Lexical Micro-adaptation for Neural Machine Translation

Jitao Xu, Josep Crego, Jean Senellart

SYSTRAN
5 rue Feydeau, 75002 Paris (France)
firstname.lastname@systrangroup.com

Abstract

This work is inspired by a typical machine translation industry scenario in which translators make use of in-domain data for facilitating translation of similar or repeating sentences. We introduce a generic framework applied at inference in which a subset of segment pairs are first extracted from training data according to their similarity to the input sentences. These segments are then used to dynamically update the parameters of a generic NMT network, thus performing a *lexical micro-adaptation*. Our approach demonstrates strong adaptation performance to new and existing datasets including pseudo in-domain data. We evaluate our approach on a heterogeneous English-French training dataset showing accuracy gains on all evaluated domains when compared to strong adaptation baselines.

1. Introduction

High-quality domain-specific translation is crucial for nowadays machine translation industry, which has adopted neural machine translation (NMT) as its dominating paradigm [1, 2, 3, 4, 5]. The data-driven nature of NMT conditions the quality of translations to the availability of large volumes of adequate training resources for a given domain. Despite that an ever increasing amount of data is becoming available, domain-specific resources are usually scarce and the most common approach when dealing with domain specific data is either to train multi-domain models [6] and dynamically provide a token to select the domain or to fine-tune a generic translation model by running additional learning iterations (fine-tuning) over in-domain data [7]. However, these solutions involving prior training of models are a) expensive on industrial production workflows, given the large number of specialised models usually required, resulting from the combination of genres, domains, styles, customer products, etc, b) impracticable and not flexible-enough to account for further fine-tuning to specific translators, covering very specific datasets [8, 9].

This work covers a typical industry scenario where high-quality translations are needed of repetitive material for which similar examples are available in the training dataset (*e.g.* translation of product manuals, drafts of legislation, technical documentation, *etc.*). Texts to be translated are relatively large and built from semantically related sentences. In this context translations are usually human post-edited, consequently, making available a valuable knowledge that could be incorporated by MT engines to further boost translation of upcoming documents. We suggest a simple but yet powerful micro-adaptation framework applied on-the-fly at inference-time in two steps: first, a reduced set of training instances similar to input sentences are retrieved from the training dataset; second, such retrieved sentences together with their corresponding human translations are used to update the parameters of a generic NMT network. These two steps being repeated for each new translation batch. A main drawback of our micro-adaptation framework is the additional workload when translating new documents. However, little overhead is introduced by sentence retrieval, which can be performed very efficiently, while adaptation can be speed up by reducing the number of training examples it considers.

The contributions of this paper are the following: we analyse the adaptation ability of our framework under reduced training data conditions and compare this ability to traditional adaption process; an extensive analysis of the different parameters of our framework is conducted; we evaluate the performance of different similarity measures to collect training examples related to input sentences; and we assess the ability of the presented framework to further improve performance using out-of-domain datasets.

We briefly introduce neural machine translation in Section 2. Section 3 presents the similarity measures considered in this work. Section 4 reports on the experiments conducted to evaluate the proposed framework. Section 5 outlines related work and finally, Section 6 draws conclusions and details further work.

2. Neural Machine Translation

Our micro-adaptation framework is built on top of the state-of-the-art Transformer model introduced by [5]. A neural network following the encoder-decoder architecture, where each word x_j in the input sentence x_1^J is encoded in a continuous space. Fixed positional embeddings are also added to the word vectors to represent a word embedding $\bar{x}_j$. The encoder is a self-attentive module that maps an input sequence of words $\bar{x}_1^J$ into a sequence of continuous representations $h_1^J = H_{enc}(\bar{x}_1^J; \theta_{enc})$ where θ_{enc} are encoder parameters. The decoder is also a self-attentive module that at each time

