Online Learning over Time in Adaptive Neural Machine Translation

Thierry Etchegoyhen* **David Ponce*** **Harritxu Gete Ugarte** **Victor Ruiz Gómez**
Vicomtech Foundation, Basque Research and Technology Alliance (BRTA)
{tetchegoyhen,adponce,hgete,vruiz}@vicomtech.org

Abstract

Adaptive Machine Translation purports to dynamically include user feedback to improve translation quality. In a post-editing scenario, user corrections of machine translation output are thus continuously incorporated into translation models, reducing or eliminating repetitive error editing and increasing the usefulness of automated translation. In neural machine translation, this goal may be achieved via online learning approaches, where network parameters are updated based on each new sample. This type of adaptation typically requires higher learning rates, which can affect the quality of the models over time. Alternatively, less aggressive online learning setups may preserve model stability, at the cost of reduced adaptation to user-generated corrections. In this work, we evaluate different online learning configurations over time, measuring their impact on user-generated samples, as well as separate in-domain and out-of-domain datasets. Results in two different domains indicate that mixed approaches combining online learning with periodic batch fine-tuning might be needed to balance the benefits of online learning with model stability.

1 Introduction

Machine Translation (MT) quality has increased significantly in recent years, notably with the advent of modern Neural Machine Translation (NMT) approaches (Bahdanau et al., 2015; Vaswani et al., 2017). Despite this progress, machine translated output requires post-editing in many cases, a process which is made more taxing when the same errors are repeated by MT systems segment after segment.

To tackle this issue, adaptive approaches to machine translation aim to incorporate user feedback, oftentimes in post-editing scenarios (Turchi et al., 2017), although on-the-fly adaptation is also relevant for interactive machine translation (Peris et al., 2017). In NMT, responsive model adaptation can be achieved via online learning approaches, where network parameters are updated based on each new sample of user-edited data. To perform this type of adaption from single data points, higher optimiser learning rates (LR) are typically required, which can affect the quality of the models over time. Alternatively, less aggressive online learning setups may preserve model stability, at the cost of reduced adaptation to user-generated corrections.

In this work, we study the evolution of online learning over time in a post-editing scenario, to determine optimal configurations in terms of both adaptation to continuous user input and model stability. For this purpose, we examine the behaviour of four different gradient-descent optimisers in two different domains, with varying learning rates, and evaluate their behaviour as the number of samples increases over time.

To measure system responsiveness to user input over time, we evaluate online learning on dynamically increasing sets of samples formed by simulated user corrections. To determine model stability as online learning is performed, we also measure the quality of the MT models on static test sets pertaining to the domain at hand and on out-of-domain datasets, as additional measures of model evolution over time via online learning. Additionally, we compare the best online learning variants to models trained via batch fine-tuning on accumulated user data.

To our knowledge, this type of evaluation has not been previously explored and our results can support further work on online learning for NMT, as well as help practitioners in the field determine optimal configurations to design responsive and balanced adaptive MT systems.

[*] Equal contribution.

Proceedings of Recent Advances in Natural Language Processing, pages 411–420
Sep 1–3, 2021.
https://doi.org/10.26615/978-954-452-072-4_047

2 Related Work

Most studies of online learning for machine translation have taken place in the context of Statistical Machine Translation (SMT) (Brown et al., 1990; Koehn, 2010). Several methods have thus been proposed to adapt phrase tables and language models of SMT models, in post-editing, interactive or streaming scenarios (Hardt and Elming, 2010; Ortiz-Martínez et al., 2010; Levenberg et al., 2010; Bertoldi et al., 2014). Evaluations of user productivity in adaptive SMT have notably shown a significant overall reduction of effort in post-editing scenarios (Bentivogli et al., 2015).

In Neural Machine Translation, comparatively fewer studies have been dedicated to online learning approaches. Turchi et al. (2017) explore different strategies based on a posteriori integration of human post-edits, a priori adaptation by tuning to similar sentences in the training data, and a combination of both, showing substantial improvements over static models. Peris et al. (2017) compared SMT and NMT models in interactive MT scenarios, demonstrating significant improvements in effort reduction with the latter over strong phrase-based systems. In Peris and Casacuberta (2019), online training of NMT models is evaluated under both post-editing and interactive scenarios. Similarly to the present work, they compared different optimisers with varying learning rates on datasets covering five different domains and different scenarios, assuming availability or lack of in-domain data prior to online learning. We complement their work in the present study by measuring the precise evolution of online learning over time and comparing it to batch fine-tuning at different time steps.[1]

Several user-centric studies have also demonstrated the usefulness of online learning for NMT, via analyses of user effort in static and adaptive environments (Karimova et al., 2018; Simianer et al., 2019; Domingo et al., 2019, 2020).

3 Experimental Setup

In this section, we describe in turn the core components of our experiments, namely the selected corpora, the training modalities of the different types of NMT models, and the selected optimisers.

