Few-shot learning through contextual data augmentation

Farid Arthaud, Rachel Bawden, Alexandra Birch

Few-shot learning through contextual data augmentation

Farid Arthaud *
Ecole Normale Supérieure
farid.arthaud@ens.psl.eu

Rachel Bawden *
Inria
rachel.bawden@inria.fr

Alexandra Birch
University of Edinburgh
A.Birch@ed.ac.uk

Abstract

Machine translation (MT) models used in industries with constantly changing topics, such as translation or news agencies, need to adapt to new data to maintain their performance over time. Our aim is to teach a pre-trained MT model to translate previously unseen words accurately, based on very few examples. We propose (i) an experimental setup allowing us to simulate novel vocabulary appearing in human-submitted translations, and (ii) corresponding evaluation metrics to compare our approaches. We extend a data augmentation approach using a pre-trained language model to create training examples with similar contexts for novel words. We compare different fine-tuning and data augmentation approaches and show that adaptation on the scale of one to five examples is possible. Combining data augmentation with randomly selected training sentences leads to the highest BLEU score and accuracy improvements. Impressively, with only 1 to 5 examples, our model reports better accuracy scores than a reference system trained with on average 313 parallel examples.

1 Introduction

News agencies dealing with articles in multiple languages often rely on machine translation (MT) to provide an initial translation, which a journalist reworks into a final article. This can involve changing the structure or phrasing, but also correcting (or *post-editing*) mistranslated words or expressions, which can frequently occur when new topics emerge, bringing new vocabulary that has been rarely or never seen in the data used for training. The willingness of a journalist to use MT technology is dependent on the general quality of the models, but also on whether they can learn from the journalist's corrections, to avoid them having to correct the same errors time and time again.

Various strategies have been explored to learn from a journalist's post-edits. One option is to use an automatic post-editing (APE) model trained on the journalist's post-edits. However, state-of-the-art APE systems (Junczys-Dowmunt and Grundkiewicz, 2017) typically require large numbers of post-edits for training, which are rarely available or hard to generate (particularly for low-resource languages). An alternative, commonly used strategy is to fine-tune models to in-domain data, but this is prone to overfitting (Miceli Barone et al., 2017). More advanced ways of continually learning through fine-tuning have been explored, selecting similar training instances based on their similarity with test sentences (Li et al., 2018; Turchi et al., 2017). These methods achieve good results according to automatic MT metrics but can also overfit when training is continued. So far there has been little focus on the speed of adaptation required—so as to minimize the number of human interventions required—and the trade-off between successfully adapting to specific new post-edits and maintaining a good global translation quality.

In this article, we aim to explore in more depth this trade-off between overall translation quality and the ability of the MT model to learn specific novel words in the context of life-long learning of MT from journalistic post-edits. We explore the setup in a simulated few-shot learning scenario, whereby we track the translation performance on specific test words that are removed from the training data and are gradually re-introduced through fine-tuning. To improve the model's capacity to learn these new examples, we explore an approach similar to Turchi et al. (2017) and inspired by a contextual data augmentation technique used for rare word translation (Fadaee et al., 2017) to reduce the number of corrections that need to be seen.

We show in our experiments on the Gujarati-English language pair that it is not only possible

*Work done at University of Edinburgh

to surpass the accuracy of our baseline fine-tuning approach, but also of a reference model which has already seen the new words dozens to hundreds of times during training. However we find that in most cases adaptation to new words comes at a variable cost in BLEU (Papineni et al., 2002), due to overfitting to the new examples. We show that this cost can be kept at a minimum by padding our data-augmentation approach with randomly selected sentences from our training set. The appropriate choice of hyper-parameters is also important for final performance.

Our code is freely available online.[1]

2 Related work

The topic of few-shot learning from post-edits is relatively novel, and we were therefore left with few comparison points. A somewhat similar task that requires quick adaptation is low-resource MT, for which transfer learning (Zoph et al., 2016) and meta-learning (Gu et al., 2018) approaches exist. These techniques generally apply for adaptation from hundreds of thousands of sentences, rather than a dozen available in our scenario; this is because we aim to learn individual new words rather than a whole language or domain.

A widespread technique in the MT literature to adapt a model to new data is fine-tuning, often used for domain adaptation. Turchi et al. (2017) and Li et al. (2018) explore the use of a similarity search in the training corpus in order to fine-tune an MT model before translating a novel sentence or to gradually adapt a model to post-edits. This approach does not apply to our scenario, where new words appear and therefore a similarity search cannot yield sentences containing these new words. Our baseline approach (which we call `finetune`) is present in these works, known as *adaptation a posteriori* in (Turchi et al., 2017) and also appears as *single-sentence adaptation* in (Kothur et al., 2018). A challenge these works report with fine-tuning is overfitting, which we encounter systematically when evaluating our various techniques. Works on fine-tuning also explore several regularization techniques (Simianer et al., 2019) when adapting to new data, which we choose to leave out of our comparison due to added hyper-parameter choices and complexity of implementation in our experiments – we do however believe that future work implementing these techniques could potentially outperform ours.

Fadaee et al. (2017); Kobayashi (2018); Wu et al. (2019) and Gao et al. (2019) explore a similar contextual data augmentation technique, albeit in different scenarios and with different goals. This technique synthesizes new sentences by using sentences from the training set and substituting different words into them. In (Fadaee et al., 2017) the goal is to enhance overall translation performance by focusing on words that appear rarely in the training data, but in our case we are training our system to learn new words which were not in the training set at all, based simply on a ground-truth translation of this new word by a human. Moreover, our technique uses more recent tools and techniques such as the BERT contextual language model (Devlin et al., 2019). Kobayashi (2018) and Wu et al. (2019) also use BERT for contextual data augmentation, but with a goal of improving language model tasks such as sentiment analysis. The constraints for this task are very different; rather than having to produce a translation for augmented data, these approaches have to maintain the sentiment label of the sentences. Gao et al. (2019) work in a similar context to us, but focus on overall translation performance rather than learning new words, and apply contextual data augmentation during the training step, thus removing the challenge of adapting to new data as it becomes available.

