Few-Shot Regularization to Tackle Catastrophic Forgetting in Multilingual Machine Translation

Salvador Carrión salcarpo@prhlt.upv.es
Francisco Casacuberta fcn@prhlt.upv.es
PRHLT Research Center, Universitat Politècnica de València

Abstract

Increasing the number of tasks supported by a machine learning model without forgetting previously learned tasks is the goal of any lifelong learning system. In this work, we study how to mitigate the effects of the catastrophic forgetting problem to sequentially train a multilingual neural machine translation model using minimal past information. First, we describe the catastrophic forgetting phenomenon as a function of the number of tasks learned (language pairs) and the ratios of past data used during the learning of the new task. Next, we explore the importance of applying oversampling strategies for scenarios where only minimal amounts of past data are available. Finally, we derive a new loss function that minimizes the forgetting of previously learned tasks by actively re-weighting past samples and penalizing weights that deviate too much from the original model. Our work suggests that by using minimal amounts of past data and a simple regularization function, we can significantly mitigate the effects of the catastrophic forgetting phenomenon without increasing the computational costs.

1 Introduction

The catastrophic forgetting is the phenomenon whereby a neural network forgets previously learned information after learning new one (McCloskey and Cohen, 1989).

Given the ubiquity nature of machine learning models in our lives, tackling the catastrophic forgetting phenomenon is a problem of particular interest for the industry as machine learning models tend to lose performance over time due to the changing nature of our world. To counteract this problem, researchers and engineers must periodically re-train these models. However, despite the inefficiency of re-training a large model from scratch and the carbon footprint that this practice entails in the long run, previous training data is not always available due to privacy issues, licensing, data losses, or simply, because the training data is not available.

This problem is incredibly challenging since any learning system with a limited amount of memory will, at some point, have to forget past information in order to keep learning new information (Carpenter and Grossberg, 1987). Fortunately, we can develop mechanisms so that our machine learning models can selectively forget as little information as possible by penalizing changes in weights that deviate too much from a reference model (Li and Hoiem, 2016; Kirkpatrick et al., 2016), designing dynamic architectures that grow linearly with the number of tasks (Rusu et al., 2016; Draelos et al., 2016), or using Complementary Learning Systems (CLS) that, inspired by how the human brain work, generate synthetic data to control the forgetting (Kemker and Kanan, 2017).

From a practical point of view, these approaches tend to be quite hard to implement and often are very computationally intensive. In addition, most of these strategies are not specifically

designed for natural language tasks, making their implementation even more difficult. Therefore, we decided to tackle the catastrophic forgetting problem in machine translation, framed as a sequential learning problem for a multilingual machine translation system, where each new task is a different language pair (English-Spanish, English-French, English-German, and English-Czech).

The contributions of this work are the following:

- First, we describe the catastrophic forgetting phenomenon in machine translation as a function of the tasks learned (language pairs) and the ratios of past data used during the learning of the new task, and show that even with minimal amounts of past data we can significantly mitigate these effects.
- Next, we explore the effectiveness of oversampling strategies, where we show that they are particularly useful for scenarios where only minimal amounts of past data are available.
- Finally, we derive a new loss function that minimizes the forgetting of past tasks using a few-shot strategy based on actively re-weighting past tasks and penalizing weights that deviate too much from the original model.

2 Related Work

The *Catastrophic Forgetting* (CF) phenomenon has been widely studied since it was introduced for the first time by McCloskey and Cohen (1989). However, despite the numerous works that have delved into the root causes that produce it (Carpenter and Grossberg, 1987), these findings could be reduced to the stability-plasticity dilemma, whereby there is a trade-off between the ability of a model to preserve past knowledge (stability) and the ability to learn new information effectively (plasticity).

Given this dilemma, most approaches are based on adjusting the network weights during training to control the forgetting of the model, expanding the model's capacity to support new tasks, or using some refreshing mechanism to remember past tasks.

For example, Li and Hoiem (2016) presented a model with shared parameters across tasks and task-specific parameters; Kirkpatrick et al. (2016) identified which weights were important for the past tasks so that they could penalize the updates on those weights; Jung et al. (2016) penalized changes in the final hidden layer; Zenke et al. (2017) introduced the concept of intelligent synapses that accumulate task-relevant information; Hu et al. (2019) trained a model with a set of parameters that was shared by all tasks and the second set of parameters that were dynamically generated to adapt the model to each new task. However, despite the number of works, these strategies are constrained by the model's capacity (Kaplan et al., 2020).

