuracy of domain classifier and average quantile of in-domain data when binned with ranked out-of-domain data.

5 Conclusions

This work explores data selection, a popular method for domain adaption for neural language modeling and neural machine translation. Data selection typically divides a training run into three phases: pretraining on out-of-domain data, training on out-of-domain data selected to resemble target domain data and fine tuning on target domain data. We compare the most common selection methods, contrastive data selection and discriminative model classifier and measure their complementarity with fine tuning.

Our experiments motivate several practical recommendations for the practitioner: (i) pretraining followed by data selection and fine tuning can reach a given generalization loss several time faster in terms of total training steps than pretraining with fine tuning; (ii) a data selection method should not be evaluated before fine tuning since not all methods/parameters bring the same complementary value compared to fine tuning; (iii) data selection should care about overfitting to the in-domain training set, since this type of overfitting results in selected data very similar to the fine tuning set and impacts the complementarity of data selection and fine tuning; (iv) longer pretraining runs are always beneficial to later adaptation stages for fine-tuning, data selection and their combination but pretraining has diminishing return; (v) despite the popularity of contrastive data selection, discriminative domain classifiers consistently outperformed this method in our experiments.

References

Roee Aharoni and Yoav Goldberg. 2020. Unsupervised domain clusters in pretrained language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7747–7763, Online. Association for Computational Linguistics.

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, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.

Marta Bañón, Pinzhen Chen, Barry Haddow, Kenneth Heafield, Hieu Hoang, Miquel Esplà-Gomis, Mikel L. Forcada, Amir Kamran, Faheem Kirefu, Philipp Koehn, Sergio Ortiz Rojas, Leopoldo Pla Sempere, Gema Ramírez-Sánchez, Elsa Sarrías, Marek Strelec, Brian Thompson, William Waites, Dion Wiggins, and Jaume Zaragoza. 2020. ParaCrawl: Web-scale acquisition of parallel corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4555–4567, Online. Association for Computational Linguistics.

Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. 2013. One billion word benchmark for measuring progress in statistical language modeling. Technical report, Google.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(76):2493–2537.

Hal Daumé III and Jagadeesh Jagarlamudi. 2011. Domain adaptation for machine translation by mining unseen words. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 407–412, Portland, Oregon, USA. Association for Computational Linguistics.

deepset.ai. Open sourcing german bert. Https://deepset.ai/german-bert.

Lucio Dery, Yann Dauphin, and David Grangier. 2021. Auxiliary task update decomposition: The good, the bad and the neutral. In *International Conference on Learning Representation (ICLR)*.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Zi-Yi Dou, Antonios Anastasopoulos, and Graham Neubig. 2020. Dynamic data selection and weighting for iterative back-translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5894–5904, Online. Association for Computational Linguistics.

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 (Volume 2: Short Papers)*, pages 678–683, Sofia, Bulgaria. Association for Computational Linguistics.

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

Jenny Rose Finkel and Christopher D. Manning. 2009. Hierarchical Bayesian domain adaptation. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 602–610, Boulder, Colorado. Association for Computational Linguistics.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *ICML*.

Yoav Goldberg. 2017. Neural network methods for natural language processing. *Synthesis lectures on human language technologies*, 10(1):1–309.

Alex Graves, Marc G. Bellemare, Jacob Menick, Rémi Munos, and Koray Kavukcuoglu. 2017. Automated curriculum learning for neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1311–1320. PMLR.

Han Guo, Ramakanth Pasunuru, and Mohit Bansal. 2019. AutoSeM: Automatic task selection and mixing in multi-task learning. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3520–3531, Minneapolis, Minnesota. Association for Computational Linguistics.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Jonathan Heek, Anselm Levskaya, Avital Oliver, Marvin Ritter, Bertrand Rondepierre, Andreas Steiner, and Marc van Zee. 2020. Flax: A neural network library and ecosystem for JAX.

Alon Jacovi, Gang Niu, Yoav Goldberg, and Masashi Sugiyama. 2021. Scalable evaluation and improvement of document set expansion via neural positive-unlabeled learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 581–592, Online. Association for Computational Linguistics.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain. Association for Computational Linguistics.

Philipp Koehn, Vishrav Chaudhary, Ahmed El-Kishky, Naman Goyal, Peng-Jen Chen, and Francisco Guzmán. 2020. Findings of the WMT 2020 shared task on parallel corpus filtering and alignment. In *Proceedings of the Fifth Conference on Machine Translation*, pages 726–742, Online. Association for Computational Linguistics.

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 1 (Long and Short Papers)*, pages 2054–2061, Minneapolis, Minnesota. Association for Computational Linguistics.

Yuri Kuratov and Mikhail Arkhipov. 2019. Adaptation of deep bidirectional multilingual transformers for russian language. *arXiv preprint arXiv:1905.07213*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Robert C. Moore and William Lewis. 2010. Intelligent selection of language model training data. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 220–224, Uppsala, Sweden. Association for Computational Linguistics.

Yonatan Oren, Shiori Sagawa, Tatsunori Hashimoto, and Percy Liang. 2019. Distributionally robust language modeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4227–4237, Hong Kong, China. Association for Computational Linguistics.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

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 (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Anders Søgaard. 2013. Semi-supervised learning and domain adaptation in natural language processing. *Synthesis Lectures on Human Language Technologies*, 6(2):1–103.

