# Knowledge Distillation for Sustainable Neural Machine Translation

**Wandri Jooste** wandri.jooste@adaptcentre.ie
**Andy Way** andy.way@adaptcentre.ie
ADAPT Centre, Dublin City University, Dublin, Ireland

**Rejwanul Haque** rejwanul.haque@ncirl.ie
National College of Ireland, Dublin, Ireland

**Riccardo Superbo** riccardos@kantanai.io
KantanAI, DCU Alpha, Dublin, Ireland

**Abstract**

Knowledge distillation (KD) can be used to reduce model size and training time, without significant loss in performance; in some instances, it even leads to performance gains. These smaller models, also known as student models, are much more efficient in terms of time and energy costs, and they emit far less $CO_2$. However, the process of distilling knowledge requires translation of sizeable data sets, and the translation is usually performed using large cumbersome models, also known as teacher models. The intuition is to produce smaller student models that can mimic well the large teacher models which are usually good in quality. Nevertheless, producing translations of sizeable data sets by large-scale teacher models for KD is expensive in terms of both time and cost, which is a significant concern for translation service providers (TSPs). On top of that, the use of cumbersome models for translating large-scale data sets can be the cause of higher carbon footprints. In this work, we tested different variants of a teacher model in order to produce translations of a large-scale data set, tracked the power consumption of the graphic processing units (GPUs) used during translation, recorded overall translation time, estimated translation cost, and measured the accuracy of the student models. The findings of our investigation demonstrate to the translation industry a cost-effective, high-quality alternative to the standard KD training methods which are highly time-consuming and computationally expensive. More importantly still, we show that our proposed solutions are the most environmentally friendly training methods to distil knowledge from a teacher to a student model, while maintaining an insignificant drop in accuracy.

## 1 Introduction

Deep neural networks (DNNs) underpin state-of-the-art applications of artificial intelligence in almost all fields, such as image (Voulodimos et al., 2018), speech (Park et al., 2019) and natural language processing (NLP) (Wolf et al., 2019). However, DNN architectures (LeCun et al., 2015) are often data-, compute-, space-, power- and energy-hungry, typically requiring powerful GPUs or large-scale clusters to train and deploy, which has been viewed as a "non-green" technology (Strubell et al., 2019). Furthermore, often the best-performing models are ensembles of hundreds or thousands of base-level models, which require large amounts of space and time for storage and execution (Singh et al., 2016; Wen et al., 2017; Fedus et al., 2021).

Neural MT (NMT) (Cho et al., 2014; Sutskever et al., 2014; Bahdanau et al., 2014) systems have greatly improved MT compared to statistical MT (SMT) (Koehn, 2009) systems. These state-of-the-art NMT systems, however, require much more computing power and data than SMT systems (Östling and Tiedemann, 2017; Dowling et al., 2020), and if the effects of climate change are to be controlled they are unsustainable in the long run. These systems are also largely unsuitable for training engines for low-resource languages and scenarios, where the data simply does not exist in the amounts required for high quality results to be achieved. To some extent, model compression, more specifically knowledge distillation techniques, can remedy this.

Knowledge distillation (Buciluǎ et al., 2006) can be used to transfer the knowledge from a teacher network (a large model) to a student network (a smaller model). This is generally done by using a smaller, fast model to approximate the function learned by a much larger and slower model with better performance (Buciluǎ et al., 2006; Hinton et al., 2015).

The methods described by Buciluǎ et al. (2006) and Hinton et al. (2015) can be used for word-level knowledge distillation, since NMT models make use of multi-class prediction at the word-level. These models, however, need to predict complete sequences that are dependent on previous predictions as well.

Kim and Rush (2016) propose sequence-level KD wherein a new training set is generated by translating a data set with the teacher model using beam search. The newly generated training set is then used to train a smaller student model. The data sets often contain millions of sentences and thus translating the training set by a large-scale teacher model for KD training (Bapna et al., 2022) can be a cumbersome task and computationally expensive process. Thus, KD training in MT is responsible for a considerable amount of $CO_2$ emissions. This is a concerning matter for the environment. Besides, this is also a concern for TSPs who want to increase their margins in translation productivity by offering translations by smaller student models to their clients. More specifically, the standard and computationally expensive KD training process can negatively impact the translation productivity gain in industry.