Similar to us, Huck et al. (2019) focus on improving the MT of words which are unseen in the training set. They use bilingual lexicons to hypothesize translations for their unseen terms. They find these translations in monolingual target side data and backtranslate them inserting the unseen term. They show that this improves translation performance in the medical domain. However they do not analyze the accuracy of translation of the novel terms, or explore how fast you can learn from very few examples.

3 Lifelong learning from post-edits

MT models are inevitably adapted towards the topics and vocabulary from the time period associated with their training data. In the long term, they therefore struggle to correctly translate novel words and expressions associated with new topics, unless they can be adapted to them. This is particularly a problem in journalism, where current topics and names are constantly changing. A prime example is the

[1] https://gitlab.com/farid-fari/fewshot-learning

recent COVID-19 pandemic: prior to January 2020, newspapers would contain little to no mention of the words *coronavirus*, *respirators*, *PPE masks* and *hydroxychloroquine*. On top of these topic-specific words, there may also be novel but very frequent expressions that are initially hard to translate, such as *flattening the curve* and *social distancing*, which are likely to be poorly translated and therefore need to be subsequently corrected by the journalist.

We are interested in developing approaches to quickly learn from journalistic post-edits in a way that maintains the general high translation quality of the model. In order to analyze and objectively evaluate our different approaches, we simulate this scenario in a reproducible way by using a publicly available corpus in which we select the rarest words and separate out all sentences containing them: these words will be our *evaluation words*. These words must appear at least a few times in the test set as well in order to enable proper evaluation of adaptation using our approaches. The sentences separated from the training set containing our evaluation words will be used to simulate reference sentences submitted by a journalist for our models to use for adaptation. Finally, the training set with the evaluation words removed will be our *filtered training set*, used for initial training of our model.

Our test set is also made up of publicly available data sets whose sentences also contain the evaluation words we selected. This setup prohibits the use of a development set and a test set due to the dependency of the rare words on the test set: they must appear a minimum number of times in the test set to have a way of evaluating our approaches. In turn, the choice of rare words changes the training set and therefore the models we train. We are therefore in a transductive learning scenario, where our method is adapted to the task we aim to solve.

We also use as a reference point a model which has been trained on the complete (unfiltered) training set, which is a very strong comparison point since it has seen many more occurrences of the rare words we selected than our few-shot models.

4 Our approaches

We propose and evaluate four approaches: `finetune` (our baseline), `randompad`, `augmented` and `half`. The first two approaches only involve fine-tuning whereas the latter two include contextual data augmentation. Each approach works from a given number of reference sentence pairs containing our evaluation words, acting as human-provided references. In our experimental setup, these sentences are chosen randomly among the held-out sentences from the filtered training set.

4.1 Baseline: fine-tuning

Our first approach, which is also our baseline, is naive fine-tuning referred to as *adaptation a posteriori* in (Turchi et al., 2017), and which we refer to as `finetune`. When presented with a set of reference sentence pairs to adapt to, our model is trained from its previous state for a few epochs using the sentence pairs. This approach is especially prone to overfitting as it will see very few sentences, and they all contain a potentially repetitive set of words. To minimize this effect, we choose to fine-tune on a batch of several reference sentence pairs rather than individually for each evaluation word, so as to diversify the data fed to the model.

4.2 Padded fine-tuning

The second approach, `randompad`, attempts to minimize the overfitting from `finetune` by introducing variety and generality to the fine-tuning set. This is done by adding randomly chosen sentences from the filtered training set to the reference sentences, which is known as *padding*. These random sentences do not contain the evaluation words and so encourage generalization during fine-tuning.

4.3 Contextual data augmentation

The aim of this approach, similar to the one used by Fadaee et al. (2017) and which we refer to as `augmented`, is to use the reference sentences provided by the journalist to create additional novel sentences containing the evaluation words. If successful, this approach simulates the presence of more reference sentences containing the new words, which accelerates learning, effectively reducing the number of corrections required from the human translator or journalist. More formally, given a reference sentence pair (s_{src}, s_{tgt}) containing an unseen word $w \in s_{tgt}$, the goal is to generate new sentence pairs containing w in the target language, varied enough to be beneficial in fine-tuning.

Our data augmentation process has 2 steps: (i) for each reference sentence pair provided by the journalist containing a word of interest w, find sentences from the training corpus with similar contexts to the one surrounding w, and then (ii) insert w and its translation into the retrieved sentences.

