Bandits Don't Follow Rules: Balancing Multi-Facet Machine Translation with Multi-Armed Bandits

Julia Kreutzer and **David Vilar** and **Artem Sokolov**
Google Research
{jkreutzer, vilar, artemsok}@google.com

Abstract

Training data for machine translation (MT) is often sourced from a multitude of large corpora that are multi-faceted in nature, e.g. containing contents from multiple domains or different levels of quality or complexity. Naturally, these facets do not occur with equal frequency, nor are they equally important for the test scenario at hand. In this work, we propose to optimize this balance jointly with MT model parameters to relieve system developers from manual schedule design. A multi-armed bandit is trained to dynamically choose between facets in a way that is most beneficial for the MT system. We evaluate it on three different multi-facet applications: balancing translationese and natural training data, or data from multiple domains or multiple language pairs. We find that bandit learning leads to competitive MT systems across tasks, and our analysis provides insights into its learned strategies and the underlying data sets.

1 Introduction

Parallel training data for machine translation (MT) is commonly sourced and combined from multiple large sub-corpora to obtain the maximum number of training examples. The WMT shared tasks, for example, provide a number of distinct training corpora since (Koehn and Monz, 2006). Such corpora are *multi-faceted* in nature, consisting of a generally unbalanced *mixture of data sources that differ from each other* in word distribution, domain or other traits. Examples of such differences could range from strongly heterogeneous data like distinct languages for training multi-lingual systems (Dong et al., 2015; Firat et al., 2016; Arivazhagan et al., 2019) to rather subtle variations in data provenance (e.g. human-generated vs. machine-produced data crawled from web), through a mid-strength variation in multi-domain MT (Farajian et al., 2017; Müller et al., 2020; Pham et al., 2021). The nature of data facets and their identity is known

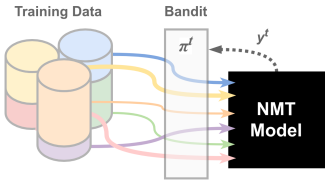

Figure 1: Multi-armed bandits for NMT data selection.

at training time, either from the data sources directly, for example meta-data from data collection pipelines, or can be provided by dedicated classifiers—but this important information is discarded when mixing and shuffling them for training (Arjovsky et al., 2019; Teney et al., 2020). Thus, the optimal balance of facets needs to be decided beforehand, and with the requirements at test time in mind. At test time, facets may be equally important, but they might not all have the same amounts of training data. These data balancing decisions are time-consuming and expensive as they often require multiple iterations for striking the right balance between, on the one hand, robust performance at test time on underrepresented facets and, on the other hand, preserving valuable linguistic and lexical information contained in the higher-represented ones. For additional complication, potential positive and negative transfer between facets should be taken into account (Arivazhagan et al., 2019; Wang et al., 2021). The complexity of these decisions exacerbate as training data grows.

Even with established data balancing heuristics in place (e.g. upsampling with a tuned temperature τ[1] (Devlin, 2019; Arivazhagan et al., 2019)), different balances might be needed at different stages of training. This realization kick-started a development of training curricula (Bengio et al., 2009) which, despite efforts in neural MT, have yet to produce a recipe applicable to concrete data at hand (Zhang et al., 2018). Existing curricula

[1] Sampling from an annealed and renormalized empirical distribution over facets f, $p(f) = \text{softmax}_f(\ln(\hat{p}(f))/\tau)$.

Findings of the Association for Computational Linguistics: EMNLP 2021, pages 3190–3204
November 7–11, 2021.

presuppose fixed notions of difficulty and come with hand-crafted schedules, often inspired by the human learning process (Kocmi and Bojar, 2017; Zhang et al., 2018; Platanios et al., 2019). Such approaches are brittle in that they may not generalize well across tasks, and there has been evidence that even the reverse of the initially hypothesized order works well (Bengio et al., 2009; Wang et al., 2018; Zhang et al., 2018). This suggests that our human intuitions about difficulty and data succession may not correspond to the optimization process of an NMT system (Li and Gong, 2021).

In this paper we argue for *automatically learned and adaptive data curricula*, where the learning system explicitly chooses a facet at each point in training, and does not depend on presupposed schedules. This has three major advantages: First, it relieves system developers from lots of manual work. Second, it can improve quality by ignoring irrelevant, redundant or already learned data. Third, it can directly optimize for a uniform performance objective to maintain quality on all facets. As a side effect, post-training analyses may improve data interpretability and efficiency (Gascó et al., 2012). However, outsourcing data decisions to an auxiliary ML model sacrifices some of control and understanding. In particular, multiple training data selection strategies can lead to models of comparable quality, especially when measured by crude metrics like BLEU.

We formulate *multi-faceted training as a multi-armed bandit learning problem*, where the arms/actions correspond to the available facets in the training data. At each training step, the bandit chooses one facet for the MT system to train on and receives a reward signal whether this choice was beneficial for the training progress (see Figure 1). We implement the EXP3 algorithm (Auer et al., 2002) as proposed for automated curriculum learning (Graves et al., 2017) (§2), and evaluate it on three different multi-facet applications for machine translation. These require balancing training data that is natural or translationese (§4.1), comes from a variety of domains (§4.2), or from many different languages (§4.3). To the best of our knowledge this is the first study that addresses these problems jointly and provides a competitive solution to all of them. We analyze the effects of different reward signals and chosen facets over time, shedding new light on the importance of different facets for each of the tasks.

2 Learning to Select Data with Bandits

Learning a data curriculum can be framed as a multi-arm bandit problem, where the decision to train on a particular subset of data is outsourced to a bandit algorithm that is learned alongside the main task (Graves et al., 2017). After the bandit chooses a facet, the NMT system is updated on a uniformly sampled batch of data from this facet. The system then provides a reward to the bandit, telling how successful this selected batch of data was in terms of overall training progress (see Figure 1).

Formally, the bandit selects actions from a set $\mathcal{A}$ which is a discrete set of ids. In each round t, the bandit selects an action $a^t \in \mathcal{A}$ and observes a scalar loss, $y^t = Y^t_{a^t}$, where Y^t is the complete but unobserved loss vector for each possible action. The bandit parameters are updated to minimize the regret $R = \mathbb{E}[\sum_t y^t] - \min_a \sum_t Y^t_a$ of not playing the arm that is best in hindsight. We operate in a fully adversarial setup assuming that reward vectors Y^t can be arbitrary, i.e., they can depend on the full history, data etc., but cannot be adaptive to the selected action a^t.

With a collection of subsets of training data (facets), covering the full training data, $\cup \mathcal{D}_a = D$, the EXP3 algorithm proceeds as follows (Auer et al., 2002; Graves et al., 2017):

Algorithm 1: Multi-Facet EXP3 for NMT

Input : NMT model θ^0, number of facets n, exploration rate γ, bandit learning rate μ, training facets $\mathcal{D}_a$

Result: Sequence of arms $\{a^1, a^2, \ldots, a^T\}$

Initialize weights $\mathbf{w} = \mathbf{0} \in \mathbb{R}^n$

for $t = 0, \ldots, T$ **do**

- $\pi^t(a) := (1-\gamma)\frac{\exp(\mathbf{w}_a)}{\sum_a \exp(\mathbf{w}_a)} + \frac{\gamma}{n}$
- sample $a^t \sim \pi^t$
- sample a batch B^t uniformly from $\mathcal{D}_{a^t}$
- NMT update step on B^t to get θ^{t+1}
- measure learning progress y^t
- update $\mathbf{w}_a = \mathbf{w}_a + \mu y^t [\![a = a^t]\!] / \pi^t(a)$

The regret R behaves as $O(\sqrt{T \ln d})$ (Auer et al., 2002), so in the limit the bandit will do as good as the best arm from $\mathcal{A}$, i.e., $R/T \to 0$ as $T \to \infty$. Graves et al. (2017) used a slightly modified algorithm EXP3.S (Auer et al., 2002) that competes against any *sequence* of actions to reflect dynamic changes. In practice, we found the performance of the vanilla EXP3 sufficient for NMT.

