Target Conditioned Sampling:
Optimizing Data Selection for Multilingual Neural Machine Translation

Xinyi Wang
Language Technologies Institute
Carnegie Mellon University
`xinyiw1@cs.cmu.edu`

Graham Neubig
Language Technologies Institute
Carnegie Mellon University
`gneubig@cs.cmu.edu`

Abstract

To improve low-resource Neural Machine Translation (NMT) with multilingual corpora, training on the most related high-resource language only is often more effective than using all data available (Neubig and Hu, 2018). However, it is possible that an intelligent data selection strategy can further improve low-resource NMT with data from other auxiliary languages. In this paper, we seek to construct a sampling distribution over all multilingual data, so that it minimizes the training loss of the low-resource language. Based on this formulation, we propose an efficient algorithm, Target Conditioned Sampling (TCS), which first samples a target sentence, and then conditionally samples its source sentence. Experiments show that TCS brings significant gains of up to 2 BLEU on three of four languages we test, with minimal training overhead[1].

1 Introduction

Multilingual NMT has led to impressive gains in translation accuracy of low-resource languages (LRL) (Zoph et al., 2016; Firat et al., 2016; Gu et al., 2018; Neubig and Hu, 2018; Nguyen and Chiang, 2018). Many real world datasets provide sentences that are multi-parallel, with the same content in a variety of languages. Examples include TED (Qi et al., 2018), Europarl (Koehn, 2005), and many others (Tiedemann, 2012). These datasets open up the tantalizing prospect of training a system on many different languages to improve accuracy, but previous work has found methods that use only a single related (HRL) often out-perform systems trained on all available data (Neubig and Hu, 2018). In addition, because the resulting training corpus is smaller, using a single language is also substantially faster to

train, speeding experimental cycles (Neubig and Hu, 2018). In this paper, we go a step further and ask the question: can we design an intelligent data selection strategy that allows us to choose the most relevant multilingual data to further boost NMT performance and training speed for LRLs?

Prior work has examined data selection from the view of domain adaptation, selecting good training data from out-of-domain text to improve in-domain performance. In general, these methods select data that score above a preset threshold according to some metric, such as the difference between in-domain and out-of-domain language models (Axelrod et al., 2011; Moore and Lewis, 2010) or sentence embedding similarity (Wang et al., 2017). Other works use all the data but weight training instances by domain similarity (Chen et al., 2017), or sample subsets of training data at each epoch (van der Wees et al., 2017). However, none of these methods are trivially applicable to multilingual parallel datasets, which usually contain many *different* languages from the *same* domain. Moreover, most of these methods need to pretrain language models or NMT models with a reasonable amount of data, and accuracy can suffer in low-resource settings like those encountered for LRLs (Duh et al., 2013).

In this paper, we create a mathematical framework for data selection in multilingual MT that selects data from *all* languages, such that minimizing the training objective over the sampled data approximately minimizes the loss of the LRL MT model. The formulation leads to an simple, efficient, and effective algorithm that first samples a target sentence and then conditionally samples which of several source sentences to use for training. We name the method Target Conditioned Sampling (TCS). We also propose and experiment with several design choices for TCS, which are especially effective for LRLs. On the TED multilin-

---

[1]The code can be found at `https://github.com/cindyxinyiwang/TCS`.

gual corpus (Qi et al., 2018), TCS leads to large improvements of up to 2 BLEU on three of the four languages we test, and no degradation on the fourth, with only slightly increased training time. To our knowledge, this is the first successful application of data selection to multilingual NMT.

2 Method

2.1 Multilingual Training Objective

First, in this section we introduce our problem formally, where we use the upper case letters X, Y to denote the random variables, and the corresponding lower case letters x, y to denote their actual values. Suppose our objective is to learn parameters θ of a translation model from a source language s into target language t. Let x be a source sentence from s, and y be the equivalent target sentence from t, given loss function $\mathcal{L}(x, y; \theta)$ our objective is to find optimal parameters θ^* that minimize:

$$\mathbb{E}_{x,y \sim P_S(X,Y)}[\mathcal{L}(x, y; \theta)] \qquad (1)$$

where $P_s(X, Y)$ is the data distribution of s-t parallel sentences.