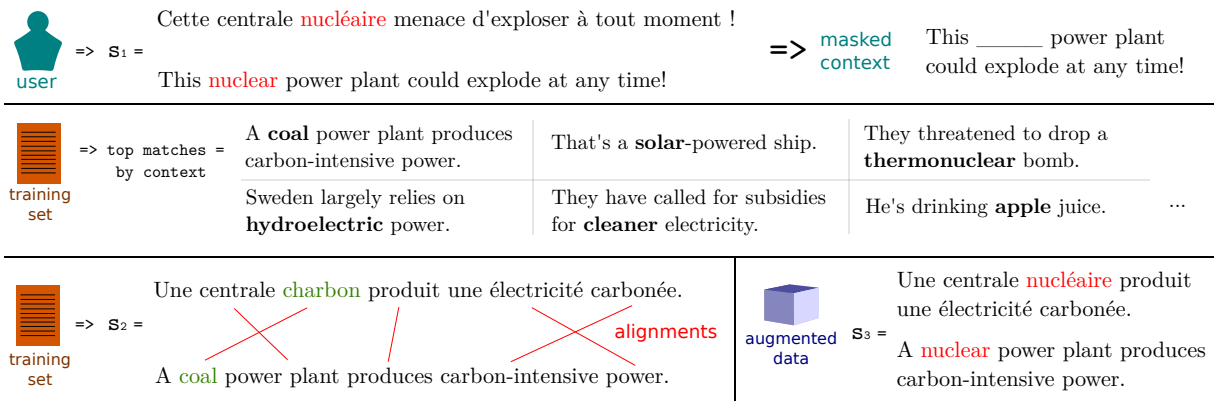

Figure 1: Contextual data augmentation applied to an example sentence

We illustrate this process in Figure 1, using the French-English language pair to aid readability: the sentence pair s_1 containing the new word *nuclear* is provided to our system by the journalist or translator. We first find sentences in the training set that contain a word (in bold) with a similar context to that of w in s_{tgt}. In each of these sentences, the word in bold could be replaced by *nuclear*, although there are sometimes false positives such as the last sentence: *He's drinking* ***nuclear*** *juice*, would not be a good reference sentence for training. Next, we use an alignment tool to replace the aligned source word (e.g. *charbon*) in the selected sentence pair s_2 with the word *nucléaire*, providing a brand new sentence pair s_3 for the reference word *nucléaire-nuclear*.

Finding similar contexts The first step is to find suitable sentences in which we can substitute w and its translation. As in (Wu et al., 2019), we use the BERT contextual language model (Devlin et al., 2019) to provide a contextual representation of w in s_{tgt}, noted v. This feature vector is taken from the second-to-last layer of BERT (usually used to compute vocabulary probabilities through a softmax layer). Given the masked and bidirectional nature of BERT, this contextual representation contains information about the context surrounding w in s_{tgt}, with no prior knowledge of w required by BERT. The advantage of this approach is that truly novel words can be data-augmented using this technique with any pre-trained BERT model.

The context search consists in (i) extracting the feature vector v for w in s_{tgt}, (ii) for each sentence in the filtered training set, randomly selecting a word u in it and computing the corresponding feature vector[2] and (iii) selecting the top k sentences based on the cosine similarity of these feature vectors with v.

This is shown in Figure 1 where 6 sentences have been selected based on their feature vector similarity to s_1. These sentences are selected because the context surrounding the random word chosen in each of them is similar to the context surrounding w in s_1, which generally means the bold words loosely correspond to adjectives describing power plants. Using a masked model means that knowledge of the meaning of w is not required, but only an understanding of the context in s_{tgt}. An additional real use example is provided in Appendix B.

This process significantly differs from Fadaee et al. (2017), which iterates over the training set to find sentences for which the language model gives a high probability of the rare words appearing. Moreover, by using feature vectors rather than raw probability distributions (as in previous works), we capture richer information about the surrounding sentence than only which words are likely to replace w.

Creation of new training examples The second step is to substitute the randomly masked word u—which is *coal* in our example—in the retained sentence pair s_2 with w in the target language and its translation in the source language. In the target language, we simply replace u by w, but the task is harder in the source language since we have no prior information as to which word translates to u, or which word translates to w to replace it with. In our scenario, we assume that the translation in the source language of w, noted w', is known to us – for

[2]Comparing all positions in all sentences would lead to the best results, but is inefficient. With sufficient training examples (as is the case here), randomly choosing a single position in each sentence avoids having to to do all computations and still provides varied contexts that can be tested.

example, a human translator providing (s_{src}, s_{tgt}) could identify w' in s_{src}. Having no human translator for our experiments, we train an alignment tool on the complete unfiltered training set and use it to select the words most often aligned to each w from our evaluation words. A different approach could be used in future work, by training the alignment tool on the filtered training set and then re-training it on reference sentences as they are provided, removing the need for human intervention or prior knowledge of w. Note that all alignment steps rely on automatic tools which can introduce some noise in the process. However, we observed satisfactory results, and this allows for the entire process to be automated.

Once the translation w' for w is established, we have to determine which word or words in the source sentence must be replaced by w'. This is done by using an alignment tool trained only on the filtered training set, to compute the aligned word to u. The word aligned to u is replaced by w'; if multiple consecutive words are aligned to u they are all replaced with w', and if no words or non-consecutive words are aligned to u then the sentence is discarded. This is illustrated in Figure 1, where *coal* is successfully aligned to *charbon*, allowing us to replace it with *nucléaire*.

The final result in the example is s_3, a sentence pair which makes sense and will be useful for the translation of *nucléaire* to *nuclear*. The sentences generated by this method will at the very least contain w' and w in roughly aligned source and target positions and the best sentences will have the same quality as human-provided references.

4.4 Padded data augmentation

The `augmented` approach does not benefit from the added generality of the `randompad` approach; it could suffer from overfitting due to the repetition of our evaluation words (and their semantic fields) in the fine-tuning sentences. This is addressed by the `half` approach, designed to overcome potential overfitting in the `augmented` approach. This approach uses half of the synthesized sentences from `augmented` and replaces the other half with random sentences from the filtered training set. This provides more sentences containing our rare words through augmentation but also more generality from the random sentences, aiming to combine the strengths of both approaches.