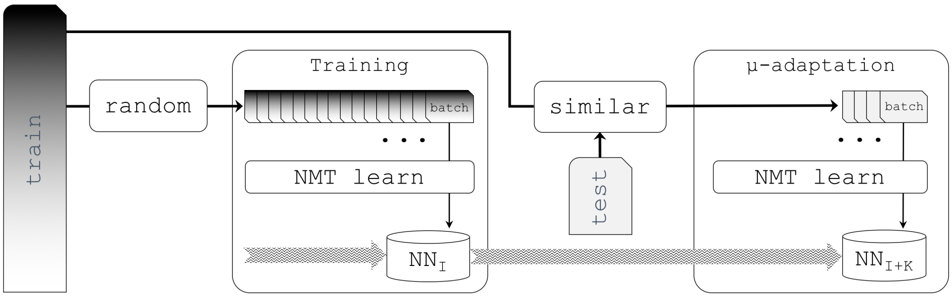

Figure 1: Generic training (left) and micro-adaptation (right) workflows. Training is performed using random samples of the entire data set. Micro-adaptation updates the network using a reduced amount of samples found similar to a given test set.

step outputs a single hidden state $s_i = H_{dec}(h_1^J, \bar{y}_{<i}; \theta_{dec})$, conditioned on the sequence of previously seen embedded target words $\bar{y}_{<i}$ and the encoder outputs h_1^J, where θ_{dec} are decoder parameters. The hidden state s_i is projected to the output vocabulary and normalised with a softmax operation resulting in a probability distribution over target words: $p(y_i|y_{<i}, x_1^J) = softmax(W \cdot s_i + b)$.

Parameters $(\theta_{enc}, \theta_{dec}, \dots)$ of the model need to be learned in order to generate high-quality translations. The standard way of training neural MT networks is by minimising cross entropy of the training data. Distance (error) between hypotheses and reference translations are measured and parameters are moved one step towards the direction that reduces the error. The length of the step is moderated by the *learning rate*, that may vary according to the optimisation algorithm. As in the original research work of Transformer, in this work we use the Adam [10] optimiser. During training, the optimisation algorithm goes through multiple *iterations* or several so-called *epochs*. Figure 1 (left) illustrates the training process with a model built after I iterations. In the case of micro-adaptation, the network is updated with a reduced set of examples considered similar to input sentences. Figure 1 (right) illustrates the micro-adaptation process where K additional iterations are performed. Since micro-adaptation is performed using few training samples, it results extremely important to control overfitting and catastrophic forgetting risk. In particular for training samples with low similarity to test sentences.

3. Selection of Similar Sentences

As illustrated by Figure 1 (right), our framework collects on-the-fly training sentence pairs similar to test sentences. Such sentences are then used to update the network parameters (micro-adaptation). Thus, the quality of the collected sentences is crucial to improve translation performance. There exists many methods to calculate the similarity between two sentences. We employ and compare two standard methodologies. The first is fully lexicalised, based on n-gram overlaps. The second employs distributed word representations.

3.1. Lexical Overlap

Our initial method relies on fuzzy matches. This is, for each input sentence to translate (s_t) we identify the N most similar training sentences (s_T) as measured by a fuzzy match score (FM). We define $FM(s_t, s_T)$ as:

$$FM(s_t, s_T) = 1 - \frac{ED(s_t, s_T)}{max(|s_t|, |s_T|)} \tag{1}$$

where $ED(s_t, s_T)$ is the edit distance between s_t and s_T, and $|s|$ is the length of s. Table 1 shows examples of a test sentence (s_t) and training sentences (s_{T_1} and s_{T_2}) with scores $FM(s_t, s_{T_1}) = 1 - \frac{1}{6} = 0.8\widehat{3}$ (one substitution) and $FM(s_t, s_{T_2}) = 1 - \frac{4}{10} = 0.6$ (four insertions).

s_t : *The 2nd paragraph of the Article*
s_{T_1}: *The* **3rd** *paragraph of the Article*
s_{T_2}: *The 2nd paragraph of the Article* **1 is deleted .**

Table 1: Test (s_t) and train (s_{T_1} and s_{T_2}) sentences.

Note that good micro-adaptation candidates (like s_{T_2} in our example) may receive a low score when training and test sentences have different sizes (needing for multiple insertion/deletion operations). Thus, we introduce a second similarity score that focus on the number of n-gram matchings (NM) between s_t and s_T. We define NM as:

$$NM_{\alpha,\beta}(s_t, s_T) = \sum_{n=\alpha}^{\beta} \sum_{\mathcal{C}_n \in s_t} [\mathcal{C}_n \in s_T] \tag{2}$$

where $\mathcal{C}_n$ is an n-gram (consecutive sequence of n words), α and β are the lower and higher n-gram lengths considered and $[\mathcal{P}]$ is the Iverson bracket that returns one if $\mathcal{P}$ is true and zero otherwise. Following with the examples of Table 1, n-gram matching scores are $NM_{2,4}(s_t, s_{T_1}) = 6$ and $NM_{2,4}(s_t, s_{T_2}) = 12$. In order to implement fast retrieval we use Suffix Arrays [11]. Training sentences containing any n-gram ($n \in [2, 4]$) present in the input sentence are initially retrieved. Then, FM and NM scores are computed leading to collect the most similar train sentences of a given input sentence.