Trevor Standley, Amir Roshan Zamir, Dawn Chen, Leonidas J. Guibas, Jitendra Malik, and Silvio Savarese. 2020. Which tasks should be learned together in multi-task learning? In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 9120–9132. PMLR.

Songbo Tan, Xueqi Cheng, Yuefen Wang, and Hongbo Xu. 2009. Adapting naive bayes to domain adaptation for sentiment analysis. In *European Conference on Information Retrieval*, pages 337–349. Springer.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).

Marlies van der Wees, Arianna Bisazza, and Christof Monz. 2017a. Dynamic data selection for neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1400–1410.

Marlies van der Wees, Arianna Bisazza, and Christof Monz. 2017b. Dynamic data selection for neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1400–1410, Copenhagen, Denmark. Association for Computational Linguistics.

V.N. Vapnik. 1998. *Statistical Learning Theory*. A Wiley-Interscience publication. Wiley.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, undefinedukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

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*, pages 133–143, Brussels, Belgium. Association for Computational Linguistics.

Xinyi Wang, Ankur Bapna, Melvin Johnson, and Orhan Firat. 2021. Gradient-guided loss masking for neural machine translation.

Jiawei Wu, Lei Li, and William Yang Wang. 2018. Reinforced co-training. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1252–1262, New Orleans, Louisiana. Association for Computational Linguistics.

Sen Wu, Hongyang R. Zhang, and Christopher Ré. 2020. Understanding and improving information transfer in multi-task learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. 2020. Gradient surgery for multi-task learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 5824–5836. Curran Associates, Inc.

Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. 2014. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*.

A Appendix

A.1 Training Steps

Figure 3 shows the acceleration of training as a function of pretraining + finetuning (PT+FT) steps needed to reach an equivalent loss for translation.

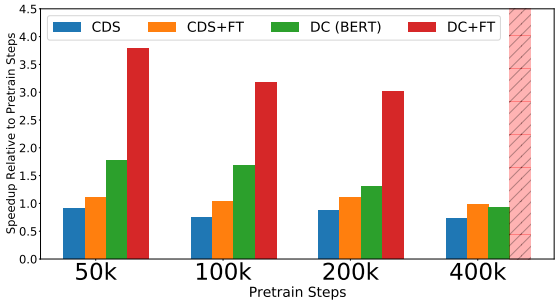

Figure 3: Data selection (MT En-De) as an acceleration method. This table shows the speedup of reaching a given loss at each checkpoint relative to how many steps of pretraining and finetuning are required to reach the same loss. Values lower than 1 indicate that the loss can be reached in fewer steps without data selection. The final bar for DC is shaded to indicate extrapolation and is off the y-axis because the loss is lower than any loss reachable in 1 million steps with pretraining and finetuning.

This figure highlights the effectiveness of pretraining since the performance obtained by data selection for early checkpoints can be matched by simply pretraining longer. Furthermore, DC+FT at 400k pretraining steps cannot be matched, even when pretraining for up to 1m steps. This figure shows that a practitioner with a given generalization requirement can consider data selection early since the target domain generalization gain for early checkpoints might avoid a long pretraining run.

At 50k steps, data selection accelerates training by a factor of about 3.5x, meaning the same performance can be reached with an additional 150k steps of pretraining. However, for later checkpoints, the marginal benefits of pretraining decreases while the improvements from data selection are steady making data selection a clear choice for later checkpoints. In particular for well trained smaller models, such as the LSTM we evaluate for language modeling, the performance after data selection may actually be unreachable just through pretraining either due to the noisiness of the training data that might be filtered from data selection or due to the limited model capacity.

A.2 Complementary Finetuning vs Overfitting

Figure 4 measures the correlation between the relative difference between the train and valid best in-domain loss prior to fine tuning (selection overfitting rate) and the relative difference between the valid loss before and after fine tuning (fine tuning

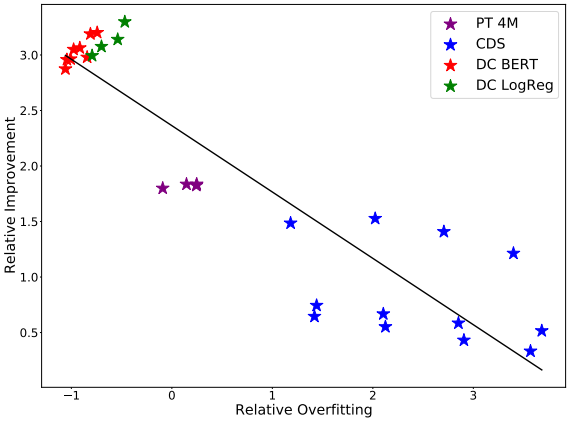

Figure 4: Impact of selection overfitting (MT En-De). When data selection overfits to the in domain set, the improvements from finetuning are lower. The x-axis is the overfitting relative difference and the y-axis is the relative improvement from finetuning. Pearson Correlation Coefficient : -0.91

rate). There is a strong anti-correlation between these factors, showing that overfitting at the selection stage indeed impacts negatively the impact of FT. We include points on this graph selecting the top 4m examples, effectively filtering out the bottom 500k, which has a slight overfitting effect, to include more points with an intermediate overfitting-to-improvement tradeoff.