Unfortunately, we do not have enough data to accurately estimate θ^*, but instead we have a multilingual corpus of parallel data from languages $\{s_1, S_2, ..., S_n\}$ all into t. Therefore, we resort to multilingual training to facilitate the learning of θ. Formally, we want to construct a distribution $Q(X, Y)$ with support over $s_1, s_2, ..., s_n$-T to augment the s-t data with samples from Q during training. Intuitively, a good $Q(X, Y)$ will have an expected loss

$$\mathbb{E}_{x,y \sim Q(X,Y)}[\mathcal{L}(x, y; \theta)] \qquad (2)$$

that is correlated with Eqn 1 over the space of all θ, so that training over data sampled from $Q(X, Y)$ can facilitate the learning of θ. Next, we explain a version of $Q(X, Y)$ designed to promote efficient multilingual training.

2.2 Target Conditioned Sampling

We argue that the optimal $Q(X, Y)$ should satisfy the following two properties.

First, $Q(X, Y)$ and $P_s(X, Y)$ should be **target invariant**; the marginalized distributions $Q(Y)$ and $P_s(Y)$ should match as closely as possible:

$$Q(Y) \approx P_s(Y) \qquad (3)$$

This property ensures that Eqn 1 and Eqn 2 are optimizing towards the same target Y distribution.

Second, to have Eqn 2 correlated with Eqn 1 over the space of all θ, we need $Q(X, Y)$ to be correlated with $P_s(X, Y)$, which can be loosely written as

$$Q(X, Y) \approx P_s(X, Y). \qquad (4)$$

Because we also make the target invariance assumption in Eqn 3,

$$\frac{Q(X, Y)}{Q(Y)} \approx \frac{P_s(X, Y)}{P_s(Y)} \qquad (5)$$

$$Q(X|Y) \approx P_s(X|Y). \qquad (6)$$

We call this approximation of $P_s(X|Y)$ by $Q(X|Y)$ **conditional source invariance**. Based on these two assumptions, we define Target Conditioned Sampling (TCS), a training framework that first samples $y \sim Q(Y)$, and then conditionally samples $x \sim Q(X|y)$ during training. Note $P_s(X|Y = y)$ is the optimal back-translation distribution, which implies that back-translation (**?**) is a particular instance of TCS.

Of course, we do not have enough s-t parallel data to obtain a good estimate of the *true* back-translation distribution $P_s(X|y)$ (otherwise, we can simply use that data to learn θ). However, we posit that even a small amount of data is sufficient to construct an adequate data selection policy $Q(X|y)$ to sample the sentences x from multilingual data for training. Thus, the training objective that we optimize is

$$\mathbb{E}_{y \sim Q(Y)} \mathbb{E}_{x \sim Q(X|y)}[\mathcal{L}(x, y; \theta)] \qquad (7)$$

Next, in Section 2.3, we discuss the choices of $Q(Y)$ and $Q(X|y)$.

2.3 Choosing the Sampling Distributions

Choosing $Q(Y)$. Target invariance requires that we need $Q(Y)$ to match $P_s(Y)$, which is the distribution over the target of s-t. We have parallel data from multiple languages $s_1, s_2, ..., s_n$, all into t. Assuming no systematic inter-language distribution differences, a uniform sample of a target sentence y from the multilingual data can approximate $P_s(Y)$. We thus only need to sample y uniformly from the union of all extra data.

Choosing $Q(X|y)$. Choosing $Q(X|y)$ to approximate $P_s(X|y)$ is more difficult, and there are a number of methods could be used to do so. To do so, we note that conditioning on the same target y and restricting the support of $P_s(X|y)$ to the

sentences that translate into y in at least one of s_i-t, $P_s(X = x|y)$ simply measures how likely x is in s. We thus define a heuristic function $\text{sim}(x, s)$ that approximates the probability that x is a sentence in s, and follow the data augmentation objective in Wang et al. (2018) in defining this probability according to

$$Q^*(x|y) = \frac{\exp\left(\text{sim}(x, s)/\tau\right)}{\sum_{x'} \exp\left(\text{sim}(x', s)/\tau\right)} \quad (8)$$

where is a temperature parameter that adjusts the peakiness of the distribution.