3.1 Corpora

We first selected four datasets to train a generic model based on out-of-domain data.[2] To mimic a typical multi-domain generic model, we selected the following corpora: Europarl (Koehn, 2005), MultiUN (Eisele and Chen, 2010), OpenSubs (Lison and Tiedemann, 2016) and CC-Align (El-Kishky et al., 2020). Each of the four corpora was downsampled to the first 1M parallel sentences and the resulting datasets merged into a unique parallel corpus (*Generic*), from which development and test sets were extracted via uniform sampling.

As a basis for online learning, we selected two separate domain-specific datasets. In both cases, we used publicly available datasets and followed a similar methodology: the available test sets were used as is, to measure in-domain model stability over time, and are referred to as *static in-domain* test sets; the first 100K of the training sets were selected to simulate user post-editing, with the reference translations taken to be the post-edited version of the translated source segments, following standard practices in experimental protocols to evaluate MT adaptation and online learning (Ortiz-Martínez, 2016; Peris and Casacuberta, 2019). We refer to these datasets as *dynamic in-domain*, which are used to both perform incremental training and test the models in an online learning scenario. Dynamic ID datasets are further split in gradually increasing subsets of order 10^n, with $n \in 0, 1, 2, 3, 4, 5$, starting from the first sentence.

As our first in-domain (ID) test case, we selected the TED corpus (Cettolo et al., 2012), using *tst2015* as development set, *tst2016* as test set and the first 100K pairs of the 2016 training set as dynamic ID set for this domain.[3] As this corpus consists of first-person presentations on varied scientific or technological topics, it is markedly different from the datasets selected to train the generic domain.

We chose NewsCommentary v16 as our second in-domain dataset, in the concatenated version available on OPUS, using the first 100K pairs as online training data, the next 1522 pairs as development data, and the last 3000 as test set. This corpus consists of third-person news commentary and is thus relatively closer to the generic corpora in terms of topics and style.

[1] Peris and Casacuberta (2019) also include a scenario where online learning is applied over models first trained via batch fine-tuning on in-domain data, a setup which differs from our experiments.

[2] Unless otherwise specified, all datasets are those available on the OPUS repository (Tiedemann, 2012), as of April 2021.

[3] We used the version of the corpus available here: https://wit3.fbk.eu/2016-01-d

	Train	Test	Dev
Generic	3,726,891	2,000	1,000
TED	100,000	1,197	1,155
NewCommentary	100,000	3,000	1,522

Table 1: Corpora statistics (English-Spanish)

All corpora were tokenised and truecased with Moses scripts (Koehn, 2005) and words segmented via joint Byte Pair Encoding (Sennrich et al., 2016), using 30K merge operations. Statistics of the prepared corpora are summarised in Table 1.

3.2 Models

All translation models were based on the Transformer-base architecture (Vaswani et al., 2017) and built with the MarianNMT toolkit (Junczys-Dowmunt et al., 2018). The models consist of 6-layer encoders and decoders, feed-forward networks of 2048 units, embeddings vectors of dimension 512 and 8 attention heads. The dropout rate between layers is 0.1 and embeddings for the source, target and output layers were tied.

For the generic static models, we used the Adam optimiser (Kingma and Ba, 2015) with $\alpha = 0.0003$, $\beta_1 = 0.9$, $\beta_2 = 0.98$ and $\epsilon = 10^{-9}$. The learning rate was set to increase linearly for the first 16,000 training steps and decrease thereafter proportionally to the inverse square root of the corresponding step. We set the working memory to 6000MB and with a mini-batch set to automatically fit the specified memory. The validation data were evaluated every 3,500 steps and patience was set to 10.

For the online models, trained on the dynamic ID sets, the models were updated incrementally, with batches of one source-reference pair and a single update of the network based on the sample at hand.[4] Each version of the model resulting from an update as described was taken as the basis for the next online update. We selected four representative optimisers, described in the next section.

To measure the impact of domain variation, all models were trained for translation from English to Spanish. Evaluation was performed on the BLEU metric (Papineni et al., 2002), computed with the sacreBLEU toolkit (Post, 2018).[5]

3.3 Optimisers

Stochastic gradient descent (SGD) (Robbins and Monro, 1951) is one of the most common methods to estimate the parameters of a network, given the gradient of an error function, as it computes estimates on a per-sample basis. In NMT, SGD usually takes the form of mini-batch SGD, where the gradient is computed as the average gradient of the samples in a mini-batch; we use SGD as a shortcut for mini-batch SGD in what follows.

Several optimisations have been proposed to address some limitations of SGD, in particular methods that include a parameter-level update of the learning rate. Among these approaches, Adagrad (Duchi et al., 2011) uses past gradients for each parameter to compute parameter-level updates. Adadelta (Zeiler, 2012) extends it by mainly restricting the accumulation of past gradients to a fixed window size, an approach independently proposed as the basis of the RMSProp optimiser.[6] Another popular alternative is Adam (*op. cit*), which includes bias-corrected estimates of the 1st and 2nd moment, and is the default optimiser to train Transformer models in toolkits such as MarianNMT.

