Overcoming Catastrophic Forgetting
During Domain Adaptation of Neural Machine Translation

Brian Thompson[†] **Jeremy Gwinnup**[°] **Huda Khayrallah**[†] **Kevin Duh**[†] **Philipp Koehn**[†]
[†]Johns Hopkins University, [°]Air Force Research Laboratory
{brian.thompson, huda, phi}@jhu.edu,
kevinduh@cs.jhu.edu,
jeremy.gwinnup.1@us.af.mil

This paper is accepted and will be presented at NAACL. Please cite as:

```
@InProceedings{Thompson2019NAACL,
author = {Brian Thompson and Jeremy Gwinnup and Huda Khayrallah
and Kevin Duh and Philipp Koehn},
title = {Overcoming Catastrophic Forgetting During Domain
Adaptation of Neural Machine Translation},
booktitle = "Proceedings of the 2019 Conference of the North
American Chapter of the Association for Computational
Linguistics: Human Language Technologies, Volume 2 (Short
Papers)",
year = "2019",
publisher = "Association for Computational Linguistics",
location = "Minneapolis, Minnesota"
}
```

Overcoming Catastrophic Forgetting During Domain Adaptation of Neural Machine Translation

Brian Thompson[†] **Jeremy Gwinnup**[°] **Huda Khayrallah**[†] **Kevin Duh**[†] **Philipp Koehn**[†]
[†]Johns Hopkins University, [°]Air Force Research Laboratory
{brian.thompson, huda, phi}@jhu.edu,
kevinduh@cs.jhu.edu,
jeremy.gwinnup.1@us.af.mil

Abstract

Continued training is an effective method for domain adaptation in neural machine translation. However, in-domain gains from adaptation come at the expense of general-domain performance. In this work, we interpret the drop in general-domain performance as *catastrophic forgetting* of general-domain knowledge. To mitigate it, we adapt Elastic Weight Consolidation (EWC)—a machine learning method for learning a new task without forgetting previous tasks. Our method retains the majority of general-domain performance lost in continued training without degrading in-domain performance, outperforming the previous state-of-the-art. We also explore the full range of general-domain performance available when some in-domain degradation is acceptable.

1 Introduction

Neural Machine Translation (NMT) performs poorly without large training corpora (Koehn and Knowles, 2017). Domain adaptation is required when there is sufficient data in the desired language pair but insufficient data in the desired *domain* (the topic, genre, style or level of formality). This work focuses on the supervised domain adaptation problem where a small in-domain parallel corpus is available for training. Continued training (Luong and Manning, 2015; Sennrich et al., 2015) (also called fine-tuning), where a model is first trained on general-domain data and then domain adapted by training on in-domain data, is a popular approach in this setting as it leads to empirical improvements in the targeted domain.

One downside of continued training is that the adapted model's ability to translate general-domain sentences is severely degraded during adaptation (Freitag and Al-Onaizan, 2016). We interpret this drop in general-domain performance as *catastrophic forgetting* (Goodfellow et al., 2013) of general-domain translation knowledge. Degradation of general-domain performance may be problematic when the domain adapted NMT system is used to translate text outside its target domain, which can happen if there is a mismatch between the data available for domain-specific training and the test data. Poor performance may also concern end users of these MT systems who are expecting good performance on 'easy' generic sentences.[1]

Elastic Weight Consolidation (EWC) (Kirkpatrick et al., 2017) is a method for training neural networks to learn a new task without forgetting previously learned tasks. We extend EWC to continued training in NMT (see §3): Our first task is to translate general-domain sentences, and our second is to translate domain-specific sentences (without forgetting how to translate general-domain sentences). EWC works by adding a per-parameter regularizer, based on the Fisher Information matrix, while training on the second task. At a high level, the regularization term keeps parameters which are important to general-domain performance close to the initial general-domain model values during continued training, while allowing parameters less important to general-domain performance to adapt more aggressively to the in-domain data.

We show that when adapting general-domain models to the domain of patents, EWC can substantially improve the retention of general-domain performance (up to 18.1 BLEU) without degrading in-domain translation quality. Our proposed method outperforms the previous state-of-the-art method (Dakwale and Monz, 2017) at retaining general-domain performance while adapting to a new domain.

[1] See Cadwell et al. (2018) and Porro Rodriguez et al. (2017) for discussions about lack of trust in MT.