To deal with this issue, many researchers decided to focus their efforts on linearly expanding the model's capacity as the number of tasks grows. Accordingly, (Rusu et al., 2016) retained a pool of pre-trained models throughout training to learn lateral connections for the new task; (Draelos et al., 2016), which was inspired by the neurogenesis in the hippocampus of the brain decided to add new neurons to deep layers so that novel information could be acquired more efficiently; and (Lee et al., 2017a) introduced an architecture that dynamically controls the network capacity.

Similarly, other researchers have addressed this problem by using data from past tasks during the training of new tasks, such as Lopez-Paz and Ranzato (2017), who proposed a model that alleviates the catastrophic forgetting problem by storing a subset of the observed examples from an old task (episodic memory), and Shin et al. (2017), who instead of storing actual training data from past tasks, trained a deep generative model that replayed past data (synthetically) during training to prevent forgetting.

In addition to these works, there are others worth to mention due to their results and original approaches, such as iCaRL (Rebuffi et al., 2016), PathNet (Fernando et al., 2017), FearNet (Kemker and Kanan, 2017), IMM (Lee et al., 2017b) or MAS (Aljundi et al., 2017).

Nonetheless, despite the progress made on lifelong learning strategies and the recent breakthroughs in the natural language field (Sutskever et al., 2014; Sennrich et al., 2016; Vaswani et al., 2017; Zhang et al., 2019), the catastrophic forgetting problem has not been so widely studied in the field of machine translation. Along these lines, Xu et al. (2018) proposed a meta-learning method that exploits knowledge from past domains to generate improved embeddings for a new domain; Qi et al. (2018) showed that pre-trained embeddings could be effective in low-resource scenarios; Liu et al. (2019) learned corpus-dependent features by sequentially updating sentence encoders (previously initialized with the help of corpus-independent features) using Boolean operations of conceptor matrices; Sato et al. (2020) presented a method to adapt the embeddings between domains by projecting the target embeddings into the source space, and then fine-tuning them on the target domain; Garcia et al. (2021) introduced a vocabulary adaptation scheme to extend the language capacity of multilingual machine translation models; and more recently, Thompson et al. (2019) adapted the Elastic Weight Consolidation method (Kirkpatrick et al., 2016) to mitigate the drop in general-domain performance of NMT models.

3 Models

3.1 Transformer architecture

Neural encoder-decoder architectures such as the Transformer (Vaswani et al., 2017) are the current standard in Machine Translation (Barrault et al., 2020), and most Natural Language Tasks (Devlin et al., 2018).

This state-of-the-art architecture is based entirely on the concept of *attention* (Bahdanau et al., 2015; Luong et al., 2015) to draw global dependencies between the input and output. Because of this, it can process all its sequences in parallel and achieve significant performance improvements compared to previous architectures (Sutskever et al., 2014; Cho et al., 2014; Wu et al., 2016). Furthermore, this architecture does not use any recurrent layer to deal with temporal sequences. Instead, it uses a mask-based approach along with positional embeddings to encode the temporal information of its sequences.

4 Experimental setup

4.1 Datasets

The data used for this work comes from the Europarl dataset (See Table 1), which contain parallel sentences extracted from the European Parliament website[1].

Dataset	Languages	Train size	Val/Test size
Europarl	en-es	100K	1000
Europarl	en-fr	100K	1000
Europarl	en-de	100K	1000
Europarl	en-cz	100K	1000

Table 1: Datasets partitions. In order to avoid potential biases during the experimentation, each dataset was forced to contain 100,000 sentences.

[1] Europarl dataset: `https://www.statmt.org/europarl/`

4.2 Training details

First, all language pairs were concatenated to train a multilingual vocabulary based on Unigrams (Kudo, 2018), with a size of 16,000 tokens plus another 256 for byte-fallback, using SentencePiece (Kudo and Richardson, 2018). Moreover, to avoid language biases, all language pairs had the same number of sentences (and a similar amount of tokens).

To train our models, we used AutoNMT (Carrión and Casacuberta, 2022), a tool to streamline the research of seq2seq models, by automating the preprocessing, training, and evaluation of NMT models. Specifically, we used a simplified version of the standard Transformer with around 4.1M to 25M parameters depending on the vocabulary size. This small Transformer consisted of 3 layers, 8 heads, 256 for the embedding dimension, and 512 for the feedforward layer. Similarly, the training hyper-parameters were quite standard for all models: CrossEntropy (without label smoothing), Adam as the optimizer, 4096 tokens/batch or a batch of 128 sentences, max token length of 150, clip-norm of 1.0, a maximum epoch of 50 epochs with early stopping (patience=10).