The standard KD training methods use large-scale teacher models. We tested a number of variants of a teacher model for translating the source sentences of our training data. For example, we investigated the effects of changing the beam size and using quantisation (Polino et al., 2018; Prato et al., 2020) while translating the training data. More specifically, we tested the approaches of both Bogoychev et al. (2020) and Behnke et al. (2021), who focused on quantisation during inference to reduce the size of their student models and increase the speed at which these models translate sentences. The quality of translations by a quantised teacher model would naturally be worse than that of the translations by non-quantized teacher model. The same is true when one uses a very small batch size (e.g. 1) at decoding. As a result, the translations by these fast decoders would naturally impact the quality the resultant student model. In other words, you are likely to obtain a worse student model when you use a quantised teacher model to distil knowledge. Our investigation focused on examining the magnitude of quality drop of the student models when using the different variants of the teacher models for KD, and in return how faster, cheaper and environmental friendly the KD training process would be.

We considered English-to-German for our investigation, and recorded a number of parameters (i.e. $CO_2$ emissions, translation time, accuracy) including power consumption of the GPUs used for translation. We empirically demonstrated that our proposed KD training methods are computationally less expensive in comparison to the standard KD methods, and more importantly, they do not deteriorate the accuracy of the student models much. As far as we are aware, related work on model efficiency and knowledge distillation (Kim and Rush, 2016; Bhandare et al., 2019; Bogoychev et al., 2020; Prato et al., 2020; Heafield et al., 2021) focus on performance during inference whereas in this paper we focus on the effects of quantisation for

KD training, in order to make the process as a whole more efficient rather than just reducing the size and increasing the speed of the student models.

## 2 Experimental Setup

In this section we describe the various aspects of our experiments. We first discuss the data and how it was preprocessed and then move on to describe how our MT systems are trained. Lastly, we describe how we evaluate the quality of these MT systems.

### 2.1 Data

We use the Europarl [1] (Koehn, 2005) corpus with parallel sentences in German and English for the language direction German to English. The corpus is randomly divided into three subsets, namely the training set, validation set and test set. The training set consists of roughly two million sentences and the validation and test sets of 3,000 sentences, respectively.

The Moses toolkit (Koehn et al., 2007) was used to tokenize and clean the three datasets by removing all sentences with a length greater than 100. The toolkit was also used to decase all sentences before training and after training, we used a pretrained truecaser to recase all translated sentences. Furthermore, SubwordNMT[2] (Sennrich et al., 2016) was used to segment the sentences in the corpus into subword units. More specifically, the Byte Pair Encoding (BPE) vocabularies were set to $32k$ words. Jooste et al. (2022) experimented with limiting the vocabulary sizes during training and the models with smaller vocabularies (16k and 8k) trained faster on average, albeit with lower quality in terms of runtime performance. Since this work aims at investigating how sustainable the KD training process would be, we kept this hyperparametr constant across our all experiments.

### 2.2 MT Systems

We use the MarianNMT[3] toolkit (Junczys-Dowmunt et al., 2018) and Transformer (Vaswani et al., 2017) architecture to train the models for our experiments. All models were trained for a maximum of 20 epochs, since that was the lowest number of epochs needed to finish training for one of our models. We used NVIDIA RTX 2080ti GPUs to train our models as well as during decoding when distilling knowledge.

We used the same setup for training as described in the work of Jooste et al. (2022), who investigated how sustainable today's neural MT systems are on industrial setups. The student models and baseline models have an encoder and decoder depth of 3, whereas the teacher models have an encoder and decoder depth of 6. Other than the difference in encoder and decoder layers, the training parameters are the same but the student models are trained on the knowledge-distilled data set instead of the original training set.

| Setup | Beam Size | Mini/Maxi Batch | Quantisation |
|---|---|---|---|
| Original | 12 | 10/100 | fp32 |
| Beam | 1 | 10/100 | fp32 |
| Quantisation | 12 | 10/100 | fp16 |
| Combined | 1 | 128/256 | fp16 |

Table 1: Comparison of decoding experiments.

Table 1 shows the various setups we used for decoding. Originally we used a beam size of 12 without quantisation when distilling knowledge. We then investigated the effects of changing

[1] https://opus.nlpl.eu/Europarl-v3.php
[2] https://github.com/rsennrich/subword-nmt
[3] https://github.com/marian-nmt/marian

the beam size to one and using quantisation to use 16 bit floating point numbers (fp16) rather than 32 bit floating point numbers (fp32). Since using smaller floating point numbers requires less memory, we are also able to use a better combination of the mini- and maxi-batch sizes.