2 Related Work

A few prior studies address the drop in general-domain NMT performance during continued training. Freitag and Al-Onaizan (2016) found that ensembling general- and in-domain models provides most of the in-domain gain from continued training while retaining most of the general-domain performance. Ensembling doubles memory and computational requirements at translation time, which may be impractical for some applications and does not address our more fundamental goal of building a single model that is robust across domains. Chu et al. (2017) found that mixing general-domain data with the in-domain data used for continued training improved general-domain performance of the resulting models, at the expense of training time.

Dakwale and Monz (2017) share our goal of improving the general-domain performance of continued training. They introduce two novel approaches which use the initial, general-domain model to supervise the in-domain model during continued training. The first, *multi-objective fine-tuning*, which they denote MCL, trains the network with a joint objective of standard log-likelihood loss plus a second term based on knowledge distillation (Hinton et al., 2015; Kim and Rush, 2016) of the general-domain model. The second, *multiple-output layer fine tuning*, adds new parameters to the output layer during continued training that are specific to the new domain. They found both methods performed similarly, significantly outperforming ensembling in the more challenging case where domain shift is significant, so we select the simpler MCL as our baseline.

We do not assume that the domain of input sentences is known, thus we do not compare to methods such as LHUC (Vilar, 2018). Our work applies a regularization term to continued training, similar to Miceli Barone et al. (2017) and Khayrallah et al. (2018), but for the purpose of retaining general-domain performance as opposed to improving in-domain performance.

3 Method

Compared to Kirkpatrick et al. (2017), we present a more general derivation of EWC to address the fact that our tasks are not independent. We also show that the diagonal of the Fisher matrix used in EWC is intractable to compute for sequence-to-sequence models with large vocabularies. Instead we propose to approximate it with the diagonal of the *empirical Fisher* (Martens, 2014), which can be computed efficiently using gradients from back-propagation.

At a high level, our method works as follows:

1. Train on the general-domain data, resulting in parameters $\hat{\theta}^G$.
2. Compute the diagonal of the empirical Fisher matrix $\bar{F}$. $\bar{F}_{i,i}$ estimates how important the i^{th} parameter $\hat{\theta}_i^G$ is to the general-domain translation task.
3. Initialize parameters to $\hat{\theta}^G$ and train on in-domain data, using an EWC regularization term which incorporates the diagonal of $\bar{F}$.

Intuitively, the regularization term during continued training keeps a parameter θ_i close to corresponding general-domain parameter $\hat{\theta}_i^G$ if the model's general-domain performance is sensitive to that parameter (i.e., large $\bar{F}_{i,i}$). Parameters to which general-domain performance is less sensitive (i.e., small $\bar{F}_{i,i}$) are allowed to be updated more aggressively to fit the in-domain data.

3.1 Bayesian Rationale for EWC

For the following discussions, let X be the set of all well-formed source sentences and Y be the set of all possible sequences of target words. Training data D consists of translations (x, y). We assume $x \in X$ is drawn from a true underlying distribution of source sentences Q_x, and $y \in Y$ is drawn from a true conditional distribution of correct translations $Q_{y|x}$. Our model, parameterized by θ, computes the conditional probability $P_{y|x} \triangleq P(y|x;\theta)$, which estimates $Q_{y|x}$. Our dataset D is assumed to have come from two distinct tasks: general-domain translation with data D^G and in-domain translation with data D^S(domain-specific). Without loss of generality, $p(D){=}p(D^G)p(D^S|D^G)$. Applying Bayes rule to $\log p(\theta|D)$ and simplifying gives:

$$\log p(\theta|D) = \log p(D^S|D^G,\theta) + \log p(\theta|D^G) - \log p(D^S|D^G) \qquad (1)$$

We aim to maximize Equation 1 for θ:

$$\hat{\theta}^* = \arg\max_\theta \left[\log p(D^S|D^G,\theta) + \log p(\theta|D^G)\right] \qquad (2)$$

3.2 Approximating $\log p(\theta|D^G)$

To efficiently compute Equation 2, we first approximate $p(\theta|D^G)$ as a multivariate Gaussian[2] with mean $\hat{\theta}^G$, obtained by training the network on D^G with standard negative log likelihood (NLL) loss, and diagonal precision matrix (inverse of the covariance matrix) given by the diagonal of the Fisher Information Matrix F:

$$F{=}E_{P_{x,y}}\left[\nabla \log p(x,y|\theta)\nabla \log p(x,y|\theta)^T\right]$$

$$={E_{Q_x}}\left[E_{P_{y|x}}\left[\nabla \log p(y|x,\theta)\nabla \log p(y|x,\theta)^T\right]\right]$$

This is the expected variance of the likelihood function's gradient at θ.[3] The magnitude of $F_{i,i}$ indicates the model's sensitivity to parameter θ_i, on the general-domain translation task. Note that the first expectation is taken with respect to the true distribution of x and can be approximated by training samples. The second expectation is taken with respect to the model distribution $P_{y|x}$, which is impractical for a large sequence-to-sequence model as it requires summing over all possible output sequences.