Measuring learning progress Graves et al. (2017) propose a variety of reward functions for measuring learning progress. In this work, we focus on rewards that are functions of the loss value:

- `loss`: the plain loss objective value, $\mathcal{L}(\theta^t)$;
- `pg`: absolute prediction gain, $\mathcal{L}(\theta^t) - \mathcal{L}(\theta^{t+1})$;
- `pgnorm`: relative `pg`, $1 - \mathcal{L}(\theta^{t+1})/\mathcal{L}(\theta^t)$.

They can be evaluated on the training batch B^t or on a development batch B^t_{dev} (denoted with prefix `dev-`). Due to the ambiguity of what can lead to high losses on the training set—such as noisy, unseen or untypical examples—rewards calculated on batches sampled from dev sets (that in our work contain an equal mix of facets) proved to be more successful in our experiments. This also allowed us to inject the equal importance of facets into the rewards. We calculate rewards on randomly sampled batches, since it is cheaper than on the full dev set (Kumar et al., 2019). This is not an issue for EXP3, as it allows rewards to be non-deterministic: it provably converges for adversarial feedback, and so for identically distributed random rewards too. We linearly re-scale the rewards to $[0; 1]$ clipping them between the 20th and the 80th quantiles of the most recent 5k rewards (Graves et al., 2017).

Training costs The memory overhead of training the bandit alongside the main NMT is negligible since it only requires the storage of reward and sampling statistics across arms (see Algorithm 1). There is no overhead in terms of speed for the `loss` reward since it is already computed during the normal MT training, but a second forward pass is needed to calculate prediction gains (`pg` or `pgnorm`). With the cost of a forward pass c, the additional computational cost per iteration is $O(c)$ for one evaluation batch, independent of the number of facets. This is notable cheaper than recent methods based on gradient similarity that require backward passes on training and development sets for each facet, plus a gradient update for a parametrized policy (Wang et al., 2020a).

3 Experiments

Data To ensure that our recipe generalizes to multiple setups we empirically tested our approach on three different tasks varying across several dimensions (Table 1):

1. **Natural vs. translationese**: For large-scale `en-de` translations we model two facets with a subtle distinction, namely the distinction of "translationese" and "natural" target sides (§4.1). The difficulty lies in the weak demarcation between classes of signals, large provenance diversity of data and the overall large data size.
2. **Multi-domain**: We train a multi-domain NMT system for `en-de` with mid-size training data (§4.2). The automated curriculum has to balance facets of the same language, but with subtle domain-specific differences.
3. **Multilingual**: We experiment with multilingual NMT models on two small-scale subsets of 8 language pairs from the TED57 dataset and the large-scale OPUS100 set with 99 language pairs (§4.3). Facets are defined as language pairs and they are related to varying degrees, so reward signals are expected to vary in terms of dynamics and strength.

Corpus	Lang. pairs	Facets	Entropy	Sent.
Nat.-Transl.	1	2	96.9%	57M
Multi-domain	1	5	85.4%	1.5M
TED57 diverse	8	8	78.9%	766k
TED57 related	8	8	73.1%	586k
OPUS100 M2O/O2M	99	99	91.6%	55M
OPUS100 M2M	198	198	92.7%	109M

Table 1: Overview of multi-faceted training data sets. The entropy of the frequency distribution of facets as present in the corpus is measured in percents of the maximum natural entropy.

Implementation We implemented the Transformer model (Vaswani et al., 2017) in JAX (Bradbury et al., 2018), using the neural network library Flax (Heek et al., 2020) (more details in §A.1). After training we select the model for testing that obtained the highest SacreBLEU score (Post, 2018) on development sets containing a balanced selection of all facets.

4 Results

For each task we evaluate whether the bandit-directed training schedules can outperform the zero-effort "take-it-all" approach where datasets are concatenated and training examples are presented in random order. In addition, we compare it to task-specific best practices, and investigate which strategies are learned by the bandit schedules.

4.1 Natural vs. translationese NMT

Setup We train a `big` Transformer on the concatenation of the News Commentary (v15),

		Avg	Translationese	Natural		Translationese	Natural	
			WMT20	**WMT20-paraph**	**WMT20-rev**	**WMT18**	**WMT18-paraph**	**WMT18-rev**
	Baseline	26.42	27.64	9.23	22.87	52.19	12.61	34.00
	Tagged	27.12	28.05 (29.37)	9.96	23.92	52.24 (50.85)	13.12	35.44
Bandit	`loss`	27.37	27.81	9.37	24.36	50.34	12.73	39.61
Bandit	`pg`	27.66	28.32	9.48	24.66	51.52	12.96	39.03
Bandit	`pgnorm`	26.40	27.49	9.30	22.42	51.37	12.33	35.48
Bandit	`dev-loss`	27.47	28.19	9.49	23.99	51.57	12.64	38.94
Bandit	`dev-pg`	27.39	27.53	9.28	24.56	51.18	12.67	39.12
Bandit	`dev-pgnorm`	27.22	27.68	9.40	24.35	50.21	12.67	38.98
	Baseline+CDS	27.74	29.52	9.76	23.85	53.62	12.99	36.71
	Tagged+CDS	27.50	29.17 (28.98)	10.00	23.58	53.60 (50.18)	13.17	35.46
Bandit	`dev-pgnorm`+CDS	28.01	29.09	9.72	24.20	53.60	13.07	38.44

Table 2: WMT `en-de`: BLEU scores on the natural vs. translationese task. Source tags for tagged baselines correspond to the test set's facet; for translationese sets, BLEU for the natural tag is in brackets.

ParaCrawl (v5.1), Europarl (v10) and CommonCrawl training corpora.[2] Since natural vs. translationese facets are not explicitly marked in the corpora, we train two neural LMs for the target language, one on natural text and one on translated text, and select the higher-scoring one as label (§A). We are interested in improving the naturalness of the translation output, but this is hardly measured by automatic metrics, because standard reference translations are translationese, so BLEU might even give contradictory signals (Freitag et al., 2020b). Therefore, we also evaluate on the *reverse* direction WMT20 set, i.e. the reversed test set for `de-en` (suffix '-rev'), which consist of original German text. This serves as a proxy for measuring the naturalness of the system output. We additionally use the references provided by Freitag et al. (2020b) which were paraphrased versions of the official ones, with the goal of improving their naturalness (denoted with the suffix '-paraph' in the results). The bandit development set contains 2000 sentences of equal mixture of natural (the 'rev' part) and translationese sentences from the WMT19 newstest, and the final evaluation is on faceted WMT18 and WMT20 news test sets.