5 Experiments

5.1 Experimental setup

We use the low-resource Gujarati to English language direction. Our base model is the same as the University of Edinburgh's submission to WMT 2019 (Bawden et al., 2019), except that its training set is filtered to remove sentences containing words previously selected for evaluation. The model is a base transformer model (Vaswani et al., 2017) with 6 encoder and decoder layers, feed-forward dimension 512, 8 transformer heads, and dropout of 0.1. We train using the Marian toolkit (Junczys-Dowmunt et al., 2018) and the Adam optimizer (Kingma and Ba, 2015).[3] The complete training script with additional parameters can be found on the webpage for the University of Edinburgh's WMT19 submission.[4] We use an identical version of this model trained on the full, unfiltered training set as a reference point called `all-data`.

The main preprocessing, data augmentation and training scripts are freely available online: `https://gitlab.com/farid-fari/fewshot-learning`.

We use the pre-trained BERT model from Huggingface (Wolf et al., 2020): the `bert-large-uncased-whole-word-masking` variant, which has the benefit for our use case of masking whole words rather than subwords.

5.2 Data and preprocessing

Our training data consists of both genuine Gujarati-English parallel data and backtranslations from the WMT19 news translation task (Barrault et al., 2019). The backtranslations are produced as described by Bawden et al. (2019) and we follow their method of first training on a mixture of backtranslations and parallel data before fine-tuning on the genuine parallel data only. As mentioned previously, we filter out English words selected for evaluation from both the synthetic and genuine parallel data. However, only the genuine parallel data is used to select augmentation candidates to ensure their high quality. For testing, we use the concatenation of `newstest2019` and `newsdev2019`.

Preprocessing All data augmentation was run after tokenization but before subword segmentation to keep a consistent notion of a 'word'. We

[3] We experimented with SGD but observed no significant differences with Adam.

[4] `http://data.statmt.org/wmt19_systems/en-gu/train.sh`

first apply tokenization using the Moses tokenizer (Koehn et al., 2007) and then apply sub-word segmentation using the BPE strategy (Sennrich et al., 2016) and the `fastbpe` implementation.[5,6] Word alignment is carried out using `fast_align` (Dyer et al., 2013).[7]

Evaluation data The rare words to filter out are chosen to be the 100 rarest words in the training set that appear at least 5 times in the test set and 20 times in the training set, and are manually filtered down to 96 words to exclude low quality choices (such as plurals and punctuation). Their frequency in the unfiltered training set (used by our reference model) ranges from 20 to 775 occurrences, with a mean of 313 and a median of 275 occurrences. The complete word list can be found in Appendix A. These words appear in 701 of the 3,014 test set sentences, meaning that the BLEU score is computed on a majority of sentences not containing our evaluation words, thus giving us a good overview of how general translation performance is affected. They appear in only 26,910 of the 8.5M training sentences,[8] meaning that the filtered training set is almost the same size as the original full training data. The genuine parallel data contains 40k sentences which were used for context search in the data augmentation steps.

Evaluation setup Our aim is to evaluate how well the model adapts (through fine-tuning of the model) to the gradual addition of reference examples containing the 96 evaluation words, which were absent from the initial training data. For each evaluation word, we randomly choose 20 reference sentences that contain the word, and for each of the four approaches, we successively make 1, 2, 3, 5, 10, 15 and 20 reference occurrences of each word available to the model to learn from, evaluating at each step. Everything is run as a batch over all 96 words; for instance in an experiment using 3 occurrences, fine-tuning is conducted over 288 reference sentences, with each reference word occurring 3 times.[9]

Each approach that uses padding and/or augmentation requires choosing a ratio r between the total number of fine-tuning sentences and the original number of reference sentences provided by the journalist. For a word w, as the number of occurrences offered to the model grows from 1 to 20, the number of fine-tuning sentences grows from r to $20r$. This ratio r is calculated as follows:

$$n_{total} = n_{ref} + n_{synth} + n_{rand} \quad (1)$$

$$r = \frac{n_{total}}{n_{ref}}, \quad (2)$$

where n_{ref}, n_{synth} and n_{rand} refer respectively to the number of reference, synthetic and random sentences. The chosen ratio is written in brackets following the name of each approach.

For the `augmented` experiments we choose $r = 20$, meaning that we augment each reference sentence with 19 synthetic sentences. This is chosen by manually evaluating the quality of augmented sentences on examples, as shown in Appendix B. We run two `randompad` experiments, `randompad(2)` (half of sentences are random), and `randompad(20)` (same ratio as `augmented(20)` for comparison). The `half(20)` experiment has the same ratio as `augmented(20)`, with each reference sentence accompanied by 9 random sentences and 10 synthetic sentences.[10]

5.3 Evaluation metrics

We use two evaluation metrics: (i) the BLEU score (Papineni et al., 2002) computed with the `multi-bleu-detok` script from the Moses toolkit (Koehn et al., 2007) on the entire test set to evaluate overall MT quality, and (ii) the accuracy of the filtered evaluation words to evaluate how well the approaches learn from the post-edits. We use a clipped bag-of-words accuracy defined as follows: if in the target language the reference sentence contains the word w n times, and the translated sentence contains it p times, then the accuracy is $\frac{\min(p,n)}{n}$. This accuracy is computed separately for each of the evaluation words w, and then averaged to obtain an overall accuracy. The advantage of this metric is that it also only requires segmentation to be done in the target language, meaning that once more no prior knowledge or training fur-

[5] https://github.com/glample/fastBPE

[6] We reuse the pre-processing scripts available at http://data.statmt.org/wmt19_systems/scripts/.

[7] https://github.com/clab/fast_align

[8] Both genuine and synthetic parallel training sentences.