We approximate the true Fisher with the empirical Fisher $\bar{F}$ (Martens, 2014), where y is not enumerated but fixed to be the training labels:

$$\bar{F}{=}\frac{1}{|D^G|}\sum_{(x,y)\in D^G}\nabla \log p(y|x,\theta)\nabla \log p(y|x,\theta)^T$$

Thus we approximate maximizing $\log p(\theta|D^G)$ in Equation 2 by minimizing $\sum_i \bar{F}_{i,i}\left[\theta_i-\hat{\theta}_i^G\right]^2$. Note that the diagonal of $\bar{F}$ is easily computed from backpropagation gradients.

3.3 Approximating $\log p(D^S|D^G,\theta)$

Tasks are assumed to be independent in the original EWC work (Kirkpatrick et al., 2017), which is unrealistic in the continued training scenario since both tasks are translation in the same language.[4] Since we assume source sentences in D^G and D^S are sampled independently, all dependencies can be attributed to $Q_{y|x}$, representing knowledge of translation (i.e., $D^G \perp\!\!\!\perp D^S \mid Q_{y|x}$). $Q_{y|x}$ is unknown, so we approximate it with our general-domain model ($\hat{\theta}^G$). Furthermore, we will regularize continued training such that θ stays in a region near $\hat{\theta}^G$. Thus we assume $D^G \perp\!\!\!\perp D^S \mid \theta$ during continued training. This allows us to approximate $\log p(D^S|D^G,\theta)$ in Equation 2 with $\log p(D^S|\theta)$, which is simply the likelihood function on D^S.

General-domain			WIPO Patents		
De	Ru	Zh	De	Ru	Zh
323 M	730 M	584 M	3.2 M	0.81 M	0.80 M

Table 1: # English words in the training corpora.

3.4 EWC Loss

Combining the approximations above results in the EWC loss used in continued training:

$$L_{EWC}(\theta){=}L_{NLL}^S(\theta){+}\lambda\sum_i \bar{F}_{i,i}\left[\theta_i{-}\hat{\theta}_i^G\right]^2 \quad (3)$$

Where $L_{NLL}^S(\theta)$ is the standard NLL loss on D^S and λ is a hyper-parameter which weights the importance of the general-domain task. Note that the left-hand side of Equation 3 is still the loss over both the general- and in-domain translation tasks, but the right-hand side is based only on in-domain data. All information from the general-domain data has been collapsed into the second term, which is in the form of a regularizer.

4 Experiments

Our general-domain training data is the concatenation of the parallel portions of the WMT17 news translation task (Bojar et al., 2017) and OpenSubtitles18 (Lison et al., 2018) corpora. For De↔En and Ru↔En, we use `newstest2017` and the final 2500 lines of OpenSubtitles as our test set. We use `newstest2016` and the penultimate 2500 lines of OpenSubtitles as the development set. For Zh↔En, we use the final and penultimate 4000 lines of the UN portion of the WMT data and the final and penultimate 2500 lines of OpenSubtitles as our test and development sets, respectively.

We use the World Intellectual Property Organization (WIPO) COPPA-V2 corpus (Junczys-Dowmunt et al., 2016) as our in-domain dataset. The WIPO data consist of parallel sentences from international patent application abstracts. WIPO De↔En data are large enough to train strong in-domain systems (Thompson et al., 2018), so we truncate to 100k lines to simulate a more interesting domain adaptation scenario.

We reserve 3000 lines each for in-domain development and test sets. We apply the Moses tok-

[2] For background, see MacKay (1992).

[3] See Martens (2014) for detailed derivation.

[4] The fact that continued training works is strong evidence that the in-domain translations are *not* independent of the general-domain translations.