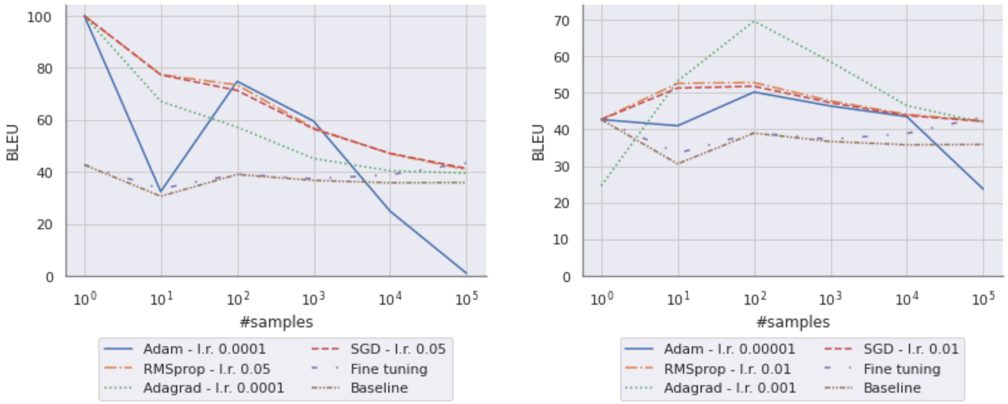
As indicated in Section 2, previous studies in online learning for NMT have compared the aforementioned parameter update methods, reaching different conclusions. Thus, Turchi et al. (2017) concluded that vanilla SGD was the optimal optimiser overall in their experiments, where the learning rate was fixed to 0.001 for all optimisers, whereas Peris and Casacuberta (2019) reached the conclusion that Adadelta, and to a lesser degree, SGD, were optimal after selecting the learning rate for each optimiser separately via grid-search on development sets. To gain further insights on optimal configurations for online learning, we selected four of the main optimisers, namely SGD, Adam, Adagrad and RMSProp, and measured the impact of different learning rates at different points in time, as described in the next section.[7]

[4] Peris (2020) evaluated the use of multiple updates on each sample for online learning, noting that it did not lead to significant improvements overall.

[5] The TER metric (Snover et al., 2006) was also used in internal experiments as a measure of post-editing effort. As the results obtained with this metric were highly correlated with those reported in this work for BLEU, we did not include them for clarity of presentation in the available space.

[6] Adadelta can also be computed with a second type of estimate, using past updates instead of Lasso regularisation. To limit our experiments to the main optimiser variants, we only considered the first update rule in Adadelta and refer to it as RMSProp, to avoid confusion over which version of the Adadelta updates is used.

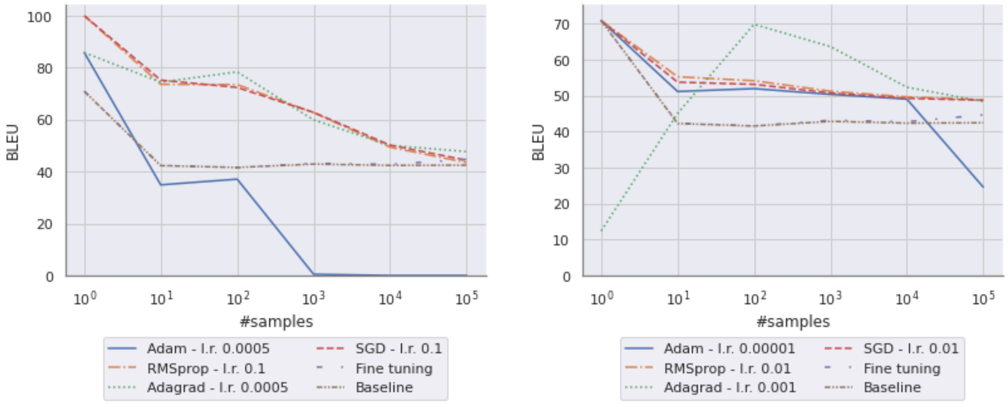
[7] The implementation of RMSProp in MarianNMT is our own; all others are based on the toolkit default implementation.



(a) Best learning rate @1

(b) Best learning rate @100K

Figure 1: BLEU scores with aggressive and conservative learning rates on the TED dynamic set



(a) Best learning rate @1

(b) Best learning rate @100K

Figure 2: BLEU scores with aggressive and conservative learning rates on the NewsCommentary dynamic set

