ESANN 2022 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 5-7 October 2022, i6doc.com publ., ISBN 978287587084-1. Available from http://www.i6doc.com/en/.

Continual Incremental Language Learning for Neural Machine Translation

Michele Resta[1] and Davide Bacciu[1]

1 - University of Pisa - Computer Science Department
Largo Bruno Pontecorvo, 3, 56127, Pisa - Italy

Abstract. The paper provides an experimental investigation of the phenomena of catastrophic forgetting for Neural Machine Translation systems. We introduce and describe the continual incremental language learning setting and its analogy with the classical continual learning scenario. The experiments measure the performance loss of a naive incremental training strategy against a jointly trained baseline, and we show the mitigating effect of the replay strategy. To this end, we also introduce a prioritized replay buffer strategy informed by the specific application domain.

1 Introduction

Large multilingual NMT systems have achieved state-of-the-art performances for both high and low resource languages [1, 2] enabling translations using a single end-to-end system. The steady increase in language proficiency of these models emerges from the increase in their parameters count and from the exposure to different languages during training, which enables better generalization. Despite the advantages, jointly training a model is not always possible: the amount of computational resources required may be unfeasible [2] and the training data for all languages of interest needs to be gathered in advance. To overcome such limitations it is possible to use large models and fine-tuning schemes to adapt them to new domains [3]. Adapting to new unseen languages, however, is challenging, and few works in literature have started to face this problem [4, 5]. The ability to continually learn new knowledge while avoiding catastrophic forgetting (CF) [6] is a key feature of biological learning systems and transferring this ability to NMT systems could allow them to mimic human language acquisition abilities. In this work, we discuss a new incremental learning setting, where a single NMT system is sequentially exposed to a stream of experiences and incrementally trained to translate to and from the languages present in each of the experiences. This setting shares some similarities with the task incremental setting in the Continual Learning (CL) literature and allows for investigating the amount of CF and several other effects. We name this setting Continual Incremental Language Learning (CILL). The remainder of this paper is organized as follows: we formalize the problem and the mitigation strategies in section 2. Section 3 describes the experimental setting and experiments' outcome, together with a discussion of the results. Lastly, we conclude by highlighting several possible future expansions in section 4.

ESANN 2022 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 5-7 October 2022, i6doc.com publ., ISBN 978287587084-1. Available from http://www.i6doc.com/en/.

2 Continual Incremental Language Learning

We focus on training an NMT model that incrementally learns new languages while retaining previous knowledge: we want to model $P(Y|X)$ for all the languages of interest, with X and Y being the source and target sentence respectively. The sub-word vocabulary is built in advance and kept fixed; the same is true for the model architecture and number of parameters. The desired translation direction is denoted by using language tokens as in [1]. The learning process is divided into E learning experiences, where in each $e_i \in E$ the model is exposed to a pair of languages l_1, l_2 and has access to a training set T_i comprising one or both possible translation directions: $T_i = \{(x_{l_1}, x_{l_2}) \cup (x_{l_2}, x_{l_1})\}$. The CILL scenario, similarly to the task incremental setting in CL, divides the learning process into experiences (if we consider translating from and to a language as a single task) but uses a fixed model and no task labels. Having a fixed number of parameters and a fixed vocabulary are the main differences between CILL and what is described in [4, 5]. We design a set of experiments to quantify the amount of catastrophic forgetting occurring during the subsequent experiences and the effectiveness of two replay strategies with different buffer sizes. The size of the buffer was chosen to be 1% or 5% of the total training data, uniformly divided for all language directions of interest: $B = \bigcup_{i=1}^{|E|} b_i$. At the end of each e_i experience the corresponding portion $b_i = \frac{B}{|E|}$ is populated with an equal number of samples from both translation directions in T_i.