[9] The number of sentences is very slightly lower due to some sentences containing 2 evaluation words, but this is rare enough that it does not meaningfully impact our results.

[10] We leave it to future work to explore additional padding and augmentation ratios. In the `half` experiments, we randomly select half of the sentences from the synthetic sentences from `augmented`: this means in particular that we do not necessarily keep the best half of the synthetic sentences, leaving room for improvement in future experiments.

ther than the base model are required for the source language.

A third evaluation metric, which is very important for our scenario but overlooked in previous work, is the speed at which the model improves its accuracy. We define this as the number of reference sentences per novel word it needs to see in order to substantially improve the accuracy in translating these words. For a journalist correcting an MT system's mistakes, it is important to correct a given mistake as few times as possible, since having to correct each mistake beyond a certain number of times might make it more worthwhile to simply manually translate the article. The best way of evaluating this metric is by comparing the evolution of the accuracy curve and the BLEU curve as a function of the number of seen occurrences, as presented in the next section in more detail.

We also compare all models to the reference model `all-data`, trained on the unfiltered training set, as in the University of Edinburgh's WMT19 submission (Bawden et al., 2019). This model is considered as a topline rather than a baseline since it does not perform the same task and was trained with much more data – our baseline being the `finetune` approach.

5.4 Hyper-parameter choice

To ensure optimal settings, for each approach we choose hyper-parameters based on the size of the fine-tuning corpus. Several hyper-parameter choices seem relevant, depending on the goal: more training leads to lower BLEU scores (as overfitting occurs) but higher accuracies on the evaluation words, exposing a trade-off between BLEU and translation performance specifically on the evaluation words. This is shown in Appendix D where we explore different values for the `finetune` approach. In our results in the following section, we use hyper-parameter values that best match the accuracy scores for all approaches, thus simplifying the trade-off to a direct comparison in BLEU score.

Choosing a single epoch and learning rate value depends on the end goal: we can focus on accuracy at all costs, even if it means decreasing overall translation performance, or take a more conservative approach by moderately increasing accuracy while maintaining BLEU. Another way of seeing this trade-off is with respect to a time scale over which adaptation occurs: if the process is to be repeated many times, then it may be wise to decrease BLEU as little as possible, whereas a system that is often reset or used less often can afford to sacrifice more BLEU for extra accuracy on novel words. It is important to note that these results are batched, so the BLEU losses or gains illustrated correspond to the learning of 96 words at once.

We present our results with two adaptation speeds for each approach: a slow (*S*) and a fast (*F*) setting, corresponding to two possible compromises in the previously explained trade-off. At each adaptation speed, for each approach, different hyper-parameters are used in order to match accuracy scores as closely as possible to make their comparison easier. The table of chosen epoch and learning rate values can be found in Appendix C.

6 Results

Figures 2 and 3 present the results for the reference model and all approaches for both slow and fast adaptation speeds. We immediately notice that at both adaptation speeds our models are capable of surpassing the `all-data` model[11] in terms of accuracy on the new words, despite having seen fewer than 20 (and as few as 1) reference sentences containing those words, whereas `all-data` has seen each evaluation word over 300 times on average. Our baseline, the `finetune` model, is shown to be largely surpassed by all other approaches in accuracy, both on the slow and fast settings.

With hyper-parameters being chosen to approximately match accuracy curves, the main comparison point is the BLEU score. In Figure 2, we see that in the slow setting our `half` approach is the only one able to improve the BLEU score while learning the new words, nearly matching `all-data` in both BLEU and accuracy at 3 occurrences seen. The `augmented` approach offers a slightly better accuracy curve than all other models on this speed, but loses out on BLEU score at higher occurrence numbers.

It is very important to look at the first points on the curve: 1-5 occurrences of each evaluation word is the realistic range to imagine a journalist making corrections, since a journalist could become frustrated with a model requiring each new word to be corrected up to 20 times for it to be correctly translated. In this respect, Figure 3 for the fast setting offers the best results, with accu-

[11] Note however that `all-data` does not constitute a baseline for our model: it does not perform the same task and has access to many more reference sentence pairs.

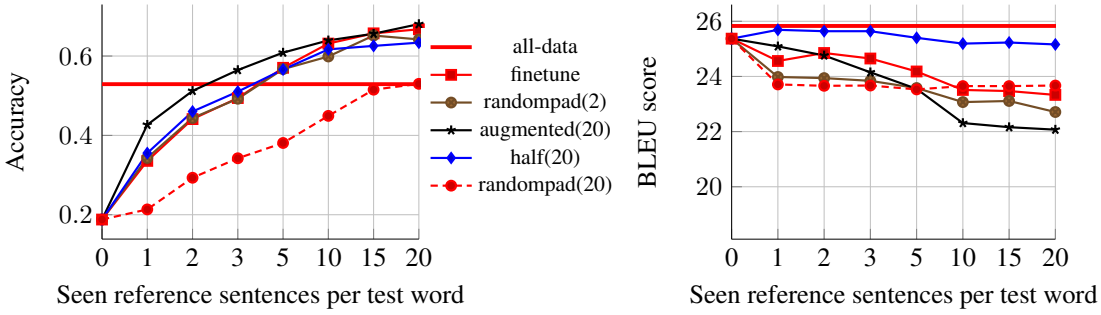

Figure 2: BLEU and accuracy results of all of our approaches for the slow speed setting