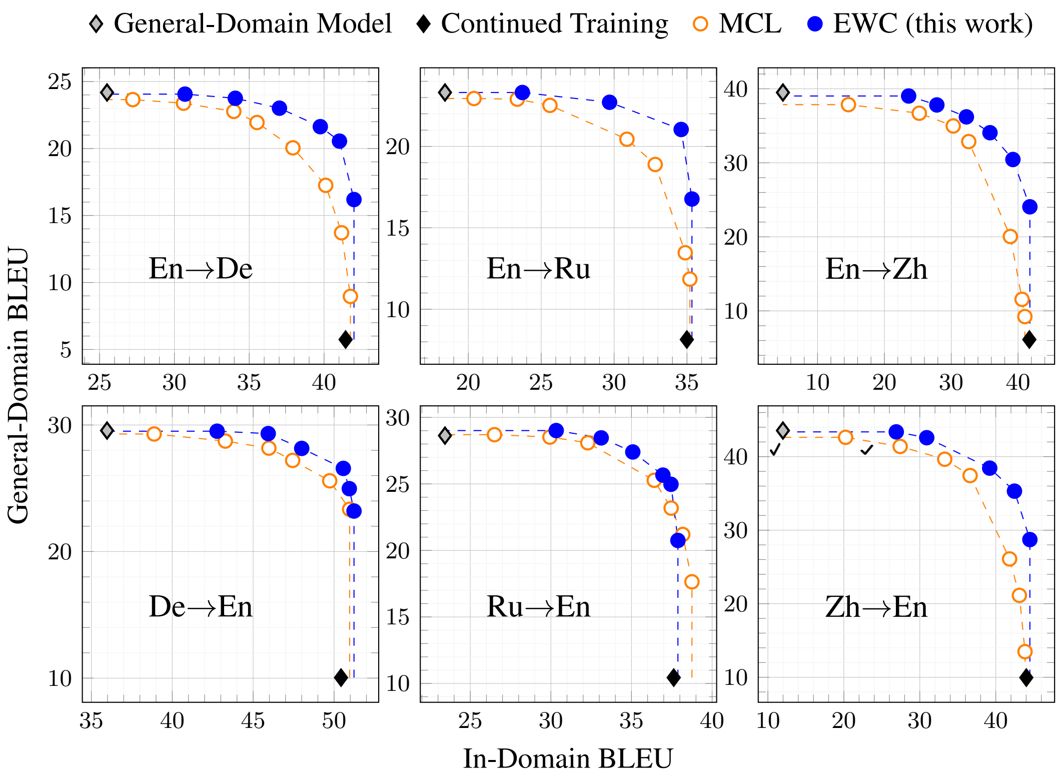

Figure 1: Performance trade-off for MCL and EWC: Convex hull of grid search over learning rate and regularization amount. x-axis is in-domain BLEU and y-axis is general-domain BLEU, so the desired operating point is the top right corner. Initial general-domain model (GD) and continued training (CT) points are shown for comparison.

enizer (Koehn et al., 2007) and byte-pair encoding (BPE) (Sennrich et al., 2016). We train separate BPE models for the source and target languages, each with a vocabulary size of approximately 30k. BPE is trained on the out-of-domain corpus only and then applied to the training, development, and test data for both out-of-domain and in-domain datasets. Token counts for corpora are shown in Table 1.

We implemented[5] both EWC and MCL in Sockeye (Hieber et al., 2017). To avoid floating point issues, we normalize the empirical Fisher diagonal to have a mean value of 1.0 instead of dividing by the number of sentences. For efficiency, we compute gradients for a batch of sentences prior to squaring and accumulating them. Fisher regularization is implemented as weight decay (towards $\hat{\theta}^G$) in Adam (Kingma and Ba, 2014).

Preliminary experiments in Ru→En found no meaningful difference in general-domain or in-domain performance when computing the diagonal of $\bar{F}$ on varying amounts of data ranging from 500k sentences to the full dataset. We also tried computing the diagonal of $\bar{F}$ on held-out data, as there is some evidence that estimating Fisher on held out data reduces overfitting in natural gradient descent (Pascanu and Bengio, 2013). However, we again found no meaningful differences. All results presented herein estimate the the diagonal of $\bar{F}$ on 500k training data sentences, which took less than an hour on a GTX 1080 Ti GPU.

We use a two-layer LSTM network with hidden unit size 512. The general-domain models are trained with a learning rate of 3E-4. We use dropout (0.1) on both RNN inputs and states. We compute lower-cased `multi-bleu.perl`. We use label smoothing (0.1) for all experiments except with MCL, because MCL explicitly regularizes the output distribution.