We propose to populate B according to different schemes. The first strategy is a random one: at the end of each e_i, we fill b_i with random sentences from the current training corpus. The second filling scheme for the buffer is inspired by the Zipf law. Given the inverse relation for the rank-frequency of utterances in a given corpus, we can observe sentences with a large number of common words and conversely, sentences made by rare words. These seldom encountered sentences and words constitute rare events for an NMT translation system and have a large information content compared to frequent ones. For such reason, we devised a scoring scheme for corpus sentences that takes into consideration word frequency to test the behaviour of the model when the buffers are filled with rare sentences. We compute word-frequencies for both the source and target corpus (CP_{src}, CP_{tgt}) used in the training experience separately, obtaining $freq_w \forall\, w \in CP_{src} \cup CP_{tgt}$. The score for a sentence s is computed considering also the corresponding target sentence s_{tgt} and is given by:

$$score_{lc}(s) = \sum_{w \in s \cup s_{tgt}} freq_w + \alpha_1 + \alpha_2 + \alpha_3,$$

with α_i being score boosting factors. We define the *char ratio* of a sentence as $c_r(s) = n_c/l$ with n_c being the number of ASCII characters and l its total length. In our experimental benchmarks leveraging the Europarl [7] dataset, we noticed that sentences with $c_r < 87\%$ are noisy and contain mostly symbols and numbers. Therefore we defined α_i as follows:

ESANN 2022 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 5-7 October 2022, i6doc.com publ., ISBN 978287587084-1. Available from http://www.i6doc.com/en/.

$$\alpha_1 = \begin{cases} (1 - c_r) \cdot 10^6, & \text{if } c_r < 0.87 \\ 0, & \text{otherwise} \end{cases}, \quad \alpha_2 = \begin{cases} 5 \cdot 10^5, & \text{if } |w| \in s < 5 \\ 0, & \text{otherwise} \end{cases},$$

$$\alpha_3 = \begin{cases} l \cdot 10^4, & \text{if } l > 300 \\ 0, & \text{otherwise} \end{cases}$$

With the above scheme short sentences, very long sentences, and noisy ones receive larger scores. After the scoring phase, we sort the sentences in ascending score order and fill the buffer with the first k sentences with $k = |b_i|$ being the available buffer size for e_i. We also experimented with a different scoring scheme aimed at selecting sentences containing frequent words:

$$score_{mc}(s) = \sum_{w \in s \cup s_{tgt}} freq_w - \alpha_1 + \beta, \qquad \beta = \begin{cases} l \cdot 10^6, & \text{if } l < 300 \\ 0, & \text{otherwise} \end{cases}$$

In this case, sentences are sorted in a reverse way and we select the top scoring ones. In the following, we refer to the first scoring scheme as $Zipf_{lc}$ (less common) and to the second as $Zipf_{mc}$ (most common).

To evaluate strategies, after each of the $e \in E$ experiences we compute the test score $R_{e,d}$ of the model on each language direction, the average BLEU, and the Backward Transfer:

$$ACC = \frac{1}{2 \cdot |E|} \sum_{i=1}^{2 \cdot |E|} R_{e,i} \quad BWT = \frac{1}{2(E-1)} \sum_{i=1}^{E-1} R_{e,i} - R_{i,i}$$

where in the BWT expresssion $R_{i,j} = R_{i,f} + R_{i,b}$ is the sum of the scores of the forward and backward translation direction for the pair of languages considered in the i-th experience.

3 Experimental Results

We use 2 million sentences from the Europarl [7] dataset in English, French, German, and Spanish with parallel sentences for all the 12 translation directions. We divide the learning process into 6 different experiences. In each experience, the model is exposed to a pair of translation directions: e.g German to English and English to German. The total amount of sentences is around 23 million. As validation and test set we use newstest2012 and newstest2013 from WMT[1], respectively. The training corpus was preprocessed with Moses scripts [8], we chose to limit the maximum length of a sentence to 100 tokens and removed all the duplicated ones. A shared Byte Pair Encoding [9] vocabulary of 32K tokens was created for all the languages by using the sententencepiece library [2]. For all

[1] https://www.statmt.org/

[2] https://github.com/google/sentencepiece

ESANN 2022 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 5-7 October 2022, i6doc.com publ., ISBN 978287587084-1. Available from http://www.i6doc.com/en/.

the experiments we used the Transformer "base" [10] provided by Marian [11]. The model was configured with 6 encoder blocks and 6 decoder blocks, each with 8 attention heads and a FF size of 2048. We used tied embedding with a dimension of 512 and set Dropout and Label smoothing to 0.1. The models are trained using the Adam optimizer ($\epsilon = 10^{-6}, \beta_2 = 0.98$), the learning rate is linearly increased from 10^{-6} to $5 \cdot 10^{-4}$ in 16K steps, then is decreased using an inverse square root schedule. We set the number of epochs to 150, validating the models every 5K steps and use early stopping with patience = 10 (we stop training if there are no improvements on development steps for 10 subsequent validations). The batch size is roughly 250 tokens for the joint model and 100 for all the others. All the models were trained on a two Nvidia Telsa V100 GPU with 16GB of RAM.