As a simple controlled translation, we also train tag-based baselines (Riley et al., 2020), where source tags correspond to facets, also during testing. As the natural mode is what often desired, we additionally evaluate translations with the 'natural' tag for translationese sets.

Results All bandit approaches improve over the baseline (Table 2) by around 0.5–0.9 BLEU on average across test sets (except for `pgnorm`), but the individual tendencies vary across reward choices. The dynamics of the bandit arm probabilities (§D, Figure 5) reveal that most rewards prefer the natural part of data since it is harder to learn for the NMT system and results in consistently higher loss values; except for `pgnorm`, which also loses on the reverse set. Additional data filtering by Contrastive Data Selection (CDS) (Wang et al., 2018) leads to major improvements for the baseline on translationese and natural test sets. This approach filters the training data by removing 30% of sentences that are considered noisy by a model iteratively trained on trusted data (here NewsCommentary v15). It was trained independently of the natural and translationese distinction, so the CDS improvements are due to a generally improved quality of the training data. It strengthens bandit results in a similar way, gaining about 0.3 and 1.7 BLEU on two natural tests set while performing comparably on the others, which shows that both approaches are complimentary—we speculate that CDS removing noisy examples allows bandits to better focus on truly difficult examples. Comparing the `dev-pgnorm` bandit without CDS and the baseline with CDS on natural 'rev' test sets suggests that the bandit could compensate the lack of data filtering.

4.2 Multi-Domain NMT

Setup We follow the multi-domain setup by Müller et al. (2020) using the data re-split by Aharoni and Goldberg (2020). By construction it contains in-domain data from five domains and no auxiliary general-domain data, thus preventing data augmentation with pseudo in-domain data selection (Axelrod et al., 2011). The goal of this evaluation is to improve uniformly on all domains using a mixed training set. As in prior work, we use

[2] WMT2020 news translation task.

		Avg	**Med**	**IT**	**Law**	**Koran**	**Subs**
	Size (k)	291.2	248.0	467.3	222.9	18.0	500.0
Base	Aharoni and Goldberg (2020)	40.2	53.3	42.1	57.2	20.9	27.6
	Ours	39.42	51.65	41.47	55.08	21.24	27.67
Static	Uniform ($\tau = \infty$)	39.26	51.91	44.24	50.93	22.34	26.90
	Upsampled ($\tau = 5$)	38.78	51.11	41.10	52.87	23.07	25.72
	Proportional ($\tau = 1$)	40.40	52.83	44.65	54.53	21.23	28.76
	Inverse Proportional ($\tau = -1$)	40.03	53.43	44.97	51.84	22.70	27.20
Bandit	`pg`	40.26	53.38	46.42	51.44	22.19	27.85
	`pgnorm`	38.78	51.76	45.97	50.03	21.43	24.72
	`loss`	39.12	50.63	42.89	51.19	23.08	27.83
	`dev-loss`	39.96	50.22	42.48	55.68	23.34	28.06
	`dev-pg`	40.56	53.35	42.66	55.95	22.62	28.23
	`dev-pgnorm`	40.56	53.23	42.99	55.89	22.79	27.92

Table 3: Test BLEU scores on the multi-domain task. Rewards in bold improve over the baseline uniformly.

the `base` Transformer architecture, and, where possible, try to match the training setup from (Aharoni and Goldberg, 2020) (§A). However, we were not able to exactly replicate their scores due to inevitable implementation differences.

Results The two most successful bandit data selection strategies (`dev-pg` and `dev-pgnorm`) converge faster than the baseline (Figure 2) *and* achieve better scores (up to +1.7 point above the baseline on some domains and 1.1 points on average). Analysing the evolution of facet sampling probabilities (§D, Figure 6), we find that `dev-pg` and `dev-pgnorm` focused largely on Law and Subtitles domains. We hypothesize that these rewards are capitalizing on the higher sentence quantity and hence potential diversity of the higher-resource domains. At the same time, they quickly neglect the IT and Koran domains, which may be structurally simple and/or monotonic. Not frequently training on examples from latter domains does not lead to a decrease of translation quality on them. In general, gains in quality over the baseline are not related to the sampling preferences of the bandits. This highlights the difficulty of designing a proper schedule manually and prior to training using intuition only. Static temperature-based sampling yields gains tied to the availability of resources, (e.g. improvements for $\tau = 1$ on the high-resource domains, and $\tau = -1$ on the low-resource domains, except for $\tau = 5$ which gains only for Koran), but they—in contrast to the dynamic bandits—fail to improve on all domains. This shows that the additional flexibility of the bandits to adapt the sampling distribution during training is beneficial for equitable quality gains.

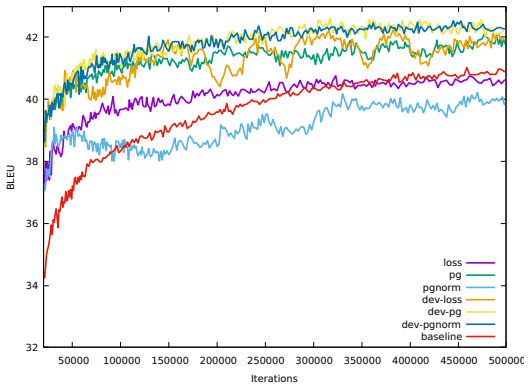

Figure 2: Evaluation scores on the mixed development set during training for the multi-domain task.

4.3 Multilingual NMT

Setup In multilingual MT, parallel training data is often paired with English, so there are three major training setups for multilingual translation: many-to-one (M2O), learning to translate many languages into English; one-to-many (O2M), translating from English, and many-to-many (M2M). We experiment with M2O translations for the `diverse` and `related` subsets of the multilingual TED dataset (Qi et al., 2018). The two subsets cover 8 languages with very different data sizes, selected as pairs of related languages of different size (Neubig and Hu, 2018) or a set of diverse languages from different language families and with different scripts (Wang et al., 2020b). There are large discrepancies in the sizes of the subsets for each language, e.g. `be` has only 4.5k sentences, while the related `ru` has 208.4k. This makes it

valuable for testing the behavior of the bandit with facets that are linguistically similar but very differently scaled. For a more data-balanced setup, we experiment with the OPUS100 dataset (Zhang et al., 2020), which contains up to 1M of training examples sampled from the entirety of the OPUS collection of parallel corpora from various domains (Tiedemann and Nygaard, 2004) for 99 languages paired with English, of which 94 come with test sets. As a result, the data has large inter- and intra-facet diversity. For both evaluation scenarios we train SentencePiece models (Kudo and Richardson, 2018) on a re-balanced corpus (Nguyen and Chiang, 2017; Fan et al., 2020)[3] to create a vocabulary of 32k tokens, add target language tags and train Transformer `base` models. We construct a balanced development set by randomly selecting a fixed number of sentences from the language-specific development sets (500 for TED; 100 for OPUS) to reflect our interest in high quality across all languages. Rewards for the bandit are computed on samples from this balanced dataset. We compare with static uniform sampling distributions ($\tau = \infty$) over facets, and size-proportional ($\tau = 1$) or upsampled ($\tau = 5$) distributions, since they have been reported successful in previous works (Wang et al., 2020b; Zhang et al., 2020). They sample batches of a single language at each step, while the vanilla baseline samples mixed-language batches from the shuffled concatenated data.[4] All other hyperparameters can be found in §A. We report experiments with the `dev-pgnorm` reward since it performed best.