The training order was always the same: 1) English-Spanish; 2) English-French; 3) English-German; and 4) English-Czech. Similarly, all models were evaluated for each language pair plus an additional one, where all pairs were merged.

All training was done using two NVIDIA GeForce RTX 2080, with 8GB each.

4.3 Evaluation metrics

Automatic metrics compute the quality of a model by comparing its output with a reference translation written by a human.

Given that BLEU (Papineni et al., 2002) is the most popular metric for machine translation, but it is pretty sensitive to chosen parameters and implementation, we used SacreBLEU (Post, 2018), the reference BLEU implementation for the WMT conference. Additionally, we contrasted our results using BERTScore (Zhang et al., 2019).

- **BiLingual Evaluation Understudy (BLEU)**: Computes a similarity score between the machine translation and one or several reference translations, based on the n-gram precision and a penalty for short translations.

5 Experimentation

5.1 Characterizing the catastrophic forgetting in Machine Translation

In this experiment, we trained a multilingual machine translation (MNMT) model sequentially to study the effects of the catastrophic forgetting phenomenon, as a function of the number of tasks learned (language pairs) and the ratios of past data used during the learning of the new task.

To do so, we began by training a base model for the English-Spanish pair alone (Task #1). Then, we re-trained it using the English-French pair (Task #2) and the English-German pair (Task #3). Later, we added the English-Czech pair (Task #4) for completeness. For each of these tasks, we trained several models for which we varied the ratio of past data that those models could see during the learning of the new task (interleaved data). Finally, we trained another multilingual model using all language pairs (en-es/fr/de) at once to serve as a comparison against the multilingual NMT model trained sequentially.

In Figure 1 we have the results of this experiment. The rows indicate the task being learned, and the columns show the performance of each model for each of the past tasks during the learning of the new task. Moreover, the values annotated at the end of each line indicate the ratio of past data used per batch during the learning of the new task. By looking at Figure 1, we can see that after training for the English-Spanish task, the model achieved a performance

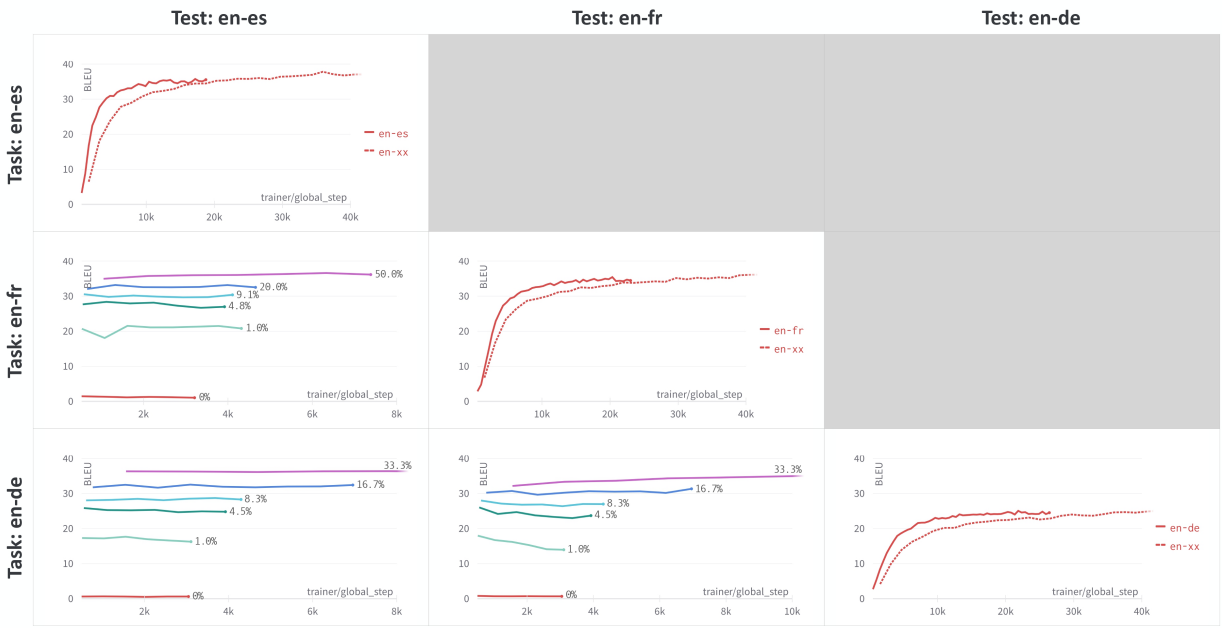

Figure 1: **Using past data to naïvely tackle the CF problem**: When no past data (0%) is used during the learning of a new task, the model forgets everything about the past tasks (flat red lines). However, when a minimal amount of past data is added as a reminder (>1%) during the learning of the new task, the forgetting of these past tasks is significantly reduced.