MCL uses an interpolation of the cross entropy between the output distribution of the model being trained and the general-domain models output distribution (scaled by α) and the standard training loss (scaled by $1-\alpha$). For MCL, we do a grid search over learning rates (10^{-4}, 10^{-5}, 10^{-6}) and α values of (0.1, 0.3, 0.5, 0.7, 0.9). For EWC, we do a grid search over the same learning rates and weight decay values of (10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}).

[5] github.com/thompsonb/sockeye_ewc

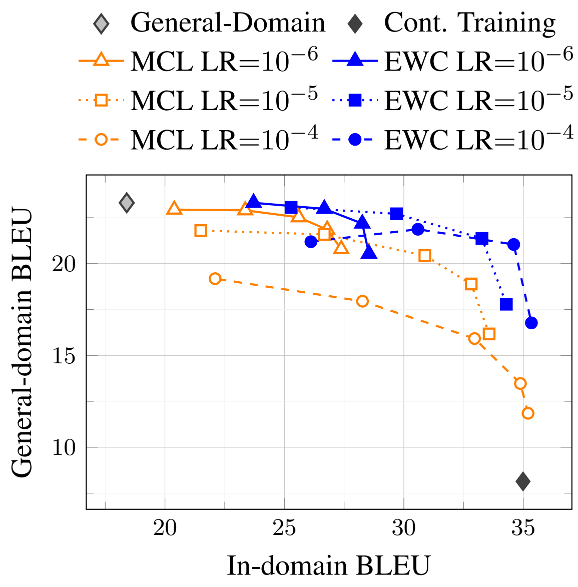

Figure 2: En→Ru results for various learning rates, for both MCL and EWC. Regularization amount increases from left to right for each trace. General-domain and continued training points shown for reference.

5 Results

We present the full in- and general-domain performance trade-off[6] for both EWC and MCL in Figure 1. This is computed by taking the convex hull of a grid search over learning rate and regularization amount for each method. EWC outperforms MCL at all operating points with the exception of Ru→En, where MCL provides a small in-domain performance improvement at lower general-domain performance; this was also observed in Khayrallah et al. (2018).

Figure 2 shows an example result (for En→Ru) of the grid search prior to taking the convex hull. We see similar trends between the three pairs of MCL/EWC curves at corresponding learning rates, but in each case EWC is further up/right, indicating better performance. Note that for both EWC and MCL, both learning rate and regularization amount have a large impact on final in- and general-domain performance.

General-domain gains for *no* in-domain performance degradation are presented in Table 2. Our method provides large general-domain gains (between 8.0 and 18.1 BLEU), regaining the majority of general-domain performance lost in continued training and substantially outperforming MCL.

Langs	GD	CT	MCL	EWC
En→De	24.2	5.7	9.0 (+3.2)	16.2 **(+10.5)**
De→En	29.6	10.0	23.3 (+13.3)	26.6 **(+16.5)**
En→Ru	23.3	8.1	11.8 (+3.7)	16.8 **(+8.6)**
Ru→En	28.6	10.4	21.2 (+10.8)	21.5 **(+11.1)**
En→Zh	39.5	6.1	6.1 (+0.0)	24.1 **(+17.9)**
Zh→En	43.5	9.9	9.9 (+0.0)	28.7 **(+18.8)**

Table 2: General-domain BLEU for: general-domain model prior to adaptation (GD), standard continued training (CT), and best performing MCL and EWC models with *no* in-domain degradation compared to CT (delta from CT). Best improvement over CT bolded.

6 Conclusion

We interpret the general-domain performance drop experienced during continued training as catastrophic forgetting of general-domain knowledge and demonstrate that it can be largely mitigated by applying Elastic Weight Consolidation.

We present the full trade-off for in- and general-domain performance and show that our method outperforms MCL (Dakwale and Monz, 2017) at all operating points in five of six language pairs. Our method is able to regain the majority of the general-domain performance lost during continued training without compromising in-domain performance and without an additional memory or computational burden at translation-time.

Our method retains the advantages of continued training while addressing one of its main shortcomings and can be used in practical situations to avoid poor performance when general-domain input is encountered, even when in-domain performance and translation efficiency are both critical.

Acknowledgments

The authors thank Paul McNamee, Matt Post, Zach Wood-Doughty, and the Johns Hopkins 2018 SCALE participants for for helpful discussions and technical assistance.

Brian Thompson is supported by the Department of Defense through the National Defense Science and Engineering Graduate Fellowship (NDSEG) Program. Jeremy Gwinnup received support from the Air Force Office of Scientific Research (AFOSR) Visiting Scientist Program. This work has been partially supported by the DARPA LORELEI and the IARPA MATERIAL programs.