TED Results Tables 4 and 5 compare our results on the `diverse` and `related` subset with the most recent work of Wang et al. (2020b), who proposed a dynamic data scheduling algorithm (Multi-DDS) based on gradient similarity between training and development data. On average, our implementations of batch-wise uniform or proportional sampling yield similar results to theirs for the `diverse` set, but on the `related` set they perform slightly worse, because Wang et al. (2020b) train on more data for `sl` (61.5k) and `pt` (185k) than is contained in the publicly available dataset, resulting in a difference of 8 and 5 BLEU on respective languages.[5] The bandit consistently outperforms the mixed-batch baseline ('Base') and performs similarly to proportional sampling of language-specific batches ('Proportional'). Trivially sampling languages according to their size is a good heuristic for both setups despite the large data discrepancies between languages, corroborating previous findings on this dataset (Neubig and Hu, 2018; Li and Gong, 2021). It yields better results than a uniform sampling scheme (and than the commonly used $\tau = 5$) which a practitioner might have chosen without prior knowledge about the task. The bandit automatically discovers this insight without having access to explicit size information, as can be seen in Figure 3 for the diverse set (related: §C, Figure 4). Compared to size-proportional sampling, it slightly upsamples all smaller languages and slightly downsamples some of the larger ones (`el`, `bg`, `fr`), but not as strongly and consistently as the $\tau = 5$ upsampling. This led to an improvement of the translation quality of the lowest-resource languages, and was incentivized by the equal presence of languages in the balanced development set used for reward calculations. With a similar incentive but much more expensive updates, Multi-DDS's gains over proportional sampling are also on the smallest datasets.

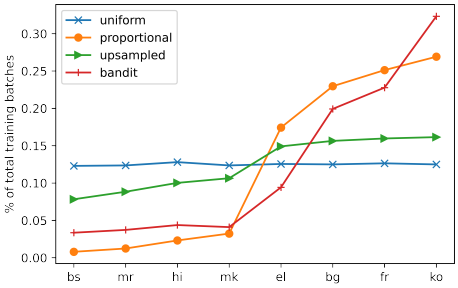

Figure 3: Total number of batches trained on for each language throughout training for TED-diverse M2O.

OPUS100 Results We compare against the M2O and O2M benchmark results set by Zhang et al. (2021), averaging results for less than 0.1M sentence pairs ('Low'), more than 1M ('High') and medium-sized ones ('Med'). Zhang et al. (2021) also use a Transformer `base`, but report averaged results for the last 5 checkpoints and create unbalanced vocabularies of twice the size of ours, resulting in a higher-capacity model. Our baselines therefore score slightly below. The bandit clearly

[3]Upsampling all languages to the maximum size.

[4]The literature has been divided whether to mix batches (Aharoni et al., 2019; Zhang et al., 2020, 2021; Li and Gong, 2021) or not (Firat et al., 2016; Wang et al., 2020b).

[5]Downloaded from https://github.com/neulab/word-embeddings-for-nmt.

	Size (k)	Avg 95.8	bs 5.7	mr 9.8	hi 18.8	mk 25.3	el 134.3	bg 174.4	fr 192.3	ko 205.6
Ours	Base	25.43	21.59	10.00	20.73	29.90	33.74	34.73	35.70	17.06
Wang et al. (2020b)	Uniform ($\tau = \infty$)	24.81	21.52	9.48	19.99	30.46	33.22	33.70	35.15	15.03
	Upsampled ($\tau = 5$)	26.01	23.47	10.19	21.26	31.13	34.69	34.94	36.44	16.00
	Proportional ($\tau = 1$)	26.68	23.43	10.10	22.01	31.06	35.62	36.41	37.91	16.91
	MultiDDS-S	27.00	25.34	10.57	22.93	32.05	35.27	35.77	37.30	16.81
Static	Uniform ($\tau = \infty$)	24.47	21.72	8.13	17.84	29.18	33.37	34.22	35.99	15.31
	Upsampled ($\tau = 5$)	26.04	24.60	9.56	19.68	30.81	34.71	35.54	36.74	16.68
	Proportional ($\tau = 1$)	**26.85**	22.00	9.73	**21.02**	**32.57**	**36.27**	**37.37**	**38.29**	**17.57**
Bandit	dev-pgnorm	26.30	**23.88**	**10.41**	20.70	31.18	34.10	35.87	36.81	17.47

Table 4: BLEU on diverse TED data for many-to-one models.

	Size (k)	Avg 73.2	az 5.9	be 4.5	gl 10.0	sl 19.8*	tr 182.5	ru 208.4	pt 51.8*	cs 103.1
Ours	Base	22.87	11.91	17.19	26.64	22.56	22.66	21.83	34.87	25.17
Wang et al. (2020b)	Uniform* ($\tau = \infty$)	22.63*	8.81	14.80	25.22	27.32*	20.16	20.95	38.69*	25.11
	Upsampled* ($\tau = 5$)	24.00*	10.42	15.85	27.63	28.38*	21.53	21.82	40.18*	26.26
	Proportional* ($\tau = 1$)	24.88*	11.20	17.17	27.51	28.85*	23.09	22.89	41.60*	26.80
	MultiDDS-S*	25.52*	12.20	19.11	29.37	29.35*	22.81	22.78	41.55*	27.03
Static	Uniform ($\tau = \infty$)	20.30	8.10	12.09	24.35	19.21	20.53	20.22	33.99	23.95
	Upsampled ($\tau = 5$)	21.92	9.71	15.14	26.18	20.84	21.79	21.18	35.31	25.18
	Proportional ($\tau = 1$)	**23.60**	11.88	15.80	27.69	**22.90**	**23.73**	**22.90**	**37.38**	**26.49**
Bandit	dev-pgnorm	23.51	**12.18**	**18.00**	**27.76**	21.76	23.36	22.72	36.51	25.82

Table 5: BLEU on related TED data for many-to-one models. *Trained on larger data than publicly available.

outperforms the vanilla baseline and static size-proportional sampling in both directions, and for M2O also uniform sampling, as reported in Table 6. It performs slightly weaker than the static upsampling approach. Uniform sampling is competitive for O2M, because it evenly balances the target language occurrence. M2M bandits improve over the baseline as well, on average +0.6 for M2O and +1.2 for O2M (§B), with the largest gain of +3.6 BLEU on O2M for the lowest-resource languages.

There is no correlation between training data size and BLEU on the test set for the baseline, nor between the sampling frequencies of the bandit and training data size for any of the directions (in contrast to the TED experiments). The bandits pursue selective strategies with very frequent switches between facets. For M2O 11% of all training steps were done on nl, and more than half the languages were sampled in less than 0.5% steps each. For O2M, samples from fy, ga, ky, mg and ug were used in more than 3% of steps each, and again around half the languages were trained on for less than 0.5% steps. Comparing M2O and O2M top-5 sampled languages, we find 4 of those to be high-resourced (1M training examples) for M2O, but for O2M these are all mid to low-resourced with 27k-591k examples (details in §C). Surprisingly, the languages which are rarely sampled do not stand out with low translation quality. The selection of domains for the data sets is not controlled for in this benchmark (Zhang et al., 2021), so we suspect domain effects might be interfering with BLEU reporting, in that some test sets might be more specialized than others, especially low-resource languages which are mainly covered by technical or religious data sets in OPUS.