of 35pts of BLEU. However, when that same model was re-trained for the English-French task, the performance for the past task (English-Spanish) was significantly affected depending on the ratio of past data used during the re-training. For example, when no past data was used during the learning of the new task (English-French), the model's performance on the past task (English-Spanish) dropped to zero (flat red lines). In contrast, as soon as we increased the ratio of past data per batch from 0.0% to 1.0%, the model retained around 60% of its previous performance for that task and 95% of it when 20% of past data was used per batch. Similarly, this very same effect was observed after re-training that trained model (*en-es* $\rightarrow$ *en-fr*) for the English-German task. When no past data was used, the model forgot both the English-Spanish and the English-French tasks. However, as soon as the ratio of past data was slightly increased, the model could retain most of its past knowledge for these tasks.

Interestingly, another thing to point out from these results is that, as the model learns the new task, the performance on the previous tasks remained fairly stable overall. This was quite unexpected for us since it is expected to observe a constant decline in the performance of all tasks as the new task was being learned. However, we did not see this effect until at least two tasks had been learned, and only when we used minimal ratios of past data (i.e., 1%).

Consequently, we explored this phenomenon more closely and added a fourth task to the experiment, the English-Czech pair. As a result, we can see in Figure 3 that with the addition of this new task (en-cz), the effects of the catastrophic forgetting problem became more significant when compared to the previous experiment (see red lines for the en-es/fr/de tasks) since now, the performance in past tasks was steadily declining while the new task was being learned. Hence, this confirmed our previous assumption, given that as the model reaches its learning capacity, that is, its saturation point, it has to forget more and more information despite the refreshments

of past data to keep learning new information.

Next, we decided to compare these results with the very same model architecture but trained from scratch, for which all language pairs were available at the training time. Interestingly, no significant differences were found between this model and its sequential version (See dashed (en-xx) and solid (en-es/fr/de) red lines in Figure 1). Therefore, this confirms that as long as a model has sufficient capacity, its performance should not vary significantly regardless of whether it has been trained for all tasks simultaneously or has been trained sequentially using a continual learning approach, such as the one from this experiment using minimal amounts of past data to retain past knowledge. Furthermore, training a model sequentially, using this approach or any other, has the advantage that the training is much more efficient since it only has to re-train the model for the new task rather than for all tasks again.

Finally, these results appear to indicate that by following a strategy as simple as adding tiny fractions of past data during the training of the new task, it is possible to significantly mitigate the effects of catastrophic forgetting problem, enabling sequential training when training data are very scarce. For example, a typical scenario for this could be to extend the number of tasks or classes supported by a pre-trained model for which we do not have the original training data but have access to other similar despite minimal datasets, or even when we do not have more data, but we can afford to annotate a few extra samples semi-automatically.

Furthermore, with this experiment we demonstrate that contrary to popular belief, to maintain the performance of a model on past tasks, one does not need to use all the previous data, but a minimal amount of past data during the learning of the new task.

5.2 Oversampling past data

Given that our base model had sufficient capacity to cope with these tasks, either sequentially or jointly (obtaining very similar performances), we decided to focus our efforts on maximizing the performance in past tasks but minimizing the amount of past data needed to control the forgetting. The reason for adopting this approach was to improve the learning efficiency of new tasks since it is not the same to learn a new single task using minimal past data refreshments than to use large amounts of past data. Besides, in a data loss scenario, it will always be more accessible to label a few past samples manually to control the catastrophic forgetting than to label a whole new dataset from scratch.

Consequently, we first tried to control the catastrophic forgetting by oversampling these sets of past data so that they would have the same weight as the new data. That is, if the new task had 10,000 samples and the previous tasks had 1,000 and 2,000 samples, we would assign a weighting coefficient of 1.0 for the first task, and 10.0 and 5.0 for the second and third tasks. We then used these weighting coefficients to oversample these task samples.