3.2. Sentence Distributed Representations

As mentioned in Section 2, the encoder module of Transformer outputs a sequence of hidden representations h_1^J corresponding to words x_1^J of the input sentence. Inspired from the work of [12], we combine the recurrent h_1^J with *mean pooling* and *max pooling* to obtain a fixed-size vector representation h (h_{mean} or h_{max}). Thus, similarity between sentences s_1 and s_2 can be easily computed via cosine similarity of their distributed representations h_1 and h_2:

$$sim(s_1, s_2) = \frac{h_1 \cdot h_2}{||h_1|| \times ||h_2||} \quad (3)$$

In order to implement fast similarity retrieval between test and training sentences we use `faiss`[1] toolkit [13]. Training sentences identified as similar to each test input sentence are used for micro-adaptation.

4. Experimental Setup

4.1. Corpora

We perform English→French translation experiments. Data used in our experiments is a combination of heterogeneous datasets publicly available[2]: documentation from the European Central Bank (ECB); documents from the European Medicines Agency (EMEA); news commentaries (NEWS); European Parliament proceedings (EPPS); crawled parallel data (COMM). Table 2 shows some statistics computed after a light tokenisation using the OpenNMT tokenizer[3] (conservative mode) which basically splits-off punctation. We pay special attention to our in-domain datasets (ECB and EMEA) for which we measure the accuracy of different NMT engines over their respective validation and test sets. We use three additional out-of-domain datasets (NEWS, EPPS and COMM) summing up to near ∼7M sentence pairs. Validation and test sets for ECB and EMEA are randomly selected from the original corpus discarding sentences that also appear in the training set. We expect our framework to perform differently for different overlapping levels between test and in-domain train sets.

Figure 2 summarises the number of n-gram overlaps between test sets and their corresponding train sets. Very similar trends are found for both domains (ECB and EMEA). The fact that ∼40% of the test 10-grams are present in their corresponding in-domain train sets indicates the important overlapping between both sets. Similarly, Figure 3 illustrates the number of validation sentences (left axis) for which the similarity score of the closest example found in training belongs to the given similarity range. For both domains, half of the validation sets are assigned a similarity score higher than 0.9. It is worth to note that low overlapping levels between test and train data is expected to result on poor adaptation performance.

Corpus	Lang.	Lines	Words	Vocab.	OOV
		Training			
ECB	En	193K	6.1M	37,149	-
	Fr		7.1M	51,211	-
EMEA	En	1,090K	15.2M	49,903	-
	Fr		17.9M	60,246	-
NEWS	En	258K	6.7M	77,656	-
	Fr		8.4M	81,119	-
EPPS	En	2,007K	56.4M	96,878	-
	Fr		65.3M	125,556	-
COMM	En	3,244K	84.2M	754,861	-
	Fr		96.2M	839,012	-
		Validation			
ECB	En	1,000	31,917	4,483	127
	Fr		36,953	5,252	137
EMEA	En	1,000	21,174	3,821	44
	Fr		24,583	4,266	51
		Test			
ECB	En	1,000	33,073	4,515	63
	Fr		39,072	5,526	114
EMEA	En	1,000	20,187	4,277	77
	Fr		23,839	4,747	103

Table 2: Statistics for training, validation and test sets for the corpora used in this work. K stands for thousands and M for millions. Statistics were computed over the original data after splitting-off punctuation.

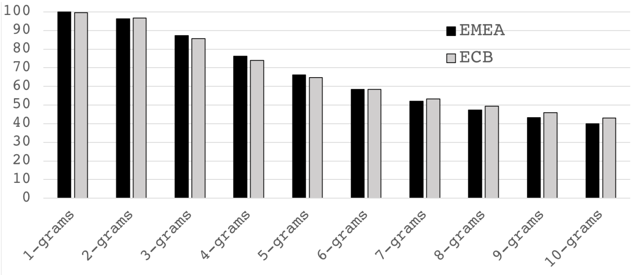

Figure 2: Lexical overlap (percentage of n-grams) between test sets and corresponding in-domain train sets.