5 Related Work

Model-based data selection van der Wees et al. (2017) reported first empirical success of hand-crafted schedules for data from different domains which are chosen according to cross-entropy scores of RNN-NMT models. Wang et al. (2018) proposed an online data denoising approach, where noise is measured as the difference of log-probabilities between a learning model and the same model fine-tuned on small set of trusted data. Batches are composed of sentences with the highest contrastive data scores (CDS) corresponding to the least noisy sentences. Our approach is similar to the above in that the multi-armed bandit acts on the online learning success of the MT model, but

		M2O				O2M			
		All	Low	Med	High	All	Low	Med	High
Base	Zhang et al. (2021)	29.27	29.71	30.10	28.55	20.93	18.02	22.36	21.39
	Ours	28.41	29.53	27.54	28.44	19.78	19.72	18.65	20.51
Static	Uniform ($\tau = \infty$)	29.06	31.55	27.52	28.85	**22.07**	**23.68**	19.91	**22.67**
	Upsampled ($\tau = 5$)	30.15	**32.72**	**28.48**	**30.00**	**22.07**	**23.68**	**20.18**	22.49
	Proportional ($\tau = 1$)	28.07	29.70	27.29	27.80	19.39	18.84	18.05	20.49
Bandit	`dev-pgnorm`	**29.53**	31.64	27.73	29.66	20.30	21.77	18.23	20.91

Table 6: Average BLEU across languages pair groups for M2O & O2M models evaluated on OPUS100 test sets.

it is significantly cheaper since it does not require contrastive models nor a pre-defined schedule. Furthermore, the requirement of trusted data is lifted.

Difficulty-based curricula Kocmi and Bojar (2017) apply the idea of curriculum learning (Elman, 1993; Bengio et al., 2009) to RNN NMT by simple ordering data in buckets corresponding to increased difficulty. Zhang et al. (2018) combine non-reusable buckets of difficulties with a manual schedule and achieve small improvements on small data with RNNs. Platanios et al. (2019) apply a competence-based schedule with lengths and rarity to Transformer NMT that re-samples already used examples as long as they fall under the current competency. Many works on manually designed curricula note that presenting examples in the reverse order (hard-to-easy) works comparably well (Bengio et al., 2009; Wang et al., 2018; Zhang et al., 2018), which may be a sign of flawed intuitions. Our proposed solution groups data into facets rather than difficulty levels and reveals counterintuitive but effective schedules.

Learned curricula Apart from (Graves et al., 2017), whose curriculum learning bandits we adapt for NMT, (Kumar et al., 2019) is closest to our work. They frame the data selection task as an RL problem and define actions as data clusters corresponding to bins of CDS scores (Wang et al., 2018). The same idea of representative batches is reused for multi-armed bandits enhanced with state representations in (Kumar et al., 2021). In (Wang et al., 2020a) another RL algorithm is deployed for optimizing a distribution over training examples using the alignment of training and development gradients as rewards, requiring two backward passes on every step and an additional forward pass on an auxiliary neural net. In contrast to the RL-based approaches, we use light-weight bandits without state representations, which reduces memory and time complexities drastically.

Bandit learning in MT Multi-armed bandits were used in MT to improve general quality, either from online simulated user feedback (Sokolov et al., 2015, 2016, 2017; Kreutzer et al., 2017, 2018b) or from offline logs (Lawrence et al., 2017; Kreutzer et al., 2018a) for domain adaptation. Naradowsky et al. (2020) applied bandit algorithm to select the best NMT system for a particular translation task, when maintaining of multiple such systems is possible. More generally, RL approaches also seek to improve quality by focusing on more task-informed objectives (Shen et al., 2016) and improved approximations to the NMT policies (Bahdanau et al., 2017). Unlike these approaches, we treat the NMT model as a black box and do not intervene with its inner workings (see Figure 1).

Translationese vs. natural MT Toral et al. (2018) have shown that the original language a sentence has been written in has a big impact on translation quality, i.e., translating a sentence originally written in the source language is more difficult than translating (back) a sentence that was originally written in the target language and then translated into the source language. This second condition is 'unnatural' for the actual use case of translation systems, but occurs frequently in translation evaluations, if the same dataset is used for evaluating both translation directions. To avoid such artifacts, source sentences for evaluation should be been written originally in the source language (Barrault et al., 2019). Recently, Vanmassenhove et al. (2021) showed that MT outputs present lower lexical diversity than human produced texts. MT systems generating outputs closer in style to the original target text are preferred by human judges (Freitag et al., 2020a). Hence our motivation (§4.1) to

produce more natural sounding translations.

Multi-Faceted MT Multi-task learning (Caruana, 1997) for NMT was introduced by Luong et al. (2016) with the motivation to support a primary tasks with auxiliary data from related tasks. When understanding languages as tasks (Dong et al., 2015; Firat et al., 2016), one single MT model can be used to translate between a multitude of languages and in particular also between translation pairs that were not in the training set (Johnson et al., 2017; Aharoni et al., 2019). To address the problem of data imbalance Devlin (2019); Arivazhagan et al. (2019) proposed temperature sampling to upsample low-resource languages and downsample higher-resource ones (with $\tau > 1$), that has since turned into the go-to data weighting strategy (Freitag and Firat, 2020; Xue et al., 2020). While it is a convenient solution and often outperforms uniform weighting ($\tau = 1$), it reduces the characteristics of languages to their size and reflects the assumption of a zero-sum game in joint training (Xue et al., 2020), ignoring more complex interactions (Fan et al., 2020; Wang et al., 2021). Our experiments reveal that even very unbalanced and counter-intuitive schedules can lead to improved results across the board thanks to more intricate and automated sampling. Closest to our work are recent approaches to schedule data based on inter-facet gradient similarity (Wang et al., 2020a,b), which are more computationally expensive.

6 Discussion and Conclusion

We showed that a simple application of the EXP3 algorithm (Auer et al., 2002; Graves et al., 2017) to the training of a black-box NMT system is a cheaper and non-invasive alternative to task-specific expensively hand-crafted curricula and to heavy RL-based approaches. Bandit-optimized data usage leads to improved performance compared to the baselines across the board, and sometimes even faster convergence. On the difficult task of improving naturalness of translations we gained +0.5–0.9 BLEU on natural on average; on the multi-domain task up to 1.7 points on certain domains using 72% of the baseline's time to converge; on the multilingual MT task on average—by +1.2 points for translations of 94 languages into English, and by +0.6 points for the reverse.

We found intuitive explanations for the learned policies on some of the tasks, but our ability to interpret bandit actions with human reasoning is very limited especially when the number of facets and training steps grow, and also defeats the purpose of replacing possibly flawed human intuitions with learned curricula. As opposed to the expensive development cycles ("train-interpret-retrain") of post-training data interpretability methods (Koh and Liang, 2017) the bandits directly act on their understanding of what is beneficial for the task at hand. After training we can report for each model how much each facet actually mattered, which would increase the transparency of model reporting (Mitchell et al., 2019), especially for large-scale models (Raffel et al., 2020; Xue et al., 2020).