This experiment can be seen in Figure 2, where the non-oversampled models are characterized by dashed lines and the oversampled models by solid lines. This oversampling approach proved to be quite beneficial when the ratios of oversampled data were minimal (around 1%), achieving up to +4pts of BLEU with respect to the non-oversampled models. However, when we increased the ratio of past data from 1% to 4.5% or more, this strategy did not provide significant results and made the models more prone to overfitting on past tasks compared to the non-oversampled models. In addition to this, the non-oversampled models performed slightly better on the new task than the oversampled ones due to the use of higher ratios of new data per batch (i.e., 1% English-Spanish + 99% English-French vs. 50% English-Spanish (oversampled) + 50% English-French).

This oversampling experiment was initially conducted *physically*, that is, by repeating sentences, due to the simplicity of this approach. However, in order to reduce the performance gap between the oversampled and non-oversampled models that was observed during the learning

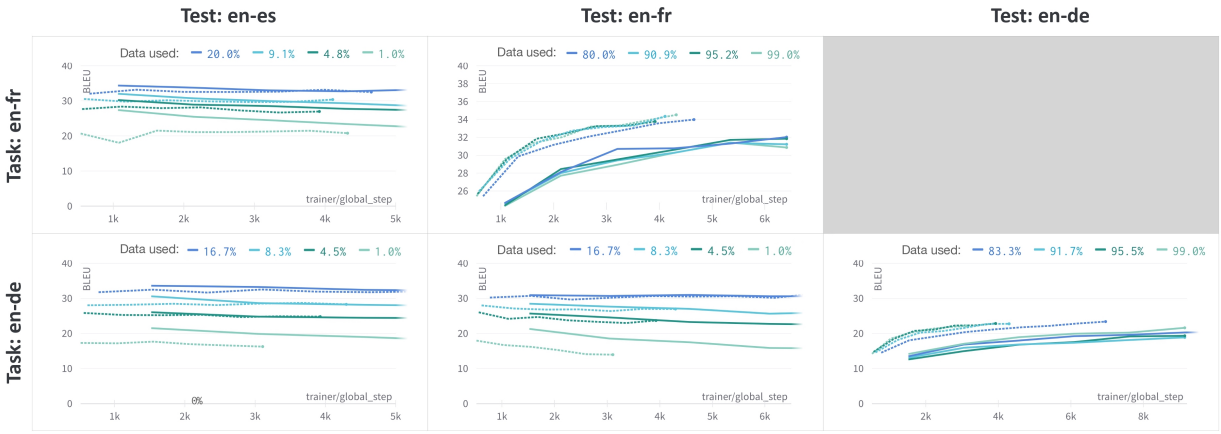

Figure 2: **Oversampling of past data** was effective, especially in scenarios where only minimal amounts of past data were available (<1%). Furthermore, in the case where all tasks had the same weighting (this figure), the oversampled models (solid lines) needed more time than the non-oversampled models (dashed lines) to achieve similar performance levels. Hence, we later reduced the weight of these past tasks.

of the new task, we decided to repeat the previous experiment but this time, performing a *virtual* oversampling so that we could actively rescale the task-weights to perform minor adjustments to better control for these convergence issues.

This idea is described in Equation 1, where x is the input, y is the target, t is the task and w_t is the weight of the task t.

$$\mathcal{L}_w(x, y, t) = \{l_1, l_2, ...l_n\}, \quad l_i = -w_t \log \frac{exp(x_i)}{\sum_{j=1}^{K} exp(x_j))} \cdot 1\{y_i \neq ignore\_index\} \quad (1)$$

These weights were determined both manually and automatically (learned). However, we obtained better results by determining them manually than automatically, since the automatic approach tended to overweight the easiest task in detriment of the others. Nevertheless, we were able to compensate part of the performance mismatches mentioned before, although we found that when weights were determined manually, it was much easier to overfit for a specific task. Besides, due to the *pareto frontier*, we could not improve performance on all tasks simultaneously by simply performing an active task re-weighting because when we improved performance on one or more tasks, we always ended up compromising performance improvements on another task.

Even though this task re-weighting strategy was primarily beneficial for low-resource scenarios of past data, we found it to be quite helpful in finding better balance compromises between the performances of the different tasks, and avoiding greedy behaviors during learning of the new task.