4 Online Learning Over Time

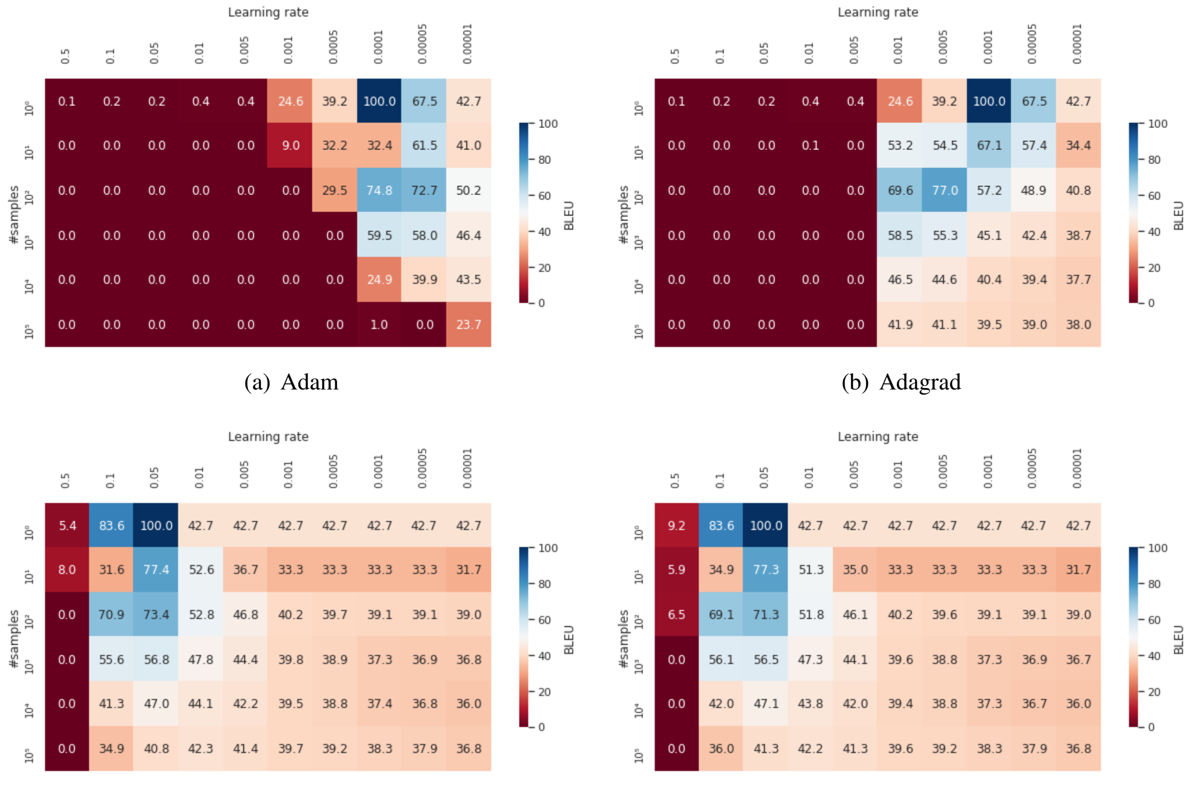
To evaluate the impact of online learning over time, we first evaluate the models against the references of the dynamic in-domain training set after 10^n online learning updates with each of the selected optimisers, with $n \in \{0, 1, 2, 3, 4, 5\}$.

4.1 Impact of Learning Rate

An important component of parameter update is the learning rate, which determines the amplitude of the updates. For online learning, a critical choice needs to be made between aggressive and conservative updates, based on high or low learning rates, respectively. The former may provide rapid adaptation, at the risk of deteriorating the network from overfitting to the samples, whereas the latter may delay or dilute the expected model adaptation, thus reducing the positive impact of online learning.

To measure both extremes, we selected the best learning rates for each optimiser according to the BLEU scores obtained on the dynamic datasets on the first (best@1) and last samples (best@100K), for each of the two selected domains. To determine an optimal learning rate for the first update, we randomly sampled 100 sentences and computed the BLEU score with all learning rates variants on the dynamic ID set, selecting the learning rate with the best average BLEU. This was meant to limit the impact of the characteristics of the first sentence in these datasets, which might not be representative of the data distribution. For the last update, all data points were considered to determine the best BLEU scores and associated learning rate.

Figure 1 and Figure 2 show the evolution of the optimisers for TED and NewsCommentary, respectively. We also include the evolution of the baseline generic models and models trained via batch fine-tuning over the available data (1 and 100K samples