4.2. Network Configuration

Our model follows the state-of-the-art Transformer architecture [5] implemented by the `OpenNMT-tf`[4] toolkit [14] learned with the hyper-parameters: size of word embedding: 512; size of hidden layers: 512; size of inner feed forward layer: 2,048; Multi-head: 8; number of layers: 6; batch size: 3,072 tokens. We use the lazy Adam optimiser. To vary the learning rate over training we use the original formula with parameters $\beta_1 = 0.9$, $\beta_2 = 0.98$ and $\epsilon = 10^{-9}$ and $4,000$ warmup steps. Learning rate is updated every 8 iterations. Fine-tuning is performed continuing Adam with the same learning rate decay schedule. Micro-adaptation is performed with a fixed learning rate value. We train a 32K joint byte-pair encoding (BPE) to jointly preprocess the

[1] https://github.com/facebookresearch/faiss

[2] http://opus.nlpl.eu

[3] https://pypi.org/project/pyonmttok/

[4] https://github.com/OpenNMT/OpenNMT-tf

French and English data [15]. We limit the sentence length to 80 based on BPE preprocessing in both source and target sides. In inference we use a beam size of 5. Then, we remove the BPE joiners and evaluate translation hypotheses with `multi-bleu.perl` [16].

4.3. Results

Table 3 summarises BLEU scores over in-domain validation and test sets for different network configurations[5]. Results are computed as average of three different optimisations (micro-adaptations). Our base network (*Mixed*) is trained with a combination of the training sets detailed in Table 2, summing up to near ∼7M sentence pairs. Training is performed over 300,000 iterations. Following [7], we fine-tune the *Mixed* network with each of the in-domain data sets, thus obtaining two networks specialised on each of the domains (*FT*). Further training the network is performed over 5 epochs. This is, $\sim$ 8000 iterations for ECB and $\sim$ 26000 iterations for EMEA. Finally, we perform micro-adaptation over our base *Mixed* network as described above: $\mu A_{h_{mean}}$ and μA_{NM}. The two similarity measures described in section 3 are considered: in μA_{NM}, sentences are collected based on n-gram matchings (Equation 2), while $\mu A_{h_{mean}}$ retrieves sentences based on distributed representations (Equation 3). We use optimal micro-adaptation parameters: $\mathcal{N} = 35$ train sentences are collected for each input sentence. Micro-adaptation is performed during 9 epochs with learning rate set to 0.0002.

Network	Validation		Test	
	ECB	EMEA	ECB	EMEA
Mixed	42.95	48.82	43.65	42.04
FT	45.98	52.38	48.41	43.48
μA_{NM}	47.15	**54.15**	**49.72**	**44.12**
$\mu A_{h_{mean}}$	**47.19**	54.02	49.37	43.98

Table 3: BLEU scores of different network configurations.

As we can see in table 3, micro-adaptation results clearly outperform those obtained by fine-tuning (*FT*) for both test sets. Very similar results are obtained when using lexicalised (μA_{NM}) and distributed representations ($\mu A_{h_{mean}}$) for similarity computation. Figure 3 illustrates the number of validation sentences (left axis) for which the similarity score of the closest example found in training belongs to the given range. In addition, it also shows the BLEU gain (right axis) of each range when comparing $\mu A_{h_{mean}}$ to *Mixed*.[6] As expected, highest improvements are measured for ranges with large similarity scores.

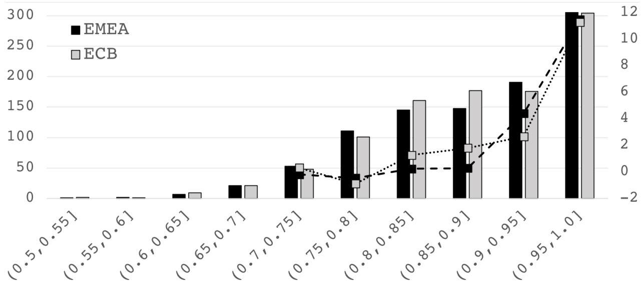

Figure 3: Number of validation sentences (bars, left axis) and BLEU gain (lines, right axis) by similarity range (h_{mean}).