5.3 Few-Shot Regularization

As discussed in Section 5.1, if we increase the number of tasks learned and do not increase the capacity of the model, sooner or later, the model will reach a saturation point. Therefore, the performance on past tasks will start to worsen instead of remaining stable as before. Furthermore, we have shown that the presence of this phenomenon is accelerated in scenarios where

the amount of past data is minimal (<1%), but at the same time, we know that not much past data is needed to mitigate the effects of the catastrophic forgetting. Accordingly, we learned a fourth task (English-Czech) where the model could only see a few samples (1% of past data per batch) in order to make this forgetting phenomenon more noticeable during our experimentation (See Figure 3, red lines). Therefore, our goal here was to show that with a minimal amount of past samples and a simple regularization mechanism, we could be able to mitigate the effects of the catastrophic forgetting phenomenon, and even, improve the performance on some past tasks.

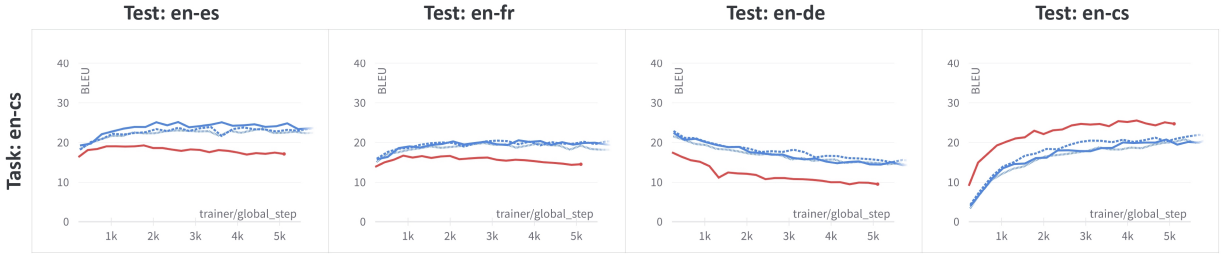

Figure 3: **Few-shot regularization**: The effects of the catastrophic forgetting problem become more noticeable as the model reaches its saturation point after learning more tasks than it can handle, so it starts to degrade its performance of past tasks (see red lines). In contrast, when using the loss function proposed in this section, we can appreciate how the effects of catastrophic forgetting are significantly reduced (see blue lines), albeit at the cost of obtaining slower convergence on the new task.

Consequently, we derived a loss function that minimizes the forgetting of previously learned tasks by actively re-weighting past samples and penalizing weights that deviate too much from the original model. That is, initially, all tasks should contribute equally to the loss regardless of the amount of past data available (oversampling), but then, these weights were slightly modified to ease the convergence of the new task (re-weighting). Additionally, errors should be penalized more severely on past tasks than on the new ones so that we could have more control over the forgetting effects. Furthermore, we wanted to penalize changes in weights that are assumed to be relevant for the past tasks but are not for the new task. To do so, we added a regularization term, based on the knowledge distillation loss derived by Hinton et al. (2015), that allowed us to control for these deviations in past tasks with respect to the previous version of the same model, which is presumed to be better in past tasks due to the effects of catastrophic forgetting phenomenon. Finally, we added the well-known L2 regularization.

This loss function is described in Equation 2, where $\mathcal{L}_w$ is the weighted loss from Section 5.2, $\mathcal{L}_m$ is the weight penalization function described above, t is the task from which the pair (x, y) belongs, y_{ref} is the output of the reference model (i.e., epoch 0), α and β are hyper-parameters to define the importance of the current loss, and the weight deviation w.r.t the past model, and δ is a vector to control the importance of past tasks with an exponential penalization. These (hyper-)parameters can be either set manually or learned during training (see below).

$$\mathcal{L}(x, y, t) = (\alpha \cdot \mathcal{L}_w(x, y, t) + \beta \cdot \mathcal{L}_m^{(t \neq T)}(y, y_{ref}))^{\delta_i} + \gamma \cdot \|\theta\|_2^2, \quad \delta_T = 1 \tag{2}$$

This equation has four components. The first component is the weighted loss $\mathcal{L}_w$, whose purpose is to ensure that each task is equally important regardless of the number of samples. The second term is the distillation loss, which acts as a regularizer to penalize the changes

in the past task between the current model (output y) and the reference model (output y_{ref}), that is, the version that was used as a starting point for this new task. The third component is the vector δ, which penalizes pasts tasks exponentially for a faster response to the effects of the catastrophic forgetting problem (during our experimentation, we set $\delta_T = 1.0$, although it could have had other values). The fourth component is the L2 regularization to help with overfitting. Finally, the hyperparameters α and β were determined both manually and (semi-)automatically. First, we tried to learn these hyperparameters (along with δ) automatically during the training of the new task. However, we obtained worse results than when we adjusted them manually due to the problems mentioned in Section 5.2 related to the *Pareto frontier* and because we were considering which tasks were more challenging to learn. Then, we tried to learn them semi-automatically. That is, we learned them automatically while clamping them into a predefined range.