Finally, there are a few limitations of our approach: Being stateless, unlike RL approaches (Kumar et al., 2019), bandits might be short-sighted and keen on exploiting easy data first. Our experiments, though, show that this, with a sufficiently large exploration rate, does not seem to be the case for the tested applications and is not an obstacle to practical use. Another limitation are additional hyperparameters to be set (learning and exploration rates, and reward definitions). Again, we found it relatively easy to navigate in practice by stopping unpromising runs early (∼50k steps in our runs, cf. Figure 2); moreover, the hyperparameters tend to generalize across tasks. We believe that the flexibility provided by the reward definitions would allow to inject domain knowledge and/or signals from potentially multiple objectives, and prior knowledge of the data imbalance could be reflected in the exploration rate.

Our implementation of EXP3 samples facets in homogeneous batches, but the current SOTA models use heterogeneous ones (Arivazhagan et al., 2019). This introduces a limitation and a potential hindrance for optimization (Li and Gong, 2021), that we hope to address in future work by learning a sampling distribution over individual sentences. With steadily growing training data from more and more sources (Raffel et al., 2020; Xue et al., 2020), it would also be desirable to model facet hierarchies or intersectionalities, e.g., differentiating between domains and translationese vs. natural within each language pair for a multilingual model.

Acknowledgements

We would like to thank Markus Freitag for the help with the setup of the natural vs. translationese experiments, and Anselm Levskaya and the Flax team for their help with JAX/Flax debugging.

References

Roee Aharoni and Yoav Goldberg. 2020. Unsupervised domain clusters in pretrained language models. In *ACL*.

Roee Aharoni, Melvin Johnson, and Orhan Firat. 2019. Massively multilingual neural machine translation. In *ACL*.

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, Colin Cherry, et al. 2019. Massively multilingual neural machine translation in the wild: Findings and challenges. *CoRR*, abs/1907.05019.

Martín Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. 2019. Invariant risk minimization. *CoRR*, abs/1907.02893.

Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. 2002. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2–3).

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *EMNLP*.

Dzmitry Bahdanau, Philemon Brakel, Kelvin Xu, Anirudh Goyal, Ryan Lowe, Joelle Pineau, Aaron Courville, and Yoshua Bengio. 2017. An actor-critic algorithm for sequence prediction. In *ICLR*.

Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, et al. 2019. Findings of the 2019 conference on machine translation (WMT19). In *WMT*.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *ICML*.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. 2018. JAX: composable transformations of Python+NumPy programs. GitHub.

Rich Caruana. 1997. Multitask learning. *Machine learning*, 28(1):41–75.

Jacob Devlin. 2019. Multilingual BERT. https://github.com/google-research/bert/blob/master/multilingual.md.

Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-task learning for multiple language translation. In *ACL*.

Jeffrey L. Elman. 1993. Learning and development in neural networks: the importance of starting small. *Cognition*, 48(1).

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, et al. 2020. Beyond english-centric multilingual machine translation. *CoRR*, abs/2010.11125.

M. Amin Farajian, Marco Turchi, Matteo Negri, and Marcello Federico. 2017. Multi-domain neural machine translation through unsupervised adaptation. In *WMT*.

Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. 2016. Multi-way, multilingual neural machine translation with a shared attention mechanism. In *NAACL*.

Markus Freitag and Orhan Firat. 2020. Complete multilingual neural machine translation. In *WMT*.

Markus Freitag, George Foster, David Grangier, and Colin Cherry. 2020a. Human-paraphrased references improve neural machine translation. In *WMT*.

Markus Freitag, David Grangier, and Isaac Caswell. 2020b. BLEU might be guilty but references are not innocent. In *EMNLP*.

Guillem Gascó, Martha-Alicia Rocha, Germán Sanchis-Trilles, Jesús Andrés-Ferrer, and Francisco Casacuberta. 2012. Does more data always yield better translations? In *EACL*.

Alex Graves, Marc G. Bellemare, Jacob Menick, Remi Munos, and Koray Kavukcuoglu. 2017. Automated curriculum learning for neural networks. In *ICML*.

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. GitHub.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. 2017. Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation. *TACL*, 5:339–351.

Tom Kocmi and Ondřej Bojar. 2017. Curriculum learning and minibatch bucketing in neural machine translation. In *RANLP*.

Philipp Koehn and Christof Monz, editors. 2006. *Proceedings on the Workshop on Statistical Machine Translation*.

Pang Wei Koh and Percy Liang. 2017. Understanding black-box predictions via influence functions. In *ICML*.

Julia Kreutzer, Jasmijn Bastings, and Stefan Riezler. 2019. Joey NMT: A minimalist NMT toolkit for novices. In *EMNLP*.

Julia Kreutzer, Shahram Khadivi, Evgeny Matusov, and Stefan Riezler. 2018a. Can neural machine translation be improved with user feedback? In *NAACL*.

Julia Kreutzer, Artem Sokolov, and Stefan Riezler. 2017. Bandit structured prediction for neural sequence-to-sequence learning. In *ACL*.

Julia Kreutzer, Joshua Uyheng, and Stefan Riezler. 2018b. Reliability and learnability of human bandit feedback for sequence-to-sequence reinforcement learning. In *ACL*.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *EMNLP*.

Gaurav Kumar, George Foster, Colin Cherry, and Maxim Krikun. 2019. Reinforcement learning based curriculum optimization for neural machine translation. In *NAACL*.

Gaurav Kumar, Philipp Koehn, and Sanjeev Khudanpur. 2021. Learning curricula for multilingual training of neural machine translation systems. In *MT-Summit*.

Carolin Lawrence, Artem Sokolov, and Stefan Riezler. 2017. Counterfactual learning from bandit feedback under deterministic logging: A case study in statistical machine translation. In *EMNLP*.

Xian Li and Hongyu Gong. 2021. Robust optimization for multilingual translation with imbalanced data. *CoRR*, abs/2104.07639.

Thang Luong, Quoc V. Le, Ilya Sutskever, Oriol Vinyals, and Łukasz Kaiser. 2016. Multi-task sequence to sequence learning. In *ICLR*.

Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa D. Raji, and Timnit Gebru. 2019. Model cards for model reporting. In *FAT*.

Mathias Müller, Annette Rios, and Rico Sennrich. 2020. Domain robustness in neural machine translation. In *AMTA*.

Jason Naradowsky, Xuan Zhang, and Kevin Duh. 2020. Machine translation system selection from bandit feedback. In *AMTA*.

Graham Neubig and Junjie Hu. 2018. Rapid adaptation of neural machine translation to new languages. In *EMNLP*.

Toan Q. Nguyen and David Chiang. 2017. Transfer learning across low-resource, related languages for neural machine translation. In *JCNLP*.

MinhQuang Pham, Josep Maria Crego, and François Yvon. 2021. Revisiting Multi-Domain Machine Translation. *TACL*, 9.

Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabas Poczos, and Tom M Mitchell. 2019. Competence-based curriculum learning for neural machine translation. In *NAACL*.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *WMT*.

Ye Qi, Devendra Sachan, Matthieu Felix, Sarguna Padmanabhan, and Graham Neubig. 2018. When and why are pre-trained word embeddings useful for neural machine translation? In *NAACL*.