2.4 Algorithms

The formulation of $Q(X, Y)$ allows one to sample multilingual data with the following algorithm:

1. Select the target y based on $Q(y)$. In our case we can simply use the uniform distribution.
2. Given the target y, gather all data $(x_i, y) \in s_1, s_2, ...s_n$-$t$ and calculate $\text{sim}(x_i, s)$
3. Sample (x_i, y) based on $Q(X|y)$

The algorithm requires calculating $Q(X|y)$ repeatedly during training. To reduce this overhead, we propose two strategies for implementation: 1) **Stochastic**: compute $Q(X|y)$ before training starts, and dynamically sample each minibatch using the precomputed $Q(X|y)$; 2) **Deterministic**: compute $Q(X|y)$ before training starts and select $x' = \text{argmax}_x Q(x|y)$ for training. The deterministic method is equivalent to setting τ, the degree of diversity in $Q(X|y)$, to be 0.

2.5 Similarity Measure

In this section, we define two formulations of the similarity measure $\text{sim}(s, x)$, which is essential for constructing $Q(X|y)$. Each of the similarity measures can be calculated at two granularities: 1) language level, which means we calculate one similarity score for each language based on all of its training data; 2) sentence level, which means we calculate a similarity score for each sentence in the training data.

Vocab Overlap provides a crude measure of surface form similarity between two languages. It is efficient to calculate, and is often quite effective, especially for low-resource languages. Here we use the number of character n-grams that two languages share to measure the similarity between the two languages.

LRL	Train	Dev	Test	HRL	Train
aze	5.94k	671	903	tur	182k
bel	4.51k	248	664	rus	208k
glg	10.0k	682	1007	por	185k
slk	61.5k	2271	2445	ces	103k

Table 1: Statistics of our datasets.

We can calculate the language-level similarity between S_i and S

$$\text{sim}_{\text{vocab-lang}}(s_i, s) = \frac{|\text{vocab}_k(s) \cap \text{vocab}_k(s_i)|}{k}$$

$\text{vocab}_k(\cdot)$ represents the top k most frequent character n-grams in the training data of a language. Then we can assign the same language-level similarity to all the sentences in s_i.

This can be easily extended to the sentence level by replacing $\text{vocab}_k(s_i)$ to the set of character n-grams of all the words in the sentence x.

Language Model trained on s can be used to calculate the probability that a data sequence belongs to s. Although it might not perform well if s does not have enough training data, it may still be sufficient for use in the TCS algorithm. The language-level metric is defined as

$$\text{sim}_{\text{LM-lang}}(s_i, s) = \exp\left(\frac{\sum_{c_i \in s_i} \text{NLL}_s(c_i)}{|c_i \in s_i|}\right)$$

where $\text{NLL}_s(\cdot)$ is negative log likelihood of a character-level LM trained on data from s. Similarly, the corresponding sentence level metric is the LM probability over each sentence x.

3 Experiment

3.1 Dataset and Baselines

We use the 58-language-to-English TED dataset (Qi et al., 2018). Following the setup in prior work (Qi et al., 2018; Neubig and Hu, 2018), we use three low-resource languages Azerbaijani (aze), Belarusian (bel), Galician (glg) to English, and a slightly higher-resource dataset, Slovak (slk) to English.

We use multiple settings for baselines: 1) Bi: each LRL is paired with its related HRL, following Neubig and Hu (2018). The statistics of the LRL and their corresponding HRL are listed in Table 1; 2) All: we train a model on all 58 languages; 3) Copied: following Currey et al. (2017), we use the union of all English sentences as monolingual data by copying them to the source side.

Sim	Method	aze	bel	glg	slk
-	Bi	10.35	15.82	27.63	26.38
-	All	10.21	17.46	26.01	26.64
-	copied	9.54	13.88	26.24	26.77
Back-Translate	TCS	7.79	11.50	27.45	28.44
LM-sent	TCS-D	10.34	14.68	27.90	27.29
LM-sent	TCS-S	$10.95^{\dagger}$	17.15	27.91	27.24
LM-lang	TCS-D	10.76	14.97	27.92	28.40
LM-lang	TCS-S	$\mathbf{11.47}^{*}$	17.61	$28.53^{\dagger}$	$\mathbf{28.56}^{*}$
Vocab-sent	TCS-D	10.68	16.13	27.29	27.03
Vocab-sent	TCS-S	$11.09^{\dagger}$	16.30	$28.36^{\dagger}$	27.01
Vocab-lang	TCS-D	10.58	16.32	28.17	28.27^{*}
Vocab-lang	TCS-S	$\mathbf{11.46}^{*}$	$\mathbf{17.79}$	$\mathbf{29.57}^{*}$	28.45^{*}