As a result, in Figure 3 we find the comparison between a model trained previously on the English-German task (red line), which only adds a minimal amount of past data (1%) to alleviate the forgetting, and the very same model (blue lines), which in addition to using minimal past data during training (1%) to tackle the forgetting problem, it uses the loss function proposed in this section. Also, we have included a few runs (not cherry-picked) of this proposed model instead of just one to better represent its behavior and support our conclusions more robustly. Accordingly, it can be seen in Figure 3 that the proposed model (blue lines) significantly mitigates the effects of catastrophic forgetting with regard to the other model. For example, on the first two tasks (en-es, en-fr), it even improves the base performance; and on the third task, despite losing some performance concerning its initial result, the loss is significantly smaller than the one from the reference model (red line). However, although our model tends to converge a bit slower due to the additional control terms, both models end up converging to similar performances [2].

Therefore, this loss function, in addition to a minimal amount of past data to exploit past knowledge information, presents itself as an extremely simple mechanism to tackle the catastrophic problem with no additional computational costs.

6 Conclusions

This work has studied the catastrophic forgetting problem in machine translation framed as a sequential learning problem for a multilingual machine translation system, where each new language pair is considered a new task.

From studying the effects of the catastrophic forgetting problem as a function of the number of learned tasks and the ratios of past data used during the learning of the new task, we discovered that even with minimal amounts of past data, we could retain up a 95% of the performance in past tasks. Then, we tried to boost the performance in past tasks through an oversampling strategy. However, this approach was primarily beneficial for scenarios where only minimal amounts of past data were available ($<$1%).

Finally, we derived a new loss function based on actively re-weighting past tasks and penalizing weights that deviate too much from the original model to minimize forgetting past tasks while learning the new one. This approach has practically no extra cost and shines by simplicity when compared to other popular but more complex and resource-hungry approaches.

This work suggests that to easily mitigate the effects of the catastrophic forgetting in machine translation with no extra cost, we only need a minimal amount of past data and a simple regularization function that exploits past knowledge information.

[2] With a smaller learning rate and bit more training both reached the same performance.

Acknowledgment

Work supported by the European Commission (H2020) under the SELENE project (grant agreement no 871467) and the project Deep learning for adaptive and multimodal interaction in pattern recognition (DeepPattern) (grant agreement PROMETEO/2019/121). We gratefully acknowledge the support of NVIDIA Corporation with the donation of a GPU used for part of this research.

References

Aljundi, R., Babiloni, F., Elhoseiny, M., Rohrbach, M., and Tuytelaars, T. (2017). Memory aware synapses: Learning what (not) to forget. *CoRR*, abs/1711.09601.

Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Barrault, L., Biesialska, M., Bojar, O., Costa-jussà, M. R., Federmann, C., Graham, Y., Grundkiewicz, R., Haddow, B., Huck, M., Joanis, E., Kocmi, T., Koehn, P., Lo, C.-k., Ljubešić, N., Monz, C., Morishita, M., Nagata, M., Nakazawa, T., Pal, S., Post, M., and Zampieri, M. (2020). Findings of the 2020 conference on machine translation (WMT20). In *Proceedings of the Fifth Conference on Machine Translation*, pages 1–55.

Carpenter, G. A. and Grossberg, S. (1987). A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer Vision, Graphics, and Image Processing*, 37(1):54–115.

Carrión, S. and Casacuberta, F. (2022). Autonmt: A framework to streamline the research of seq2seq models.

Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on EMNLP*, pages 1724–1734.

Devlin, J., Chang, M., Lee, K., and Toutanova, K. (2018). BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Draelos, T. J., Miner, N. E., Lamb, C. C., Vineyard, C. M., Carlson, K. D., James, C. D., and Aimone, J. B. (2016). Neurogenesis deep learning. *CoRR*, abs/1612.03770.

Fernando, C., Banarse, D., Blundell, C., Zwols, Y., Ha, D., Rusu, A. A., Pritzel, A., and Wierstra, D. (2017). Pathnet: Evolution channels gradient descent in super neural networks. *CoRR*, abs/1701.08734.