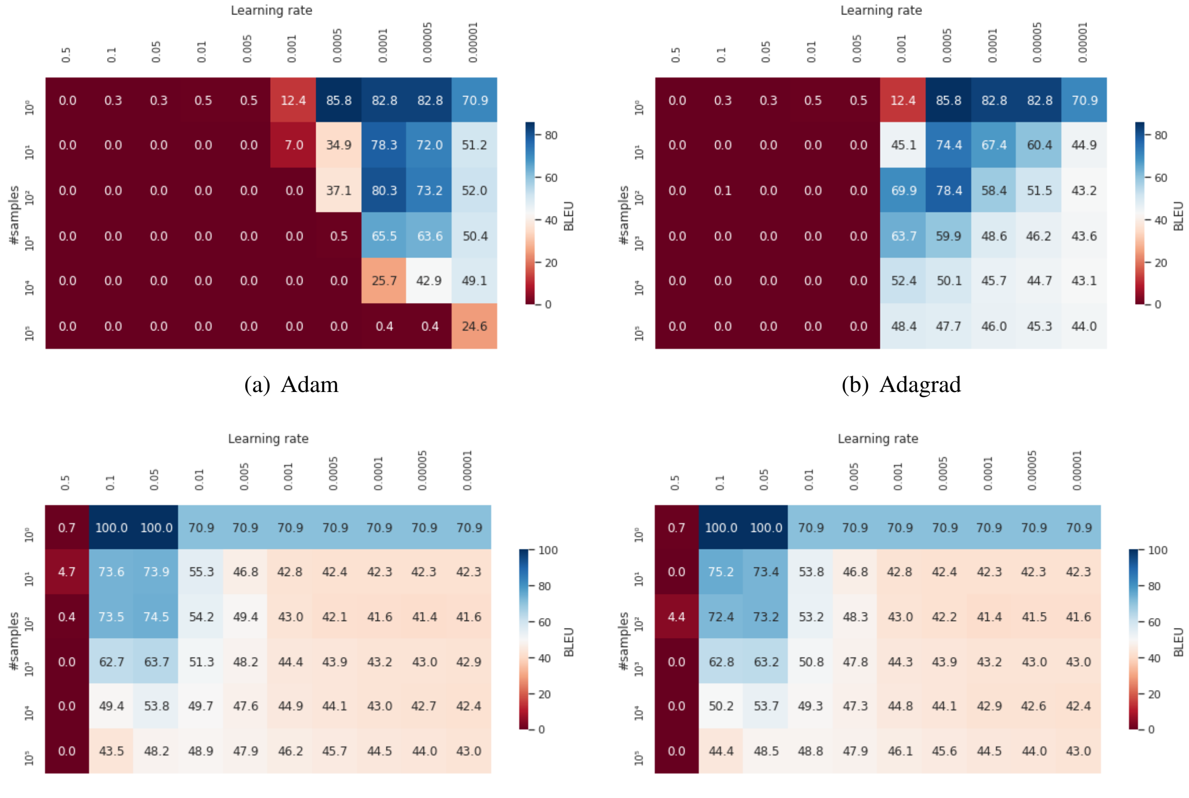
(a) Adam

(b) Adagrad

(c) RMSProp

(d) SGD

Figure 3: BLEU scores as a function of number of samples and learning rates on the TED corpus



(a) Adam

(b) Adagrad

(c) RMSProp

(d) SGD

Figure 4: BLEU scores as a function of number of samples and learning rates on the NewsCommentary corpus

for figures (a) and (b), respectively); in the latter case, the optimiser is the one used to train the baselines, namely Adam, with the learning rate and moment inherited from the last baseline update.

Adam displayed a more erratic behaviour than the other optimisers, with sharp degradation after 10K updates when selecting a conservative learning rate, and after 100 with the more aggressive LR. Except for TED with the best@1 LR, where all optimisers started with maximal adaptation on the first update, Adagrad performed worse initially but eventually converged with SGD and RMSProp, while also obtaining the highest scores between 10 and 10K samples on TED with its most conservative LR, and between 10 and 1000 on NewsCommmentary. SGD and RMSProp behave similarly with a more stable behaviour, except on NewsCommentary where RMSProp performed markedly better with a conservative LR. It is also worth noting that the best LRs, whether aggressive or conservative, differ in most cases depending on the domain. Although this might be expected considering that the selected domains differ in terms of proximity to the generic data, as previously noted, these differences illustrate the delicate task of determining optimal online learning setups across the board.

The baseline evolved as expected, with lower scores than models benefitting from in-domain data. For batch fine-tuning, the evolution featured increasing scores as more data are available, eventually converging with the best optimiser variants. Overall, all variants of online learning tended towards degraded performance as the number of samples and subsequent updates increased, particularly with high learning rates. This is not unexpected given the overfitting associated with network adaptation over minimal samples, but both SGD and RMSProp appeared beneficial at least up to the 100K mark on the dynamic in-domain sets. We will examine the behaviour of all models on the static and out-of-domain test sets in Section 5.

4.2 Optimal Optimiser Setup

So far we have examined the behaviour of the different optimisers over time with the best LR at the two extremes, i.e. for 1 and 100K samples. To determine whether other learning rates might be optimal at other time steps, we computed BLEU scores on the dynamic in-domain set as a function of both learning rates and number of samples. Figure 3 and Figure 4 show the results on TED and NewsCommentary, respectively.

With Adam, other learning rates are more stable over time than the best performing one selected on the basis of a single (averaged) sample, with higher scores and less erratic behaviour, in particular on NewsCommentary. Nonetheless, even these more balanced learning rates achieve poorer scores than the other three optimisers overall, for both initial and final updates.

For Adagrad, other values than the ones based on the extremes performed better for some intermediate sample subsets on the TED dataset, but the more aggressive LR was optimal overall on NewsCommentary, achieving better scores than all other optimisers as the number of samples increased.