In the next section we use $\mu A_{h_{mean}}$ as default model and analyse the impact on accuracy of different micro-adaptation parameters.

4.3.1. Micro-adaptation Analysis

We first evaluate the performance of the methods to compute sentence similarity (`Similarity`). For distributed representations, mean-pooling outperforms max-pooling for both validation sets. For lexicalised methods, n-gram matching (NM) obtains better results for the EMEA set while equivalent results for ECB. Results are inline with findings of [19] where improvements are reported with local n-gram matchings rather than global sentence similarity. Similar accuracies are obtained by h_{mean} and *N*M.

We also assess the impact on accuracy of the number of similar in-domain training samples collected for each input sentence (`N-best`). Note that micro-adaptation is performed using the union of samples collected for all validation sentences. In this case, best performance is found when micro-adaptation considers the 35 most similar training sentences (according to h_{mean}) of each input sentence. Thus, using 35 x 1,000 in-domain training samples. Note that micro-adaptation requires additional workload in inference. We provide efficiency results (`Time`) indicated by seconds spent per epoch and per input sentence according to the number of similar training sentences retrieved.[7] Results are inline with the efficiency results showed in [20].

Overfitting and catastrophic forgetting are important issues when learning over a reduced set of samples. Thus, we now finely tune the number of additional iterations and learning rate applied for micro-adaptation work. Concerning the number of iterations (`#Epochs`), best results are obtained when adaptation is run between 9 and 11 epochs. Since ECB and EMEA contain ∼35,000 training samples of different lengths, 9 epochs imply around 3,000 iterations (depending on the number and length of sentences in a batch). Larger values result in overfitting. Thus, limiting the generalisation ability of the network. Lower values results in under-fitting. Thus, limiting the ability to effectively learn from examples.

Regarding learning rate (`LRate`), best results are ob-

[5] Our BLEU scores are lower than those published on earlier studies [17, 18]. This is because EMEA and ECB datasets as distributed by OPUS contain many duplicates and we decided to make validation/test sets entirely disjoints from their respective train sets.

[6] No results are given for lower similarity range sets as they contain very few sentences.

[7] Time results are computed over the EMEA validation set.

tained when set to 2×10^{-4}. Lower values produce very small updates of the model, leading to similar results than those obtained by the original *Mixed* network. Higher values imply large updates, resulting in unstable training.

Similarity	ECB	EMEA	.
h_{max}	46.50	53.65	
h_{mean}	**47.19**	54.02	
NM	47.15	**54.15**	
FM	47.16	53.69	
N-best	ECB	EMEA	Time
15	45.67	52.60	0.052
20	46.54	53.98	0.073
25	46.40	53.88	0.092
30	47.02	53.64	0.106
35	**47.19**	**54.02**	0.124
40	46.83	53.46	0.142
#Epochs	ECB	EMEA	
5	46.13	53.06	
6	46.47	52.78	
7	46.86	53.15	
8	46.81	53.53	
9	47.19	**54.02**	
10	47.30	53.81	
11	**47.37**	53.55	
12	47.28	53.72	
LRate	ECB	EMEA	
1×10^{-3}	45.83	53.34	
2×10^{-4}	**47.19**	**54.02**	
1×10^{-4}	46.43	53.05	
1×10^{-5}	44.32	50.36	
1×10^{-6}	43.30	49.27	

Table 4: BLEU scores computed on validation sets and time overhead by sentence (Time) for different values of several micro-adaptation parameters.

4.3.2. Pseudo In-domain Data

In previous experiments we showed that in-domain training sentences similar to a given test set are suitable for micro-adaptation. We now show that our framework can also take advantage of similar sentences present in an out-of-domain training set. Thus, we employ h_{mean} to identify the M most similar out-of-domain training sentences to each input sentence. Notice that micro-adaptation is now performed over the union of the M most similar out-of-domain train sentences and the 35-best most similar in-domain train sentences. Table 5 shows BLEU scores over development sets when varying the number M of sentences retrieved. Scores for the best configuration over test sets are also given. Results indicate a light improvement when adding 5-best out-of-domain sentences over the ECB validation set (+0.16) while no gain is observed for the EMEA validation set. Results over test set confirm this improvement (+0.37).