Table 2: BLEU scores on four languages. Statistical significance (Clark et al., 2011) is indicated with $*$ ($p < 0.001$) and $\dagger$ ($p < 0.05$), compared with the best baseline.

3.2 Experiment Settings

A standard sequence-to-sequence (Sutskever et al., 2014) NMT model with attention is used for all experiments. Byte Pair Encoding (BPE) (Sennrich et al., 2016; Kudo and Richardson, 2018) with vocabulary size of 8000 is applied for each language individually. Details of other hyperparameters can be found in Appendix A.1.

3.3 Results

We test both the Deterministic (TCS-D) and Stochastic (TCS-S) algorithms described in Section 2.4. For each algorithm, we experiment with the similarity measures introduced in Section 2.5. The results are listed in Table 2.

Of all the baselines, Bi in general has the best performance, while All, which uses all the data and takes much longer to train, generally hurts the performance. This is consistent with findings in prior work (Neubig and Hu, 2018). Copied is only competitive for slk, which indicates the gain of TCS is not simply due to extra English data.

TCS-S combined with the language-level similarity achieves the best performance for all four languages, improving around 1 BLEU over the best baseline for aze, and around 2 BLEU for glg and slk. For bel, TCS leads to no degradation while taking much less training time than the best baseline All.

TCS-D vs. TCS-S. Both algorithms, when using document-level similarity, improve over the baseline for all languages. TCS-D is quite effective without any extra sampling overhead. TCS-S outperforms TCS-D for all experiments, indicat-

Sim	Model	aze	bel	glg	slk
-	Bi	11.87	18.03	28.70	26.77
-	All	10.87	17.77	25.49	26.28
-	copied	10.74	17.19	29.75	27.81
LM-lang	TCS-D	11.97	17.17	30.10	28.78^{*}
LM-lang	TCS-S	$\mathbf{12.55}^{\dagger}$	17.23	$30.69^{\dagger}$	28.95^{*}
Vocab-lang	TCS-D	12.30	$18.96^{\dagger}$	$\mathbf{31.10}^{*}$	$\mathbf{29.35}^{*}$
Vocab-lang	TCS-S	12.37	$\mathbf{19.83}^{\dagger}$	$30.94^{\dagger}$	29.00^{*}

Table 3: BLEU scores using SDE as word encoding. Statistical significance is indicated with $*$ ($p < 0.001$) and $\dagger$ ($p < 0.05$), compared with the best baseline.

ing the importance of diversity in the training data.

Sent. vs. Lang. For all experiments, language-level outperforms the sentence-level similarity. This is probably because language-level metric provides a less noisy estimation, making $Q(x|y)$ closer to $P_s(x|y)$.

LM vs. Vocab. In general, the best performing methods using LM and Vocab are comparable, except for glg, where Vocab-lang outperforms LM-lang by 1 BLEU. Slk is the only language where LM outperformed Vocab in all settings, probably because it has the largest amount of data to obtain a good language model. These results show that easy-to-compute language similarity features are quite effective for data selection in low-resource languages.

Back-Translation TCS constructs $Q(X|y)$ to sample augmented multilingual data, when the LRL data cannot estimate a good back-translation model. Here we confirm this intuition by replacing the $Q(X|y)$ in TCS with the back-translations generated by the model trained on the LRLs. To make it comparable to Bi, we use the sentence from the LRL and its most related HRL if there is one for the sampled y, but use the back-translated sentence otherwise. Table 2 shows that for slk, back-translate achieves comparable results with the best similarity measure, mainly because slk has enough data to get a reasonable back-translation model. However, it performs much worse for aze and bel, which have the smallest amount of data.