[6] Previous work has compared single runs of competing methods, making comparison difficult (e.g. one system may be better on in-domain, the other better on general-domain).

References

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, and Marco Turchi. 2017. Findings of the 2017 conference on machine translation (WMT17). In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 169–214, Copenhagen, Denmark. Association for Computational Linguistics.

Patrick Cadwell, Sharon OBrien, and Carlos SC Teixeira. 2018. Resistance and accommodation: factors for the (non-) adoption of machine translation among professional translators. *Perspectives*, 26(3):301–321.

Chenhui Chu, Raj Dabre, and Sadao Kurohashi. 2017. An empirical comparison of domain adaptation methods for neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 385–391. Association for Computational Linguistics.

Praveen Dakwale and Christof Monz. 2017. Finetuning for neural machine translation with limited degradation across in-and out-of-domain data. *Proceedings of the XVI Machine Translation Summit*, page 117.

Markus Freitag and Yaser Al-Onaizan. 2016. Fast Domain Adaptation for Neural Machine Translation. *CoRR*, abs/1612.06897.

Ian J Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. 2013. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv preprint arXiv:1312.6211*.

Felix Hieber, Tobias Domhan, Michael Denkowski, David Vilar, Artem Sokolov, Ann Clifton, and Matt Post. 2017. Sockeye: A toolkit for neural machine translation. *arXiv preprint arXiv:1712.05690*.

Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. In *NeurIPS Deep Learning and Representation Learning Workshop*.

Marcin Junczys-Dowmunt, Bruno Pouliquen, and Christophe Mazenc. 2016. COPPA V2.0: Corpus of Parallel Patent Applications building large parallel corpora with GNU make. In *4th Workshop on Challenges in the Management of Large Corpora Workshop Programme*.

Huda Khayrallah, Brian Thompson, Kevin Duh, and Philipp Koehn. 2018. Regularized training objective for continued training for domain adaptation in neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 36–44. Association for Computational Linguistics.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, page 201611835.

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. 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*, Prague, Czech Republic. Association for Computational Linguistics.

Philipp Koehn and Rebecca Knowles. 2017. Six Challenges for Neural Machine Translation. In *Proceedings of the 1st Workshop on Neural Machine Translation (and Generation) at ACL*. Association for Computational Linguistics.

Pierre Lison, Jörg Tiedemann, and Milen Kouylekov. 2018. OpenSubtitles2018: Statistical Rescoring of Sentence Alignments in Large, Noisy Parallel Corpora. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Minh-Thang Luong and Christopher D. Manning. 2015. Stanford Neural Machine Translation Systems for Spoken Language Domain. In *International Workshop on Spoken Language Translation*, Da Nang, Vietnam.

David J. C. MacKay. 1992. A practical Bayesian framework for backpropagation networks. *Neural Computation*, 4(3):448–472.

James Martens. 2014. New insights and perspectives on the natural gradient method. *arXiv preprint arXiv:1412.1193*.

Opinions, interpretations, conclusions and recommendations are those of the authors and are not necessarily endorsed by the United States Government. Cleared for public release on 14 Jan 2019. Originator reference number RH-19-119318. Case number 88ABW-2019-0136.

Antonio Valerio Miceli Barone, Barry Haddow, Ulrich Germann, and Rico Sennrich. 2017. Regularization techniques for fine-tuning in neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1489–1494, Copenhagen, Denmark. Association for Computational Linguistics.

Razvan Pascanu and Yoshua Bengio. 2013. Revisiting natural gradient for deep networks. *arXiv preprint arXiv:1301.3584*.

Victoria Porro Rodriguez, Lucia Morado Vazquez, and Pierrette Bouillon. 2017. Study on the use of machine translation and post-editing in swiss-based language service providers. *Parallèles*, (29 (2)).

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*.

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*, Berlin, Germany. Association for Computational Linguistics.

Brian Thompson, Huda Khayrallah, Antonios Anastasopoulos, Arya D. McCarthy, Kevin Duh, Rebecca Marvin, Paul McNamee, Jeremy Gwinnup, Tim Anderson, and Philipp Koehn. 2018. Freezing subnetworks to analyze domain adaptation in neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 124–132, Belgium, Brussels. Association for Computational Linguistics.

David Vilar. 2018. Learning hidden unit contribution for adapting neural machine translation models. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, volume 2, pages 500–505.