Table 6 illustrates two examples of ECB test sentences (t_{ECB}) together with the most similar in-domain (T_{ECB}) and out-of-domain (T_{OUT}) training sentences found using h_{mean} similarity measure. The first example shows that none of the most similar sentences identified carry the exact same meaning than the test one. Both are fairly close and may be useful for adaptation broadening the diversity of training data. In the second example both, in-domain (T_{ECB}) and out-of-domain (T_{OUT}) similar sentences are equally extremely close to the input sentence. It is therefore difficult, to determine to what an extent the use of both sentences in micro-adaptation can contribute to the resulting model. Hypotheses produced by both micro-adapted models are extremely similar (outlined using bold letters). A single word replaced by a synonym. In both cases, the synonym used in Hyp^2 appears in the reference.

M-best	ECB	EMEA
Validation		
0	47.19	**54.02**
1	47.33	53.61
3	47.10	53.44
5	**47.35**	54.00
7	46.86	53.35
9	46.88	53.37
Test		
0	49.37	43.98
5	**49.74**	**43.99**

Table 5: BLEU scores using similar sentences retrieved from both in- and out-of-domain train data.

5. Related Work

Domain adaptation has been deeply studied from a number of perspectives, ranging from theoretical analysis to more applied work, and for which many solutions have been proposed. In the case of Machine Translation, literature of domain adaptation typically distinguishes *data-based* approaches from *model-based* approaches [21, 22]. One of the most common adaptation scenarios uses out-of-domain (or heterogeneous) data sources for training, while testing on in-domain texts. In this setting, *data-based* approaches aim to bias the distribution of the training data towards matching that of the target domain, using data selection techniques [23, 24, 25], or producing synthetic parallel data following the in-domain distribution [26, 27, 28, 29]. In contrast, *model-based* approaches build domain-adapted models by biasing the training objective towards the desired domain using in-domain data [7, 30, 31], or building networks with domain-specialised layers [32, 6, 33, 17]. Thus, effectively enabling multi-domain networks, a practical scenario in the industry which allows to be both data efficient (all data is used to train all domains) and computationally efficient (a single model is built for all domains).

Our work follows a framework where a unique generic system is built off-line, and adaptation is dynamically applied

t_{ECB} :	The SEPA project represents the next major step towards closer European integration.
Ref :	Le projet SEPA constitue une nouvelle étape majeure vers une plus grande intégration européenne.
T_{ECB} :	The draft Constitution is an important step in preparing the Union for the future.
Hyp^1 :	Le projet SEPA **représente** la prochaine étape majeure vers une intégration européenne plus étroite.
T_{OUT} :	This was a significant new step along the road towards European integration.
Hyp^2 :	Le projet SEPA **constitue** la prochaine étape majeure vers une intégration européenne plus étroite.
t_{ECB} :	The external environment remains favourable, providing support for euro area exports.
Ref :	L' environnement extérieur reste bien orienté et soutient les exportations de la zone euro.
T_{ECB} :	The external environment is favourable, providing support for euro area exports.
Hyp^1 :	L'environnement extérieur **demeure** favorable, soutenant ainsi les exportations de la zone euro.
T_{OUT} :	External conditions thus continue to provide support for euro area exports.
Hyp^2 :	L'environnement extérieur **reste** favorable, soutenant ainsi les exportations de la zone euro.

Table 6: Input test sentences (t_{ECB}) together with their most similar in-domain (T_{ECB}) and out-of-domain (T_{OUT}) training sentences as found by h_{mean} similarity measure. Reference (Ref) and translation hypotheses (Hyp^1 and Hyp^2) are also shown, produced by $\mu A_{h_{mean}}$ micro-adapted to in-domain (Hyp^1, best system in Table 4 (`N-best`)) or in-domain + out-of-domain (Hyp^2, best system in Table 5) train sentences.

using a small amount of training data considered similar to input sentences. Closely related, in the context of *Statistical MT*, [34] proposes to dynamically extend the translation model with translation memory entries obtaining high levels (above 70%) of fuzzy matches with input sentences. These examples are encoded as additional translation rules in the translation model. In [35] is presented a generic framework that employs an additional language model to guide the decoder. The language model is built over lexical hypotheses produced by auxiliary translation engines.