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. *JMRL*, 21(140):1–67.

Parker Riley, Isaac Caswell, Markus Freitag, and David Grangier. 2020. Translationese as a language in "multilingual" NMT. In *ACL*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *ACL*.

Shiqi Shen, Yong Cheng, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. 2016. Minimum risk training for neural machine translation. In *ACL*.

Artem Sokolov, Julia Kreutzer, Christopher Lo, and Stefan Riezler. 2016. Stochastic structured prediction under bandit feedback. In *NIPS*.

Artem Sokolov, Julia Kreutzer, Kellen Sunderland, Pavel Danchenko, Witold Szymaniak, Hagen Fürstenau, and Stefan Riezler. 2017. A shared task on bandit learning for machine translation. In *WMT*.

Artem Sokolov, Stefan Riezler, and Tanguy Urvoy. 2015. Bandit structured prediction for learning from partial feedback in statistical machine translation. In *MTSummit*.

Damien Teney, Ehsan Abbasnejad, and Anton van den Hengel. 2020. Unshuffling data for improved generalization. *CoRR*, abs/2002.11894.

Jörg Tiedemann and Lars Nygaard. 2004. The OPUS corpus - parallel & free. In *LREC*.

Antonio Toral, Sheila Castilho, Ke Hu, and Andy Way. 2018. Attaining the unattainable? reassessing claims of human parity in neural machine translation. In *WMT*.

Marlies van der Wees, Arianna Bisazza, and Christof Monz. 2017. Dynamic data selection for neural machine translation. In *EMNLP*.

Eva Vanmassenhove, Dimitar Shterionov, and Matthew Gwilliam. 2021. Machine translationese: Effects of algorithmic bias on linguistic complexity in machine translation. In *EACL*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS*.

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 *WMT*.

Xinyi Wang, Hieu Pham, Paul Michel, Antonios Anastasopoulos, Jaime Carbonell, and Graham Neubig. 2020a. Optimizing data usage via differentiable rewards. In *ICML*.

Xinyi Wang, Yulia Tsvetkov, and Graham Neubig. 2020b. Balancing training for multilingual neural machine translation. In *ACL*.

Zirui Wang, Yulia Tsvetkov, Orhan Firat, and Yuan Cao. 2021. Gradient vaccine: Investigating and improving multi-task optimization in massively multilingual models. In *ICLR*.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2020. mt5: A massively multilingual pre-trained text-to-text transformer. *CoRR*, abs/2010.11934.

Biao Zhang, Ankur Bapna, Rico Sennrich, and Orhan Firat. 2021. Share or not? Learning to schedule language-specific capacity for multilingual translation. In *ICLR*.

Biao Zhang, Philip Williams, Ivan Titov, and Rico Sennrich. 2020. Improving massively multilingual neural machine translation and zero-shot translation. In *ACL*.

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

A Hyperparameters

A.1 Transformer implementation

We abstained from adding the plethora of architecture and pre-processing tweaks common for systems competing in MT benchmarks, and experimented with bare bone Transformer models in order to reduce confounding effects, and keep the code and resulting experiments minimal and clean (Kreutzer et al., 2019).[6]

To verify the implementation, we tested it on the WMT14 `en-de` benchmark, where it scores 27.8 BLEU (without ensembling) vs. 27.3 reported in (Vaswani et al., 2017).

A.2 Natural vs. translationese

We used the same configuration as the `big` Transformer model from (Vaswani et al., 2017), except for the MLP dimension which was increased to 8192. Training on TPUv2, we used a learning rate of 0.01, warmup 10000, label smoothing of 0.1, dropout of 0.1 and 96 as batch size. The maximum length during training was set to 100. Decoding was done with beam size 4 and maximum length 140 during beam search.

Details on the LM classifier: The first LM was trained on monolingual news crawl data provided by the organizers of the WMT campaign, which comes from news sites originally written in the desired language. The second dataset was generated by (forward) translating data in the source language into the target language by a previously trained MT system. Note that, although we train only on MT generated data, we will use this last LM for identify both human and machine translated data in the training corpus. Our experiments show that this method can help identifying both types of translationese texts, probably due to the fact that MT output exacerbates the characteristics of translationese text. Inspired by (Riley et al., 2020), for each sentence we compare the model score of each of the LMs, and select the class corresponding to the one which produces a better score.

A.3 Multi-domain

Following (Müller et al., 2020), we applied the standard Moses preprocessing pipeline (removing non-printing chars, normalizing punctuation, tokenizing, truecasing and length filtering) to all splits of the data, including the test set. The Subtitles part was limited to 500k sentences and concatenated data was preprocessed jointly with 32,000 BPE merges (Sennrich et al., 2016), resulting in a 32,298 vocabulary entries. Maximum training length was 100 post-BPE tokens.

We used the same configuration as the `base` Transformer model from (Vaswani et al., 2017). Training on TPUv2 used learning rate 0.01, warmup 4,000, label smoothing 0.1, dropout 0.2 and nominal batch size 256. Decoding was done with beam size 5 and maximum length 256 during beam search. BLEU score were calculated with SacreBLEU on deBPE'ed and detokenized sentences w.r.t. similarly preprocessed references.

The bandits used the learning rate of 0.1 and exploration rate of 0.25, found by grid search over the range [0.001, 0.01, 0.1] and [0.5, 0.25, 0.1] respectively.

A.4 Multilingual

TED For TED we train the models on 4 V100 GPUs with a batch size of 64 sentences for 50k steps, a warmup period of 4k steps for a learning rate schedule with linear increase and square-root decay and a base learning rate of 0.0625. Training sentences up to a length of 512 tokens are considered. For inference, beam width is set to 4. Models are validated every 2k steps. for OPUS100 5. Bandit learning and exploration rate were tuned over a grid search over the range [0.001, 0.01, 0.01] and [0.1, 0.2, 0.3, 0.4, 0.5] respectively, with training up to 10k training steps. For the `diverse` task the best setting was (0.1, 0.3) and for the `related` task (0.01, 0.2).

OPUS100 For OPUS the models are trained with a total batch size of 256 sentences, 1k warmup steps and the same learning rate schedule as for TED. For inference we use beams of width 5. Models are trained for 500k steps and validated every 8k. The best configuration of bandit learning and exploration rate is (0.01, 0.5) for all settings (M2O, O2M, M2M).

SentencePiece For balanced subword representations we upsample all languages to the maximum size across languages and then using the SP option `large_corpus` to subsample uniformly from their concatenation.

[6] For our pre-processing pipeline, that is built on top of Tensorflow Datasets, we found that increasing shuffle buffer had a significant positive effect on baseline performance, therefore all experiments were performed for the shuffle buffer size value that was optimal for baselines.

	M2O				O2M			
	All	Low	Med	High	All	Low	Med	High
Base	21.31	25.94	20.10	19.90	18.36	15.64	17.78	20.00
`dev-pgnorm`	21.93	26.09	20.61	20.81	19.58	19.36	17.96	20.69

Table 7: **M2M:** Avg BLEU across languages for OPUS100's 94 test sets grouped by training corpus size as in (Zhang et al., 2021).