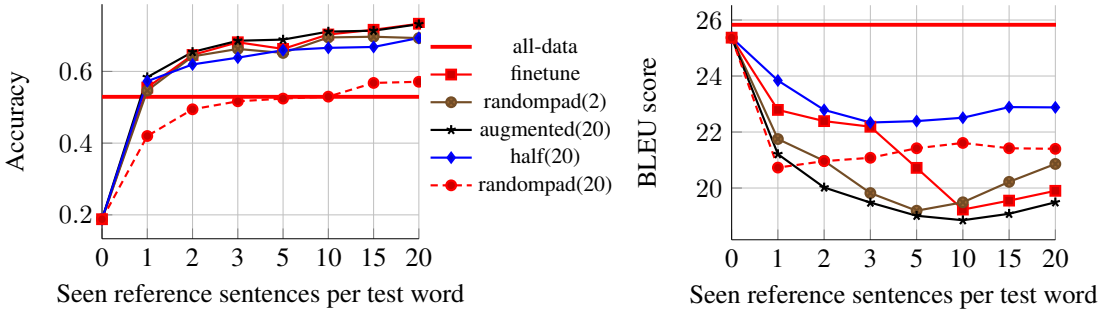

Figure 3: BLEU and accuracy results of all of our approaches for the fast speed setting

racy immediately surpassing `all-data` for all but one model, albeit at a heightened cost to BLEU. Several models, including our new `augmented` and `half` approaches, see an increase in BLEU at higher occurrence numbers, suggesting that generalization is occurring as more sentences are learned with bigger learning steps.

The `randompad` approaches surprised us in two ways, especially given `half`'s success: they generally perform worse than `finetune` despite being designed as an improvement over it, and `randompad(20)` has very poor accuracy scores compared to all other approaches. To the first point, one possible explanation is that overfitting might occur with both the random sentences and the reference sentences when few data is used for fine-tuning (low seen occurrence numbers). This would explain why in Figure 3 the BLEU curve for `randompad` ends up overtaking the `finetune` curve once enough padding data is available to allow generalization. The second point can be explained by a form of 'dilution' of the reference sentences containing the evaluation words: the model over-adapts to the sentences provided to it but does not particularly improve on the evaluation words. We also tried various hyper-parameters for `randompad(20)` but were unable to find a compromise similar to other models, resulting in this approach standing out from the others. This might also be partially explained through variations inherent to the randomness in the selection of the padding sentences.

6.1 Analysis

To gain more insight into which words were translated well or poorly, we chose to look at our accuracy metric for each evaluation word. For the fast `half` approach, averaged across all occurrence numbers, the best words were generally proper nouns such as *Sulawesi* (1.00), *Isabel* (0.98) and *Kohli* (0.68). Acronyms such as *ATM* (1.00), *RCN* (0.98) or *GB* (0.87) also performed very well, and common nouns or verbs had more varied performance: *niece* (0.00), *moustache* (0.37) and *smartphone* (0.87). Although the very best words are generally proper nouns or acronyms and the very worst generally other parts of speech, no clear pattern or general rule can be ascertained.

One phenomenon initially worrying us was the over-translation of evaluation words: a model outputting evaluation words where they should not appear would be able to 'trick' our bag-of-words accuracy metric by artificially inflating it to a cer-

tain point. However, we hypothesized and then verified that this would be counteracted by a decrease in BLEU score due to these words appearing in sentences where they should not. This is presented in Appendix E where we confirm this by demonstrating a negative correlation between over-translation and BLEU score.

7 Conclusion

We explored different techniques based on fine-tuning in order to adapt a base model to post-edits containing novel vocabulary. We proposed a data augmentation technique never applied to this task, allowing us to expand the number of occurrences of the new words available to our model to learn from. In our experiments, all proposed adaptation techniques offer better performance on the novel words than our reference model, which had seen the words hundreds of times each. Our data augmentation approaches yield faster adaptation than our baseline, but with a greatly improved BLEU score, especially when combined with generalization using padding with random training set data.

These various techniques could all be applied to lifelong adaptation of an MT system often confronted with new vocabulary or expressions. Our work can be generalized in several directions which we chose to leave for future work: word translations can automatically be retrieved with alignments, rare multi-word expressions (*n-grams*) can be used rather than rare single-token words, the language model generalization can be used in the source language or both languages and several word substitutions can be made in a single sentence. Several approaches can be deployed to further improve the BLEU score when fine-tuning, such as regularization techniques as explored by Simianer et al. (2019), keeping only the best half of synthetic sentences rather than a random half, and a more careful choice of the ratio of augmented and random data relative to reference sentences.

Acknowledgments

This work was supported by funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825299 (GoURMET) and funding by the UK Engineering and Physical Sciences Research Council (EPSRC) fellowship grant EP/S001271/1 (MTStretch).

References

Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. 2019. Findings of the 2019 conference on machine translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61, Florence, Italy. Association for Computational Linguistics.

Rachel Bawden, Nikolay Bogoychev, Ulrich Germann, Roman Grundkiewicz, Faheem Kirefu, Antonio Valerio Miceli Barone, and Alexandra Birch. 2019. The university of Edinburgh's submissions to the WMT19 news translation task. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 103–115, Florence, Italy. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648, Atlanta, Georgia. Association for Computational Linguistics.

Marzieh Fadaee, Arianna Bisazza, and Christof Monz. 2017. Data augmentation for low-resource neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 567–573, Vancouver, Canada. Association for Computational Linguistics.

Fei Gao, Jinhua Zhu, Lijun Wu, Yingce Xia, Tao Qin, Xueqi Cheng, Wengang Zhou, and Tie-Yan Liu. 2019. Soft contextual data augmentation for neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5539–5544, Florence, Italy. Association for Computational Linguistics.

Jiatao Gu, Yong Wang, Yun Chen, Victor O. K. Li, and Kyunghyun Cho. 2018. Meta-learning for low-resource neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3622–3631, Brussels, Belgium. Association for Computational Linguistics.