3.4 Effect on SDE

To ensure that our results also generalize to other models, specifically ones that are tailored for better sharing of information across languages, we also test TCS on a slightly different multilingual NMT model using soft decoupled encoding (SDE;
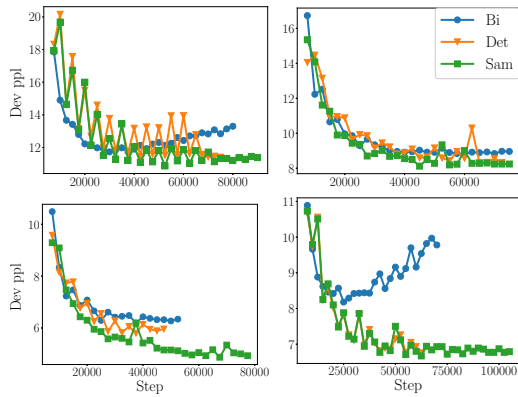
Figure 1: Development set perplexity vs. training steps. *Top left*: aze. *Top right*: bel. *Bottom left*: glg. *Bottom right*: slk.

Wang et al. (2019)), a word encoding method that assists lexical transfer for multilingual training. The results are shown in Table 3. Overall the results are stronger, but the best TCS model outperforms the baseline by 0.5 BLEU for aze, and around 2 BLEU for the rest of the three languages, suggesting the orthogonality of data selection and better multilingual training methods.

3.5 Effect on Training Curves

In Figure 1, we plot the development perplexity of all four languages during training. Compared to Bi, TCS always achieves lower development perplexity, with only slightly more training steps. Although using all languages, TCS is able to decrease the development perplexity at similar rate as Bi. This indicates that TCS is effective at sampling helpful multilingual data for training NMT models for LRLs.

4 Conclusion

We propose Target Conditioned Sampling (TCS), an efficient data selection framework for multilingual data by constructing a data sampling distribution that facilitates the NMT training of LRLs. TCS brings up to 2 BLEU improvements over strong baselines with only slight increase in training time.

Acknowledgements

References

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

Boxing Chen, Colin Cherry, George Foster, and Samuel Larkin. 2017. Cost weighting for neural machine translation domain adaptation. In *WMT*.

Jonathan Clark, Chris Dyer, Alon Lavie, and Noah Smith. 2011. Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *ACL*.

Anna Currey, Antonio Valerio Miceli Barone, and Kenneth Heafield. 2017. Copied monolingual data improves low-resource neural machine translation. In *WMT*.

Kevin Duh, Graham Neubig, Katsuhito Sudoh, and Hajime Tsukada. 2013. Adaptation data selection using neural language models: Experiments in machine translation. In *ACL*.

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

Jiatao Gu, Hany Hassan, Jacob Devlin, and Victor O. K. Li. 2018. Universal neural machine translation for extremely low resource languages. *NAACL*.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation.

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

Robert C. Moore and William D. Lewis. 2010. Intelligent selection of language model training data. In *ACL*.

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

Toan Q. Nguyen and David Chiang. 2018. Transfer learning across low-resource, related languages for neural machine translation. In *NAACL*.

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

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

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *NIPS*.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *LREC*.

Rui Wang, Andrew Finch, Masao Utiyama, and Eiichiro Sumita. 2017. Sentence embedding for neural machine translation domain adaptation. In *ACL*.

Xinyi Wang, Hieu Pham, Philip Arthur, and Graham Neubig. 2019. Multilingual neural machine translation with soft decoupled encoding. In *ICLR*.

Xinyi Wang, Hieu Pham, Zihang Dai, and Graham Neubig. 2018. Switchout: an efficient data augmentation algorithm for neural machine translation. In *EMNLP*.

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

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low resource neural machine translation. *EMNLP*.

A   Appendix

A.1   Model Details and Hyperparameters

- The LM similarity is calculated using a character-level LM[2]
- We use character n-grams with $n = \{1, 2, 3, 4\}$ for Vocab similarity and SDE.
- During training, we fix the language order of multilingual parallel data for each LRL, and only randomly shuffle the parallel sentences for each language. Therefore, we control the effect of the order of training data for all experiments.
- For TCS-S, we search over $\tau = \{0.01, 0.02, 0.1\}$ and pick the best model based on its performance on the development set.

---

[2]We sligtly modify the LM code from `https://github.com/zihangdai/mos` for our experiments.