Exp.	De-En	En-De	De-Es	Es-De	De-Fr	Fr-De	En-Es	Es-En	En-Fr	Fr-En	Es-Fr	Fr-Es	ACC	BWT	Time
Joint Trained															
1	26.0	22.8	26.3	21.6	26.1	21.1	30.5	28.7	29.7	28.0	30.8	30.7	26.85	-	378 h
Pure Incremental															
1	25.4	22.4	1.6	1.4	1.7	1.6	1.5	1.2	1.5	1.3	1.4	1.4	5.20	0.00	
2	1.7	3.0	26.1	20.9	2.6	1.9	3.4	1.8	1.4	0.8	1.7	2.1	5.61	-21.55	
3	1.6	3.1	2.4	2.3	25.8	21.0	1.5	1.1	4.2	1.9	2.7	1.9	5.79	-21.35	
4	1.2	1.7	1.3	1.7	0.8	1.0	29.7	28.2	2.6	2.2	2.0	2.6	6.25	-22.31	
5	1.0	1.6	0.8	1.2	1.3	1.6	2.6	5.8	26.8	26.2	4.7	2.8	6.36	-22.95	
6	0.7	1.2	1.4	1.7	1.3	1.9	5.8	1.8	5.3	1.9	31.3	30.9	7.10	-22.95	58 h
Buffer 5% - Random Replay															
1	26.2	22.9	1.7	1.0	1.8	1.1	1.6	0.8	1.6	1.0	1.1	0.8	5.13	0.00	
2	21.5	20.0	26.2	21.1	2.6	1.7	23.0	21.0	1.7	2.1	1.8	2.1	12.06	-3.80	
3	21.1	19.6	21.3	18.8	25.7	20.8	19.8	17.1	22.8	19.9	22.4	21.7	20.91	-3.90	
4	24.0	20.1	24.8	19.1	19.9	17.3	30.5	29.1	20.2	18.9	23.6	21.9	22.45	-2.95	
5	23.9	19.9	22.6	18.1	23.9	18.7	27.0	26.8	29.3	27.9	27.9	27.1	24.42	-2.70	
6	22.7	19.5	23.9	18.7	23.5	18.7	28.1	26.5	27.8	26.1	30.3	30.1	**24.65**	-2.42	155
Buffer 5% - $Zipf_{lc}$ Replay															
6	13.4	10.4	13.1	10.2	14.0	12.9	19.1	17.4	24.8	19.8	26.6	26.3	17.33	-9.50	157 h
Buffer 5% - $Zipf_{mc}$ Replay															
6	23.2	19.5	23.6	18.5	23.5	18.3	28.0	26.3	28.2	26.8	29.2	28.9	24.50	**-2.22**	150 h
Buffer 1% - Random Replay															
6	15.9	13.5	19.7	13.9	19.0	14.0	25.2	22.1	24.6	21.6	30.9	30.6	20.91	-7.08	182 h
Buffer 1% - $Zipf_{lc}$ Replay															
6	3.4	2.9	8.3	3.7	7.4	3.9	12.3	5.7	11.8	5.9	27.4	27.6	10.02	-16.61	112 h
Buffer 1% - $Zipf_{mc}$ Replay															
6	17.8	15.3	21.3	15.2	20.8	15.2	26.2	23.1	25.7	22.7	31.1	30.7	**22.09**	**-5.75**	135 h

Table 1: Comparison of replay strategies. Scores in BLEU (higher the better). The Time column reports the total time (in hours) across all experiences.