Matthias Huck, Viktor Hangya, and Alexander Fraser. 2019. Better OOV translation with bilingual terminology mining. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5809–5815, Florence, Italy. Association for Computational Linguistics.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2017. The AMU-UEdin submission to the WMT 2017 shared task on automatic post-editing. In *Proceedings of the Second Conference on Machine Translation*, pages 639–646, Copenhagen, Denmark. Association for Computational Linguistics.

Marcin Junczys-Dowmunt, Kenneth Heafield, Hieu Hoang, Roman Grundkiewicz, and Anthony Aue. 2018. Marian: Cost-effective high-quality neural machine translation in C++. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 129–135, Melbourne, Australia. Association for Computational Linguistics.

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

Sosuke Kobayashi. 2018. Contextual augmentation: Data augmentation by words with paradigmatic relations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 452–457, New Orleans, Louisiana. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej 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*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Sachith Sri Ram Kothur, Rebecca Knowles, and Philipp Koehn. 2018. Document-level adaptation for neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 64–73, Melbourne, Australia. Association for Computational Linguistics.

Xiaoqing Li, Jiajun Zhang, and Chengqing Zong. 2018. One sentence one model for neural machine translation. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*, Miyazaki, Japan. European Languages Resources Association (ELRA).

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.

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

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Patrick Simianer, Joern Wuebker, and John DeNero. 2019. Measuring immediate adaptation performance for neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2038–2046, Minneapolis, Minnesota. Association for Computational Linguistics.

Marco Turchi, Matteo Negri, M. Amin Farajia, and Marcello Federico. 2017. Continuous Learning from Human Post-Edits for Neural Machine Translation. *The Prague Bulletin of Mathematical Linguistics*, 108:233–244.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Xing Wu, Shangwen Lv, Liangjun Zang, Jizhong Han, and Songlin Hu. 2019. Conditional bert contextual augmentation. In *Proceedings of the 19th International Conference on Computational Science*, pages 84–95, Faro, Portugal. Springer International Publishing.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. In *Proceedings of the*

2016 Conference on Empirical Methods in Natural Language Processing, pages 1568–1575, Austin, Texas. Association for Computational Linguistics.

A Filtered words

The complete list of the 96 filtered words in English is given in Table 4.

B Data augmentation example

This is an actual example of contextual data augmentation from a given sentence. The reference sentence is, "*A powerful 7.5 magnitude earthquake hit the Indonesian island of* ***Sulawesi*** *on Friday, September 29, triggering a tsunami and leaving nearly 400 people dead.*", with the novel word being *Sulawesi*. The top five sentences matching the context of *Sulawesi* above are as follows:

1. This labour shortage prompted the authorities to import slaves from **Indonesia** and Madagascar.
2. Many of them have settled down in Ahmedabad, Vadodara, Mumbai, Kolkota, Delhi, Nagpur and far away places like **Java**, Rangoon, Singapore, Fiji, Eden, Kenya, Uganda, America etc and established their business in these places.
3. The rice lands of **Java** are among the richest in the world.
4. Rising ocean temperatures and ocean acidification means that the capacity of the ocean carbon sink will gradually get weaker, giving rise to global concerns expressed in the **Monaco** and Manado Declarations.
5. **Lara**'s first school was St. Joseph's Roman Catholic primary.

These sentences generally capture the idea that the word refers to an island, except for the last one which might be an outlier. The fourth sentence is remarkable in the fact that *Monaco* is not an island, but the context of *rising ocean temperatures and ocean acidification* as well as the mention of *Manado* alongside (the capital of North Sulawesi) make the sentence relevant here. This highlights the importance of context besides the actual words' meanings.

C Learning rate and epochs

Table 5 gives the epoch numbers and learning rates we chose for our presented examples. The `augmented` and `half` experiments both have lower learning rates because they fine-tune on more data, and thus go through more gradient steps.

D Preliminary survey of hyper-parameters

We tried different hyper-parameter combinations for the `finetune` approach to explore the dependency of accuracy and BLEU score to these. Figure 6 shows these curves labeled by '*number of epochs* / *learning rate*'.

We noticed a clear trade-off between BLEU score and accuracy as the parameters evolved, with more training leading to bigger gains in accuracy at a cost to BLEU. Higher parameters saw diminishing returns as accuracy would improve little or not at all while BLEU kept decreasing, as can be seen for the two first curves. Lower parameters also showed diminishing returns in BLEU score as accuracy dramatically drops.

E Over-translation

Table 7 shows the over-translation metric for different approaches. Over-translation is defined as the number of times each word appears in sentences in excess of the reference sentence, divided by the number of times it appears in the reference. This metric is computed similarly to the accuracy metric as explained in Section 5.3, i.e. per word over all sentences and then averaged over all words.

For reference, the `all-data` model gets an over-translation metric of 0.04 and the base model (the one referred to as 0 occurrences in figures) scores 0.01.

While some approaches have seemingly high over-translation values, the most competitive approaches do not see such a high increase in over-translation. For instance, the over-translation metric of 0.25 for `half(20)` (F) means that an over-translated word would appear for every four reference occurrences of our evaluation words, which means that it would over-translate only in about one in seventeen sentences in the test set (given that reference words only appear in 23% of test set sentences). The `half` approach has very good BLEU scores, which is aligned with the fact that it has some of the best over-translation scores amongst the presented approaches.

Moreover, Figure 8 shows for all of these data points the evolution of BLEU score with over-translation: there is a clear linear correlation, implying that BLEU captures over-translation by our approaches very well.