In Jooste et al. (2022), they showed that the student models outperform the teacher models when training them on the original training set combined with the knowledge distilled training set (KD-set). In this work, however, we will only focus on the effects when using only the KD-set for training student models.

### 2.3 Evaluation

The accuracy of all our models was measured with three automatic evaluation metrics, namely BLEU (Papineni et al., 2002), TER (Snover et al., 2006) and chrF[4] (Popović, 2015), using the MultEval toolkit[5] (Clark et al., 2011). We decided to use three metrics instead of one to better represent the accuracy of the models.

In order to gain more insight into the efficiency of distilling knowledge from our teacher models, we tracked the power usage during inference. The NVIDIA System Management Interface (nvidia-smi) was used to report the power draw of the GPUs being used, which was then used to approximate the $CO_2$ emitted during inference.

$CO_2$ can be computed using the Power Usage Effectiveness (PUE) of the data centre, kilowatt per hour (kWh) and the $CO_2$ intensity ($I^{CO_2}$) as in equation (1) (Strubell et al., 2019; Henderson et al., 2020):

$$CO_2Emissions = \frac{PUE * kWh * I^{CO_2}}{1000} \quad (1)$$

Sherionov and Vanmassenhove (2022) pointed out that the values of PUE and $I^{CO_2}$ are dependent on various factors and also constantly changing. In this paper we will use the same values as reported by Sherionov and Vanmassenhove (2022) ($PUE = 1.59$ and $I^{CO_2} = 229.8718 \pm 77.4026$). The kWh is calculated by dividing the total kW measured per second by 360 in order to get the kW per hour rather than seconds.

The translation time was also tracked in order to estimate the cost of using the various decoding methods. To estimate the cost we used the AWS Pricing Calculator[6] for EC2 instances, on demand and located in Ireland. The cost of an instance varies on the number of GPUs needed and is then multiplied by the number of hours it is used for. For each scenario the cost of adding 30 GiB of memory is a flat of of 3.30 US Dollars (USD) per month and since it is constant, it was left out of our estimated cost calculations.

## 3 Results and Discussion

Table 2 compares different setups that we described in Section 2.2 in terms of the translation time, power draw, approximate $CO_2$ emissions and estimated cost in US Dollars (USD). As can be seen from the table, the combined methods ('Combined' in Table 2) are the most efficient way of distilling knowledge. We illustrate the performance of the student models that correspond to the KD training setups of Table 2 in Table 3. We can clearly see from both tables that the difference in translation time has been improved by 7 hours on average, while the quality of the student models drops by less than 1 BLEU point. The same trends are observed with the other MT evaluation metrics too. Such a small drop in quality is unlikely to be spotted by a human, even an expert translator.

[4] https://github.com/m-popovic/chrF
[5] https://github.com/jhclark/multeval
[6] https://calculator.aws/#/createCalculator/EC2

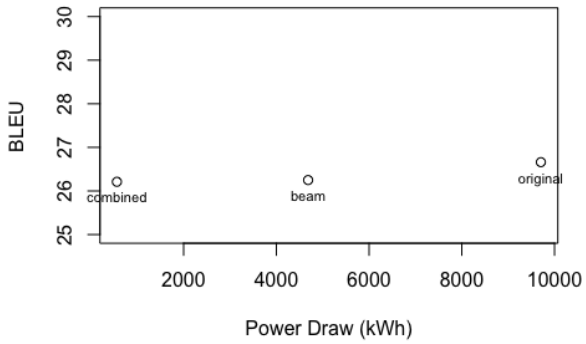


Figure 1: BLEU score of student models compared to the corresponding power draw of the distillation methods mentioned when using 1 GPU

In Figure 1 we show the BLEU scores of the student models and the power draw (required to create the training sets) of the various distillation methods proposed. In the figure we only show the distillation method when using 1 GPU since that is the most power-efficient option in most cases. When taking the power draw, and especially the $CO_2$ missions shown in Table 2, into account, the loss in BLEU score is very insignificant if we assume that all our AI models need to become much more sustainable.