A comparison of the strategies' performance is reported in Table 1: all scores are detokenized BLEU scores, obtained by sacreBLEU[3]. For the sake of brevity, we report scores at the last experience for all strategies, with the exception of the Pure Incremental and Random Replay (Buffer 5%). As expected, a pure incremental approach results in large forgetting and compared to the joint model its average BLEU scores drops by almost 20 points. The random replay strategy with 5% buffer is able to counter the loss of performances effectively, losing less than 3 points to the joint model. The $Zipf_{mc}$ is close to random with comparable average BLEU and slightly better BWT at the end of the last experience while

[3]https://github.com/mjpost/sacrebleu

ESANN 2022 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 5-7 October 2022, i6doc.com publ., ISBN 978287587084-1. Available from http://www.i6doc.com/en/.

the $Zipf_{lc}$ is significantly worse than random. Results are similar for the 1% replay buffer with an average loss of 6 points to the joint model and the $Zipf_{lc}$ strategy performing worse than random with a larger performance gap compared to the 5% case. With this buffer size, the $Zipf_{mc}$ approach outperforms the random one by more than 1 point and has better BWT. The incremental approaches with replay buffer exhibit strong forward transfer (noticeable by inspecting results of the 5% Random strategy) enabling *zero-shot* translation from the end of the second training experience for En↔Es directions and for En↔Fr and Es↔Fr at the end of the second experience. The largest forward transfer occurs with the $Zipf_{mc}$ strategy, followed by the Random, and $Zipf_{lc}$. This aspect is particularly interesting and could possibly be leveraged by choosing a specific ordering for languages in the experiences.

We compared the training time of the strategies with a model that at each experience is trained from scratch on all the languages available, up to that point. In the CILL setting, at the end of the last experience, this model has access to the same amount of data as the joint model. For a fair comparison, we report the cumulative training time in the corresponding row. For this model, the training runs for 378 hours in total, the Pure incremental model runs for 35 hours, the Random 1% and 5% for 182 and 157 hours. The pure incremental approach is less time consuming, albeit at a cost of catastrophic performance drop. The strategies with a 5% replay buffer provide up to 91% the scores of the joint approach with less than 41% of its training time (on average) while the 1% ones provide up to 82% of the scores with only 38% of training time.

4 Conclusion

In this work, we trained several NMT systems under the proposed continual incremental language learning settings and showed that simple continual learning strategies such as Replay are effective in mitigating forgetting and result in scores that are close to joint trained ones using less than half of the training time. After being exposed to several learning experiences the models show *zero-shot* capabilities for several unseen language directions. It could be worth investigating the effect of the ordering of the experiences on model performances and the effectiveness of more complex strategies to mitigate forgetting and further reduce training time. We left these two points as future work.

References

[1] Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the ACL*, 5:339–351, Dec 2017.

[2] Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wen-

zek, Vishrav Chaudhary, et al. Beyond english-centric multilingual machine translation. *Journal of Machine Learning Research*, 22(107):1–48, 2021.

[3] Chenhui Chu, Raj Dabre, and Sadao Kurohashi. An empirical comparison of domain adaptation methods for neural machine translation. pages 385–391, 2017.

[4] Alexandre Berard. Continual learning in multilingual NMT via language-specific embeddings. In *Proceedings of the Sixth Conference on Machine Translation, WMT@EMNLP November 10-11*, pages 542–565, 2021.

[5] Carlos Escolano, Marta R. Costa-jussà, and José A. R. Fonollosa. From bilingual to multilingual neural machine translation by incremental training. In *Proceedings of the 57th Conference of the ACL , ACL*, pages 236–242, 2019.

[6] Robert French. Catastrophic Forgetting in Connectionist Networks. In *Trends in Cognitive Sciences*, volume 3. January 2006.

[7] Philipp Koehn. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of Machine Translation Summit X: Papers, MTSummit 2005, Phuket, Thailand, September 13-15*, pages 79–86, 2005.

[8] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open source toolkit for statistical machine translation. In *ACL, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, June 23-30, 2007, Prague, Czech Republic*, 2007.

[9] Rico Sennrich, Barry Haddow, and Alexandra Birch. 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, August 2016.

[10] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on NeurIPS, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008, 2017.

[11] 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. Marian: Fast neural machine translation in C++. In *Proceedings of ACL, System Demonstrations*, Melbourne, Australia, 2018.