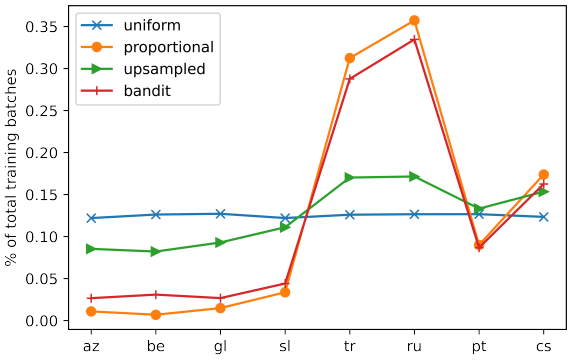

Figure 4: Ratio of training batches from each language throughout training for TED-related M20.

Setting	Lang.	% Train. Batches	Train. Size	Test BLEU
M2O	`nl-en`	10.7	1M	28.86
	`cs-en`	3.5	1M	27.39
	`ms-en`	3.0	1M	27.32
	`sh-en`	2.9	267k	28.30
	`sr-en`	2.8	1M	57.21
O2M	`en-fy`	5.9	54k	35.19
	`en-ga`	4.8	290k	12.20
	`en-ky`	4.7	27k	18.72
	`en-mg`	3.6	591k	16.52
	`en-ug`	3.6	72k	9.61
M2M	`as-en`	4.3	138k	29.24
	`ta-en`	2.5	227k	0.91
	`li-en`	2.1	26k	51.28
	`fa-en`	2.0	1M	20.28
	`da-en`	1.4	1M	19.81

Table 8: Top 5 sampled languages for M2O, O2M, and M2M OPUS.

B OPUS100 M2M

Table 7 lists the result for many-to-many translation for the OPUS100 benchmark.

C Multi-lingual bandit strategies

Figure 4 shows the ratio of training batches from each language, averaged across the complete training run. The corresponding diagram for the diverse subset is in Figure 3. Again, we find that the bandit mimics the proportional sampling strategy, with slight upsampling of the lowest-resource pairs.

Table 8 lists the top 5 sampled languages for each OPUS setting. For M2O these are largely high-resource pairs, for O2M low-resource pairs, and for M2M pairs with English as target were generally sampled more, but the top 5 are a mix of high- and resource languages.

D Bandit arm probabilities

In Figures 5 and 6 we plot the evolution of bandit arm (facet) sampling probabilities over time to illustrate the learned curricula for the `en-de` natural vs. translationese and multi-domain tasks. Best viewed in color.

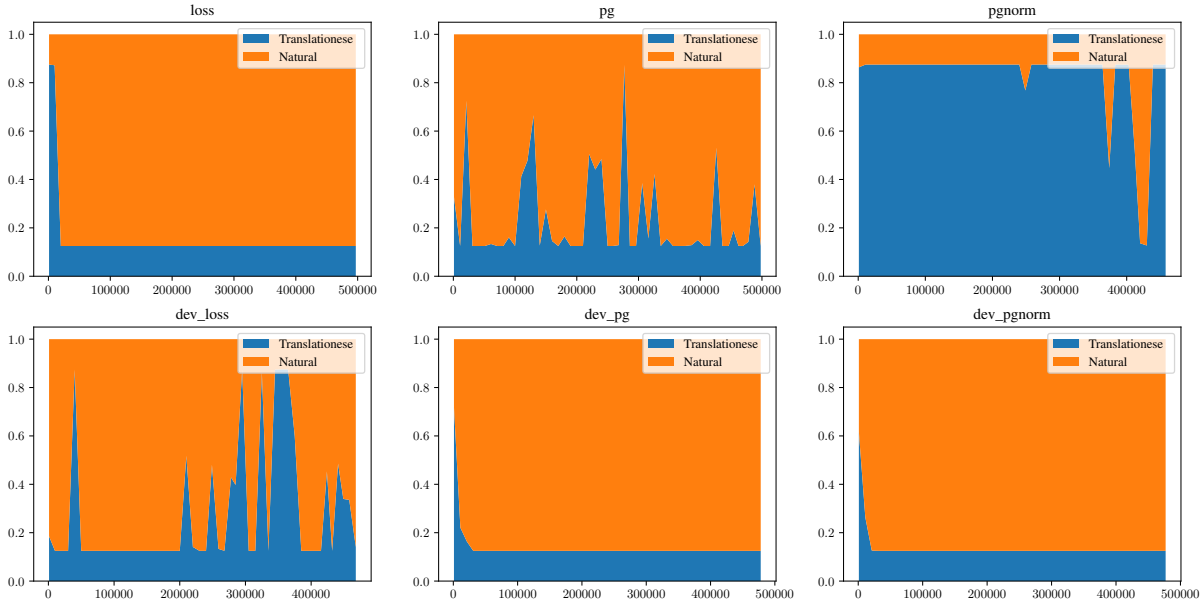

Figure 5: Evolution of probabilities during training on the WMT `en-de` task (without CDS filtering). See §4.1 for interpretation.

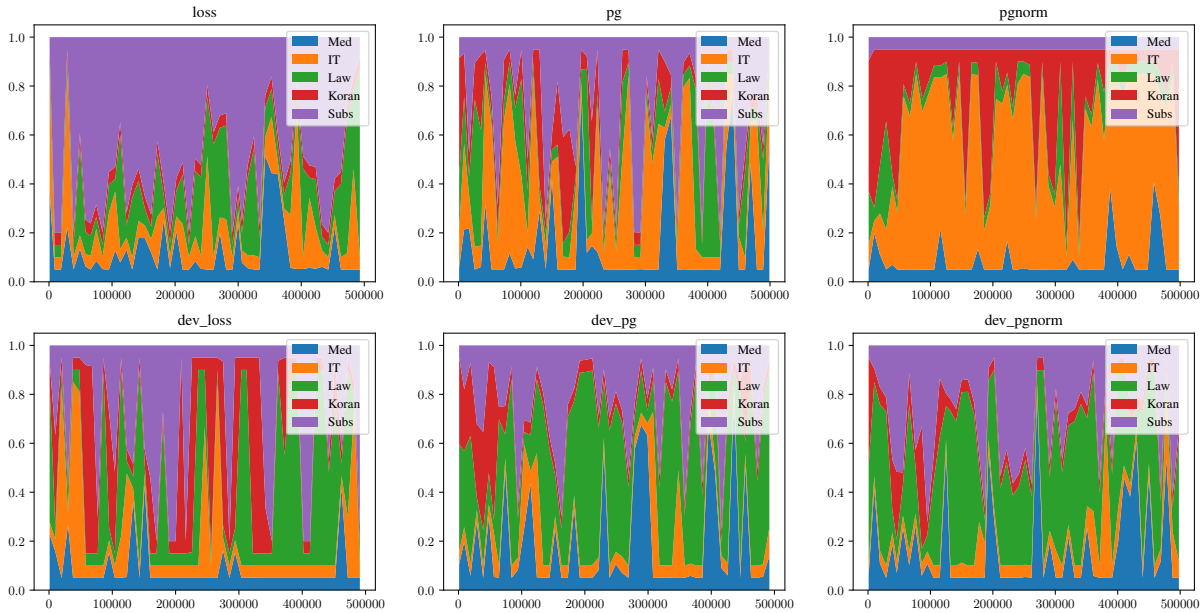

Figure 6: Evolution of probabilities during training on the multi-domain task. See §4.2 for interpretation.