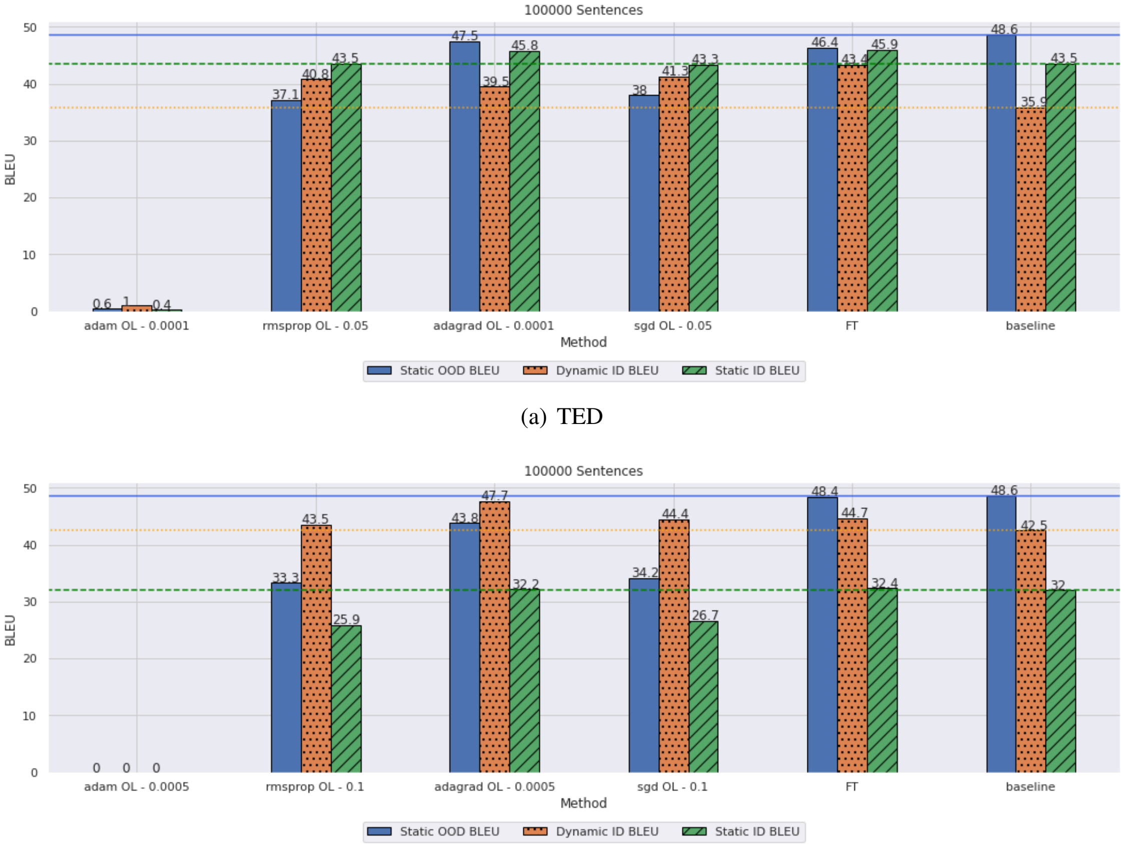
RMSProp also achieved an overall better distribution of scores when selecting the most aggressive LR on TED. On NewsCommentary, the selected best LR for the initial sample (0.1) was not the optimal choice, although it performed closely to the optimal 0.05. Note that both LRs achieve an identical score on the first sample, but on the average score obtained on the 100 randomly selected sentences used to to select the most aggressive LR, the previously selected LR of 0.1 was markedly better.

Similarly, for SGD on NewsCommentary, the best LR option over the averaged 100 unique samples (0.1) performed slightly worse overall than an LR of 0.05 as the number of samples increased; on TED, the most aggressive SGD LR performed better than the alternatives, except when the number of samples reached the 100K mark.

Although these results show that selecting an optimal learning rate for either optimiser is bound to be less than optimal at a given point in time, both SGD and RMSProp with an LR of 0.05 appear to be reasonable choices that provide overall benefits on the two selected datasets. Interestingly, this value differs from the optimal ones established for SGD in separate experiments by Turchi et al. (2017) and Peris and Casacuberta (2019) (see Section 3.3), showing that LR selection for online learning might be dependent on domains and datasets.

5 Model Stability Over Time

As described in the previous sections, online learning can support post-editing by adapting to user corrections incrementally. This is obtained via relatively aggressive learning rates that enable updates to be significant on the basis of unique training