| Setup | # of GPUs | Time | Power (kW) | $CO_2$ (kg) | Cost in (USD) |
|---|---|---|---|---|---|
| Original | 1 | 13:35:28 | 9,705.34 | $9.85 \pm 3.32$ | 8.21 |
| | 2 | 10:34:02 | 11,531.92 | $11.71 \pm 3.94$ | 20.46 |
| | 4 | 11:21:34 | 10,467.93 | $10.63 \pm 3.58$ | 31.90 |
| Beam | 1 | 11:07:05 | 4,685.21 | $4.76 \pm 1.60$ | 6.72 |
| | 2 | 07:26:50 | 4,337.43 | $4.40 \pm 1.48$ | 14.42 |
| | 4 | 08:31:21 | 5,893.63 | $5.98 \pm 2.01$ | 23.93 |
| Quantisation | 1 | 11:10:30 | 6,146.91 | $6.24 \pm 2.10$ | 6.75 |
| | 2 | 06:59:42 | 8,207.46 | $8.33 \pm 2.80$ | 13.54 |
| | 4 | 10:31:47 | 8,779.62 | $8.91 \pm 3.00$ | 29.57 |
| Combined | 1 | 00:47:36 | 560.59 | $0.57 \pm 0.19$ | 0.48 |
| | 2 | 00:30:26 | 618.17 | $0.63 \pm 0.21$ | 0.98 |
| | 4 | 00:42:58 | 646.55 | $0.66 \pm 0.22$ | 2.01 |

Table 2: Comparison of the translation time, power draw and approximate $CO_2$ emissions of various decoding setups.

When using only a smaller beam size or quantisation, respectively, the translation times are only marginally improved, i.e. two or three hours compared to six when using the combined method. Interestingly, quantisation alone leads to a minimal speedup, and it is experimenting with mini- and maxi-batch size that has the most significant impact. When trying to use the same batch sizes without quantisation however, the GPUs would run out of memory after translating only a few sentences. We therefore draw attention to utilising the GPU specifications when considering mini- and maxi-batch size when using quantisation for speeding up the distillation

| Setup | # of GPUs | Training time | BLEU | TER | chrF |
|---|---|---|---|---|---|
| Original | 1 | 07:17:39 | 26.66 | 49.5 | 59.37 |
| | 2 | 04:22:14 | 26.49 | 49.3 | 59.51 |
| | 4 | 04:05:28 | 26.22 | 50.1 | 59.11 |
| Beam | 1 | 06:42:33 | 26.25 | 48.6 | 60.51 |
| | 2 | 04:23:33 | 26.38 | 48.5 | 60.52 |
| | 4 | 04:38:00 | 26.49 | 50.3 | 59.44 |
| Combined | 1 | 06:18:08 | 26.21 | 48.7 | 60.32 |
| | 2 | 04:25:53 | 26.53 | 48.7 | 60.49 |
| | 4 | 04:38:00 | 26.21 | 49.5 | 60.02 |

Table 3: Comparison of the performance of the student models using the various decoding setups.

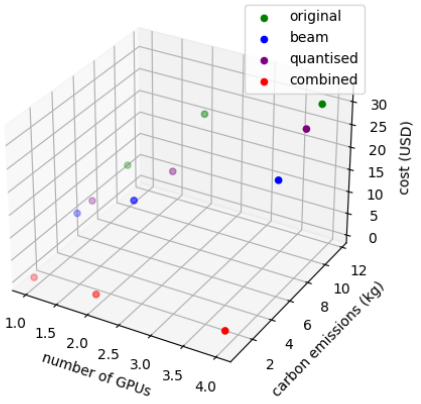


Figure 2: The $CO_2$ emissions and cost in USD of translation when using 1,2 or 4 GPUs during the distillation process.

process.

Taking the power draw and $CO_2$ emissions into account, Table 2 shows that using a beam size of 1 decreases the power draw by almost half and in turn the carbon emissions, even though the translation time is only marginally shorter. When using quantisation without optimal batch sizes however, the power draw is more than that of the beam setup while the translation time remains similar. Quantisation is therefore only an optimisation method for inference when the correct batch sizes are taking into account, depending on the GPU specifications.

The effects that these methods of speeding up decoding have on the accuracy of the student models are shown in Table 3. When taking the carbon emissions, decoding time and accuracy into account, it is clear that using the combined method is the most efficient setup to use when distilling knowledge from a teacher to a student model, since the accuracy decreased by less than 1 BLEU point while translation time decreased by 10 or more hours.

When considering the cost for TSPs, as seen in Table 2, using only 1 GPU is the most cost-effective for all methods of decoding. Interestingly, while using 2 GPUs speeds up the translation time for all methods, the estimated cost is double that of using only 1 GPU.