Garcia, X., Constant, N., Parikh, A., and Firat, O. (2021). Towards continual learning for multilingual machine translation via vocabulary substitution. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1184–1192, Online. Association for Computational Linguistics.

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

Hu, W., Lin, Z., Liu, B., Tao, C., Tao, Z., Ma, J., Zhao, D., and Yan, R. (2019). Overcoming catastrophic forgetting for continual learning via model adaptation. In *ICLR*.

Jung, H., Ju, J., Jung, M., and Kim, J. (2016). Less-forgetting learning in deep neural networks. *CoRR*, abs/1607.00122.

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D. (2020). Scaling laws for neural language models. *CoRR*, abs/2001.08361.

Kemker, R. and Kanan, C. (2017). Fearnet: Brain-inspired model for incremental learning. *CoRR*, abs/1711.10563.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N. C., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., Hassabis, D., Clopath, C., Kumaran, D., and Hadsell, R. (2016). Overcoming catastrophic forgetting in neural networks. *CoRR*, abs/1612.00796.

Kudo, T. (2018). Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the ACL (Volume 1: Long Papers)*, pages 66–75.

Kudo, T. and Richardson, J. (2018). Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In Blanco, E. and Lu, W., editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018: System Demonstrations, Brussels, Belgium, October 31 - November 4, 2018*, pages 66–71.

Lee, J., Yoon, J., Yang, E., and Hwang, S. J. (2017a). Lifelong learning with dynamically expandable networks. *CoRR*, abs/1708.01547.

Lee, S., Kim, J., Ha, J., and Zhang, B. (2017b). Overcoming catastrophic forgetting by incremental moment matching. *CoRR*, abs/1703.08475.

Li, Z. and Hoiem, D. (2016). Learning without forgetting. *CoRR*, abs/1606.09282.

Liu, T., Ungar, L., and Sedoc, J. (2019). Continual learning for sentence representations using conceptors. *ArXiv*, abs/1904.09187.

Lopez-Paz, D. and Ranzato, M. (2017). Gradient episodic memory for continuum learning. *CoRR*, abs/1706.08840.

Luong, T., Pham, H., and Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on EMNLP*, pages 1412–1421.

McCloskey, M. and Cohen, N. J. (1989). Catastrophic interference in connectionist networks: The sequential learning problem. volume 24 of *Psychology of Learning and Motivation*, pages 109–165. Academic Press.

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 ACL*, ACL '02, page 311–318.

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

Qi, Y., Sachan, D., Felix, M., Padmanabhan, S., and Neubig, G. (2018). When and why are pre-trained word embeddings useful for neural machine translation? In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*.

Rebuffi, S., Kolesnikov, A., and Lampert, C. H. (2016). icarl: Incremental classifier and representation learning. *CoRR*, abs/1611.07725.

Rusu, A. A., Rabinowitz, N. C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., Pascanu, R., and Hadsell, R. (2016). Progressive neural networks. *CoRR*, abs/1606.04671.

Sato, S., Sakuma, J., Yoshinaga, N., Toyoda, M., and Kitsuregawa, M. (2020). Vocabulary adaptation for domain adaptation in neural machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4269–4279, 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 ACL (Volume 1: Long Papers)*, pages 1715–1725.

Shin, H., Lee, J. K., Kim, J., and Kim, J. (2017). Continual learning with deep generative replay. *CoRR*, abs/1705.08690.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., and Weinberger, K. Q., editors, *NIPS*, volume 27.

Thompson, B., Gwinnup, J., Khayrallah, H., Duh, K., and Koehn, P. (2019). Overcoming catastrophic forgetting during domain adaptation of 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 2062–2068, Minneapolis, Minnesota. Association for Computational Linguistics.

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 *Proceedings of the 31st NeurIPS*, NIPS'17, page 6000–6010.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Kaiser, L., Gouws, S., Kato, Y., Kudo, T., Kazawa, H., Stevens, K., Kurian, G., Patil, N., Wang, W., Young, C., Smith, J., Riesa, J., Rudnick, A., Vinyals, O., Corrado, G., Hughes, M., and Dean, J. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.

Xu, H., Liu, B., Shu, L., and Yu, P. S. (2018). Lifelong domain word embedding via meta-learning. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 4510–4516.

Zenke, F., Poole, B., and Ganguli, S. (2017). Continual learning through synaptic intelligence. In Precup, D. and Teh, Y. W., editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3987–3995. PMLR.

Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., and Artzi, Y. (2019). Bertscore: Evaluating text generation with BERT. *CoRR*, abs/1904.09675.