2018	ATM	Ahmedabad	Ambani	Amul	Anand	Ayr
BJP	Bachchan	Becker	Bedford	Chequers	Constantinople	Conway
DM	Dinesh	Dragons	Fidelity	Fleetwood	GB	GST
Gadkari	Giga	HDFC	Hastings	Isabel	Jammu	Kapoor
Kavanaugh	Keyser	Kohli	Lavrov	Lina	Lucknow	MLA
Manish	Mayorga	Meng	Modi	Molinari	Mukesh	Musk
Márquez	Nana	Narendra	Nifty	Oldham	Palu	Patriarch
Patriarchate	Prithvi	Pune	RCN	RTI	Rajkot	Rupani
Rupee	Sachin	Salman	Scalia	Seeley	Sensex	Shetty
Shilpa	Spiegel	Sulawesi	Surat	Sushma	Tendulkar	Tesla
Tiwari	Twitter	Vadodara	Virat	Vyas	Watts	app
apps	cleanliness	crores	cylinders	dough	fortress	ghee
inaugurate	intoxicated	lakhs	litre	mentioning	moustache	niece
refrigerators	sacrificed	slab	smartphone	strawberries		

Figure 4: Filtered words from the corpus

		`finetune`	`randompad(2)`	`randompad(20)`	`augmented`	`half`
Slow (S)	Epochs	10	10	10	10	10
	Learning rate	4×10^{-5}	4×10^{-5}	1×10^{-5}	4×10^{-6}	4×10^{-6}
Fast (F)	Epochs	30	30	30	10	10
	Learning rate	1×10^{-4}	1×10^{-4}	4×10^{-5}	4×10^{-5}	4×10^{-5}

Figure 5: The chosen learning rate and epoch values in our experiments

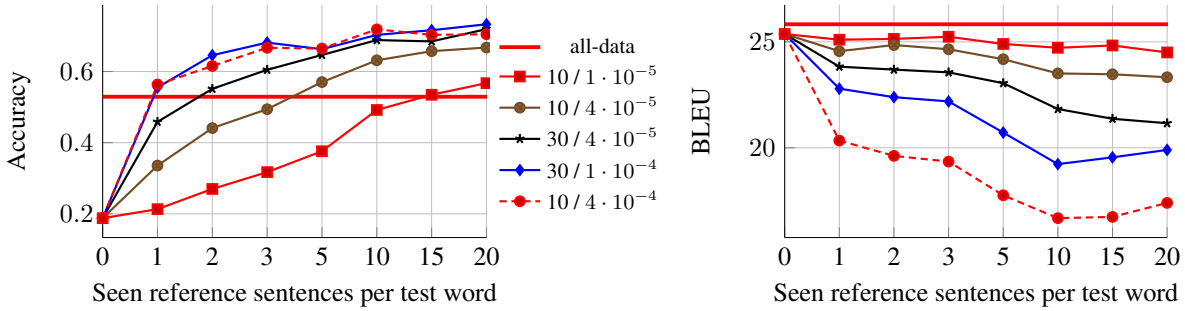

Figure 6: Evolution of the `finetune` approach's performance labeled by *epochs* / *learning rate*

Seen reference occurrences	1	2	3	5	10	15	20
`finetune` (S)	0.02	0.05	0.08	0.14	0.24	0.24	0.3
`finetune` (F)	0.16	0.33	0.5	0.59	0.85	0.82	0.78
`randompad(2)` (S)	0.03	0.06	0.09	0.13	0.23	0.23	0.25
`randompad(2)` (F)	0.17	0.33	0.48	0.48	0.55	0.6	0.52
`augmented(20)` (S)	0.06	0.1	0.12	0.19	0.26	0.29	0.32
`augmented(20)` (F)	0.32	0.4	0.47	0.57	0.68	0.73	0.76
`half(20)` (S)	0.03	0.07	0.08	0.1	0.15	0.15	0.17
`half(20)` (F)	0.15	0.22	0.24	0.26	0.26	0.28	0.25

Figure 7: The over-translation score of various approaches

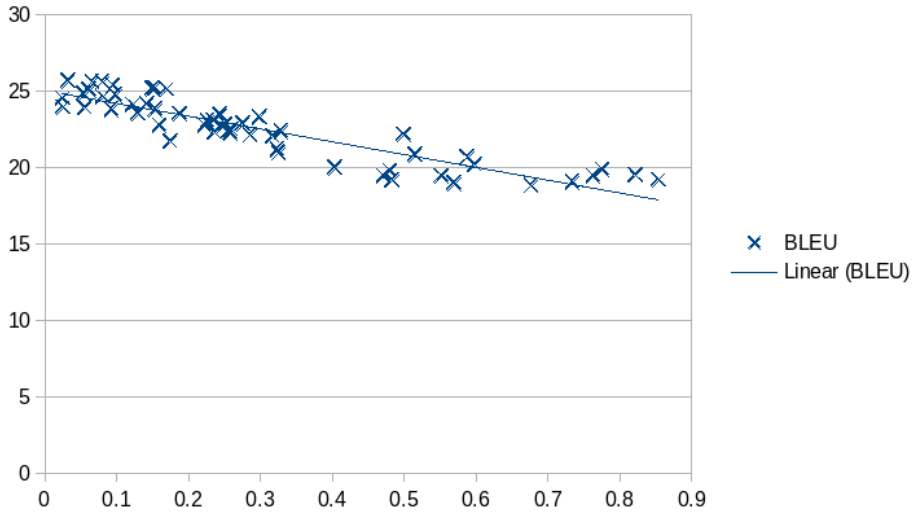

Figure 8: BLEU score as a function of over-translation score