Figure 2 provides a summary of the most notable results in Table 2. It is clear that as the number of GPUs in the translation process increases, the $CO_2$ emissions go up in most cases and the cost of using an AWS EC2 instance rises in all scenarios. The increase in $CO_2$ emissions and cost is not linear as the number of GPUs is increased, nor is the translation time shown in Table 3 due to the performance overheads of using multiple GPUs (Xu et al., 2021).

## 4 Conclusions and Future Work

We described various methods in which the distillation of knowledge can be made more efficient and in turn more sustainable. Most significantly, the impact of batch sizes when using quantisation and a smaller beam size result in a less than 1 BLEU point drop in accuracy, while at the same time reducing decoding time hby at least 10 hours compared to the original method.

In terms of efficiency, the combined setup is found to be the best method for distilling knowledge from the teacher to student models. The $CO_2$ emissions of our combined setup is on average 10kg less than the original setup while accuracy decreases only slightly. The environmental impact of distilling knowledge from a teacher model to a student model are encouraging, and we contend that more importance should be given to this issue since inefficient methods emit on average 10 times more $CO_2$ than optimised methods, yet the cost in accuracy of the student models is minimal. Taking only the end result (student model) into account is not sustainable and more consideration needs to be put into the whole process.

We have shown that during the process of distilling knowledge from a teacher model to a student model, using just 2 GPUs can result in the fastest translation time while using 1 GPU is the most cost-effective and in most cases the most environmentally friendly as well. Interestingly, from our results it is clear that when taking $CO_2$ emissions and cost into account, using 4 GPUs is much less efficient compared to using only 1 GPU.

In future we will investigate the efficiency of using CPUs during the distillation process as well as during inference when the student models are deployed. We also aim to develop a composite metric that takes carbon emissions, accuracy and access to resources into account in order to rate the performance of MT Systems. Furthermore, we aim to investigate to what extent these decoding methods work on different language pairs, especially their effect on low-resource languages where access to data is considerably more problematic.

## 5 Acknowledgements

This publication has emanated from research conducted with the financial support of Science Foundation Ireland under Grant number 18/CRT/6183. The ADAPT Centre for Digital Content Technology (https://www.adaptcentre.ie/) is funded under the Science Foundation Ireland Research Centres Programme (Grant number 13/RC/2106) and is co-funded under the European Regional Development Fund. We also wish to thank KantanAI (https://www.kantanai.io/) for hosting the first-named author on her ML-Labs (https://www.ml-labs.ie/) placement, and for permission to use the statistics quoted in this paper.

## References

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate.

Bapna, A., Caswell, I., Kreutzer, J., Firat, O., van Esch, D., Siddhant, A., Niu, M., Baljekar, P., Garcia, X., Macherey, W., Breiner, T., Axelrod, V., Riesa, J., Cao, Y., Chen, M. X., Macherey, K., Krikun, M., Wang, P., Gutkin, A., Shah, A., Huang, Y., Chen, Z., Wu, Y., and Hughes, M. (2022). Building machine translation systems for the next thousand languages.

Behnke, M., Bogoychev, N., Aji, A. F., Heafield, K., Nail, G., Zhu, Q., Tchistiakova, S., van der Linde, J., Chen, P., Kashyap, S., and Grundkiewicz, R. (2021). Efficient machine translation with model pruning and quantization. In *Proceedings of the Sixth Conference on Machine Translation*, pages 775–780, Online. Association for Computational Linguistics.

Bhandare, A., Sripathi, V., Karkada, D., Menon, V., Choi, S., Datta, K., and Saletore, V. (2019). Efficient 8-Bit Quantization of Transformer Neural Machine Language Translation Model. *arXiv:1906.00532 [cs]*. arXiv: 1906.00532.

Bogoychev, N., Grundkiewicz, R., Aji, A. F., Behnke, M., Heafield, K., Kashyap, S., Farsarakis, E.-I., and Chudyk, M. (2020). Edinburgh's submissions to the 2020 machine translation efficiency task. In *Proceedings of the Fourth Workshop on Neural Generation and Translation*, pages 218–224, Online. Association for Computational Linguistics.

Buciluǎ, C., Caruana, R., and Niculescu-Mizil, A. (2006). Model compression. In *KDD '06: Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 535–541, Philadelphia, PA, USA. ACM.

Cho, K., Van Merriënboer, B., Bahdanau, D., and Bengio, Y. (2014). On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*.