(a) TED

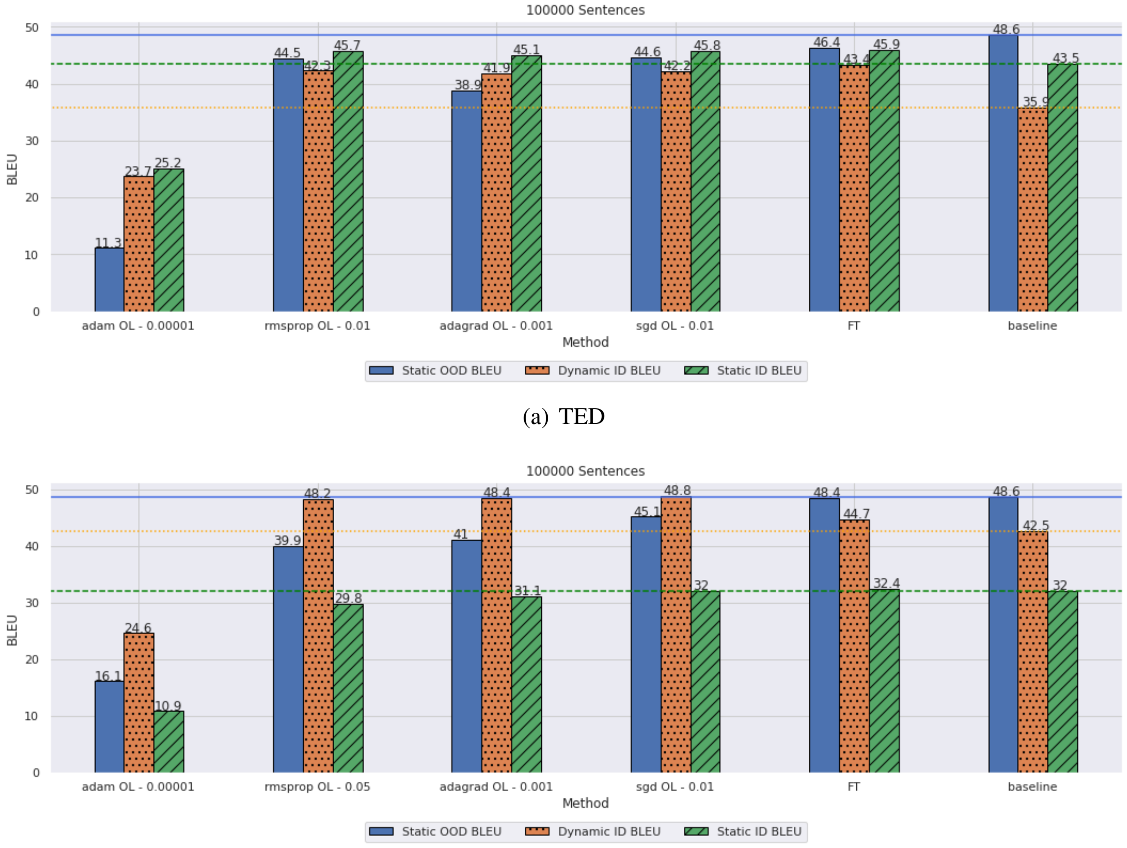
(b) NewsCommentary

Figure 5: Results on all datasets after 100K samples with best@1 learning rates

samples at each update point. This runs the risk of overfitting the model to the online training data, with incurred loss of quality on out-of-domain data, and crucially, on other portions of the in-domain data that did not undergo online training. To measure whether this is the case, we compared online learning with batch fine-tuning and the baselines, on all three types of datasets. For batch fine-tuning, the models were updated with batches that include all the online training pairs accumulated up to the current time step.

Figure 5 shows the results for TED and NewsCommentary at the 100K mark, when selecting the best learning rate for each optimiser based on the best BLEU scores for the initial updates, which, as a reminder, were computed over random samples of 100 sentences. On TED, batch fine-tuning outperformed all variants of online learning on the in-domain datasets, both static and dynamic, although only slightly over Adagrad on the static ID test set. On NewsCommentary, Adagrad outperformed all variants on the dynamic ID dataset, while also being only slightly under batch fine-tuning on the static ID test set. However, as described in previous sections, this optimiser also performed significantly worse than SGD and RMSProp for initial updates, thus being less beneficial for initial stages of online learning. When compared to the most efficient optimisers for earlier online learning, namely SGD and RMSProp, batch fine-tuning would be the favoured option when reaching at least 100K data points.

Figure 6 presents the results after the final update stage when taking the best learning rates at the 100K mark for all optimisers. In this scenario, on the TED datasets all optimisers except Adam feature results that are closer to those achieved via batch fine-tuning, although the latter obtained better results overall on both the dynamic and the static in-domain datasets. On NewsCommentary, SGD, RMSProp and Adagrad significantly outperformed batch fine-tuning on the dynamic data, while the latter performed slightly better on the static ID test set but with minor differences. Among optimisers in online learning scenarios, SGD would be favoured when selecting more conservative learning rates, although both RMSProp and SGD would



(a) TED

(b) NewsCommentary

Figure 6: Results on all datasets after 100K samples with best@100K learning rates

also be favoured over batch fine-tuning in such a scenario for NewsCommentary. However, a more conservative learning rate also reduces its benefits at earlier stages, and the single case where online learning optimisers outperform batch fine-tuning at later stages might not be relevant in actual usage.

In terms of out-of-domain data, batch fine-tuning performed better in all but one case, namely TED with the best learning rate for initial updates, where Adagrad performed better. Batch fine-tuning performed similarly to the baseline on NewsCommentary, which may be attributed to the relative proximity of NewsCommentary data to the generic training data, and conversely to the higher data difference between TED and the datasets that compose the generic training sets.

6 Conclusions

In this paper, we explored the behaviour of online learning for Neural Machine Translation over time, examining the results obtained with four different optimisers as the number of samples increases and evaluating translation model evolution after repeated network updates with different types of learning rates, from most aggressive to most conservative. We also compared online learning with batch fine-tuning on dynamic and static datasets, as well as out-of-domain test sets, to measure overall model stability.

On the two domains we explored, based on TED and NewsCommentary data, there does not appear to be an optimal configuration, where online learning would be optimal in both the short and long term. SGD and RMSProp both feature a learning rate value which provides early benefits of online learning with relatively minor degradation over time, and might be viewed as the most balanced configuration in our experiments.