Clark, J. H., Dyer, C., Lavie, A., and Smith, N. A. (2011). Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 176–181, Portland, Oregon, USA. ACL.

Dowling, M., Castilho, S., Moorkens, J., Lynn, T., and Way, A. (2020). A human evaluation of English-Irish statistical and neural machine translation. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 431–440, Lisboa, Portugal. European Association for Machine Translation.

Fedus, W., Zoph, B., and Shazeer, N. (2021). Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity.

Heafield, K., Zhu, Q., and Grundkiewicz, R. (2021). Findings of the WMT 2021 shared task on efficient translation. In *Proceedings of the Sixth Conference on Machine Translation*, pages 639–651, Online. Association for Computational Linguistics.

Henderson, P., Hu, J., Romoff, J., Brunskill, E., Jurafsky, D., and Pineau, J. (2020). Towards the systematic reporting of the energy and carbon footprints of machine learning. *Journal of Machine Learning Research*, 248:1–43.

Hinton, G., Vinyals, O., and Dean, J. (2015). Distilling the knowledge in a neural network.

Jooste, W., Haque, R., and Way, A. (2022). Knowledge distillation: A method for making neural machine translation more efficient. *Information*, 13(2).

Junczys-Dowmunt, M., Grundkiewicz, R., Dwojak, T., Hoang, H., Heafield, K., Neckermann, T., Seide, F., Germann, U., Aji, A. F., Bogoychev, N., Martins, A. F. T., and Birch, A. (2018). Marian: Fast Neural Machine Translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. ACL.

Kim, Y. and Rush, A. M. (2016). Sequence-Level Knowledge Distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, Austin, Texas. ACL.

Koehn, P. (2005). Europarl: A parallel corpus for statistical machine translation. In *Proceedings of Machine Translation Summit X: Papers*, pages 79–86, Phuket, Thailand.

Koehn, P. (2009). *Statistical machine translation*. Cambridge University Press.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. ACL.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.

Östling, R. and Tiedemann, J. (2017). Neural machine translation for low-resource languages. *arXiv preprint arXiv:1708.05729*.

Papineni, K., Roukos, S., Ward, T., and Zhu, W. J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, page 311–318, Philadelphia, Pennsylvania, USA. ACL.

Park, D. S., Chan, W., Zhang, Y., Chiu, C.-C., Zoph, B., Cubuk, E. D., and Le, Q. V. (2019). Specaugment: A simple data augmentation method for automatic speech recognition. *arXiv preprint arXiv:1904.08779*.

Polino, A., Pascanu, R., and Alistarh, D. (2018). Model compression via distillation and quantization. Number: arXiv:1802.05668 arXiv:1802.05668 [cs].

Popović, M. (2015). chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. ACL.

Prato, G., Charlaix, E., and Rezagholizadeh, M. (2020). Fully Quantized Transformer for Machine Translation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1–14, Online. Association for Computational Linguistics.

Sennrich, R., Haddow, B., and Birch, A. (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. ACL.

Sherionov, D. and Vanmassenhove, E. (2022). The ecological footprint of neural machine translation systems. *arXiv preprint arXiv:2202.02170*.

Singh, S., Hoiem, D., and Forsyth, D. (2016). Swapout: Learning an ensemble of deep architectures. *arXiv preprint arXiv:1605.06465*.

Snover, M., Dorr, B., Schwartz, R., Micciulla, L., and Makhoul, J. (2006). A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*, page 223–231, Cambridge, Massachusetts, USA. AMTA.

Strubell, E., Ganesh, A., and McCallum, A. (2019). Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650, Florence, Italy. ACL.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. *arXiv preprint arXiv:1409.3215*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, pages 5998–6008, Long Beach, CA, USA. NIPS.

Voulodimos, A., Doulamis, N., Doulamis, A., and Protopapadakis, E. (2018). Deep learning for computer vision: A brief review. *Computational intelligence and neuroscience*, 2018.

Wen, G., Hou, Z., Li, H., Li, D., Jiang, L., and Xun, E. (2017). Ensemble of deep neural networks with probability-based fusion for facial expression recognition. *Cognitive Computation*, 9(5):597–610.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., et al. (2019). Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Xu, W., Zhang, Y., and Tang, X. (2021). Parallelizing dnn training on gpus: Challenges and opportunities. In *Companion Proceedings of the Web Conference 2021*, WWW '21, page 174–178, New York, NY, USA. Association for Computing Machinery.