However, at least in the domains we explored, batch fine-tuning was shown to be preferable at later stages in terms of model stability across dynamic, static and out-of-domain datasets. For practical adaptive NMT, it might thus be preferable to combine online learning, over limited time steps, with periodic batch fine-tuning over previous model checkpoints on the data accumulated over time.

Acknowledgments

This work was partially supported by the Department of Economic Development of the Basque Government (Spri), as part of project ITAI (ZL-2021/00038). We would like to thank the anonymous RANLP reviewers for their insightful comments and useful feedback.

References

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

Luisa Bentivogli, Nicola Bertoldi, Mauro Cettolo, Marcello Federico, Matteo Negri, and Marco Turchi. 2015. On the evaluation of adaptive machine translation for human post-editing. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(2):388–399.

Nicola Bertoldi, Patrick Simianer, Mauro Cettolo, Katharina Wäschle, Marcello Federico, and Stefan Riezler. 2014. Online adaptation to post-edits for phrase-based statistical machine translation. *Machine Translation*, 28(3-4):309–339.

Peter F Brown, John Cocke, Stephen A Della Pietra, Vincent J Della Pietra, Fredrick Jelinek, John D Lafferty, Robert L Mercer, and Paul S Roossin. 1990. A statistical approach to machine translation. *Computational linguistics*, 16(2):79–85.

Mauro Cettolo, Christian Girardi, and Marcello Federico. 2012. WIT3: Web inventory of transcribed and translated talks. In *Proceedings of the 16th Annual conference of the European Association for Machine Translation*, pages 261–268, Trento, Italy.

Miguel Domingo, Mercedes García-Martínez, Álvaro Peris, Alexandre Helle, Amando Estela, Laurent Bié, Francisco Casacuberta, and Manuel Herranz. 2019. Incremental adaptation of NMT for professional post-editors: A user study. In *Proceedings of Machine Translation Summit XVII Volume 2: Translator, Project and User Tracks*, pages 219–227, Dublin, Ireland.

Miguel Domingo, Mercedes García-Martínez, Álvaro Peris, Alexandre Helle, Amando Estela, Laurent Bié, Francisco Casacuberta, and Manuel Herranz. 2020. A user study of the incremental learning in NMT. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 319–328.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7).

Andreas Eisele and Yu Chen. 2010. MultiUN: A multilingual corpus from United Nation documents. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation*, pages 2868–2872.

Ahmed El-Kishky, Vishrav Chaudhary, Francisco Guzmán, and Philipp Koehn. 2020. CCAligned: A massive collection of cross-lingual web-document pairs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 5960–5969.

Daniel Hardt and Jakob Elming. 2010. Incremental re-training for post-editing SMT. In *Proceedings of the Ninth Conference of the Association for Machine Translation in the Americas*, pages 217–237, Denver, Colorado, USA.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia.

Sariya Karimova, Patrick Simianer, and Stefan Riezler. 2018. A user-study on online adaptation of neural machine translation to human post-edits. *Machine Translation*, 32(4):309–324.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations*, San Diego, CA, USA.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of the 10th Machine Translation Summit*, pages 79–86, Phuket, Thailand.

Philipp Koehn. 2010. *Statistical machine translation*. Cambridge University Press.

Abby Levenberg, Chris Callison-Burch, and Miles Osborne. 2010. Stream-based translation models for statistical machine translation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 394–402, Los Angeles, California.

Pierre Lison and Jörg Tiedemann. 2016. OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation*, pages 923–929, Portoroz, Slovenia.

Daniel Ortiz-Martínez. 2016. Online learning for statistical machine translation. *Computational Linguistics*, 42(1):121–161.

Daniel Ortiz-Martínez, Ismael García-Varea, and Francisco Casacuberta. 2010. Online learning for interactive statistical machine translation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 546–554.

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

Álvaro Peris. 2020. *Interactivity, Adaptation and Multimodality in Neural Sequence-to-sequence Learning*. Ph.D. thesis, Universitat Politècnica de València.

Álvaro Peris and Francisco Casacuberta. 2019. Online learning for effort reduction in interactive neural machine translation. *Computer Speech & Language*, 58:98–126.

Álvaro Peris, Miguel Domingo, and Francisco Casacuberta. 2017. Interactive neural machine translation. *Computer Speech & Language*, 45:201–220.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium.

Herbert Robbins and Sutton Monro. 1951. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407.

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.

Patrick Simianer, Joern Wuebker, and John DeNero. 2019. Measuring immediate adaptation performance 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 2038–2046, Minneapolis, Minnesota, USA.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of Association for Machine Translation in the Americas*, pages 223–231, Cambridge, Massachussets, USA.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the 8th Language Resources and Evaluation Conference*, pages 2214–2218, Istanbul, Turkey.

Marco Turchi, Matteo Negri, Amin Farajian, and Marcello Federico. 2017. Continuous learning from human post-edits for neural machine translation. *The Prague Bulletin of Mathematical Linguistics*, 108:233–244.

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 *Advances in Neural Information Processing Systems*, pages 6000–6010.

Matthew D Zeiler. 2012. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.