Concerning *Neural MT*, in [36] is presented a dynamic data selection method to fine tune NMT models. The authors show that increasingly reducing the amount of data used for fine tuning and regularly computing sentence similarities outperforms static data selection. [37] follows a data selection strategy to train only on semantically related sentences. Thus, building from scratch a model adapted to a given test set. Further adaptation is applied at inference time following the work in [7]. Our work mainly differs from the previous as we apply adaptation at inference time. In [18] and [20] a generic translation model is dynamically adapted to each input sentence making use of the most similar sentences found in training datasets. To compute similarity, the first work uses a fully lexicalised n-gram matching score and adapts a model built from heterogeneous corpora, while the second includes dense vector representations and applies adaptation to a model built from parliamentary documents (UN). In our work we consider a different scenario where test sentences are semantically related while in previous works input sentences come from different domains with no order. In addition, our NMT model follows the Transformer architecture of [5] while the architecture described in [4] is used in previous works.

A non-parametric adaptation solution is proposed by [19]. Similar to our work the authors compute sentence similarities using dense vector representations. However, they focus on retrieving n-grams rather than sentences, and propose a network that is able to incorporate relevant information present in such n-grams. A similar solution is proposed by [38]. But they modify a standard NMT model to apply a bonus to hypotheses that contain the collected n-gram translations. Similarly, [40] employs similar training translations but modifies the NMT network architecture to more accurately distinguish and take into account the useful information of the retrieved similar translations. Finally, [39] uses also n-gram fuzzy matching to collect similar sentences collected on translation memories but makes use of an NMT engine trained to translate an input stream composed of the input sentence concatenated with related translations, requiring existence of translation memory before training.

6. Conclusions

In this paper we propose a generic framework to adapt a neural MT network built from heterogeneous corpora to a homogeneous test set, a very common industry scenario. At inference-time, a reduced amount of training instances similar to input sentences are dynamically retrieved, and used to update the network parameters. Experimental results demonstrate both efficiency and performance of the framework: it outperforms the state-of-the-art fine tuning, typically performed as a pre-training task incompatible to online adaptation to new data. Both experienced methodologies for similarity computation ($\mu A_{h_{mean}}$ and μA_{NM}) demonstrated similar performance. We also showed that out-of-domain corpora may also be used for micro-adaptation to further boost accuracy. In the future, we plan to study the similarity required to apply micro-adaptation as well as dynamically vary the size of the adaptation set. Similar to [41] we also plan to apply micro-adaptation over semantically highly related sentences rather than entire test sets. We also would like to improve similarity retrieval methods, which currently focus on high overlapping rates. However, some sentences may also be useful for adaptation if they include relevant n-grams.

7. References

[1] N. Kalchbrenner and P. Blunsom, "Recurrent continuous translation models," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Seattle, Washington, USA: Association for Computational Linguistics, Oct. 2013, pp. 1700–1709. [Online]. Available: https://www.aclweb.org/anthology/D13-1176

[2] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 3104–3112. [Online]. Available: http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf

[3] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1724–1734. [Online]. Available: https://www.aclweb.org/anthology/D14-1179

[4] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. [Online]. Available: http://arxiv.org/abs/1409.0473

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 5998–6008. [Online]. Available: http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf

[6] C. Kobus, J. Crego, and J. Senellart, "Domain control for neural machine translation," in *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*. Varna, Bulgaria: INCOMA Ltd., Sept. 2017, pp. 372–378. [Online]. Available: https://doi.org/10.26615/978-954-452-049-6_049

[7] M.-T. Luong and C. D. Manning, "Stanford neural machine translation systems for spoken language domain," in *International Workshop on Spoken Language Translation*, Da Nang, Vietnam, 2015. [Online]. Available: https://nlp.stanford.edu/pubs/luong-manning-iwslt15.pdf

[8] P. Michel and G. Neubig, "Extreme adaptation for personalized neural machine translation," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 312–318. [Online]. Available: https://www.aclweb.org/anthology/P18-2050

[9] J. Wuebker, P. Simianer, and J. DeNero, "Compact personalized models for neural machine translation," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, Oct.-Nov. 2018, pp. 881–886. [Online]. Available: https://www.aclweb.org/anthology/D18-1104

[10] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: http://arxiv.org/abs/1412.6980

[11] U. Manber and G. Myers, "Suffix arrays: A new method for on-line string searches," in *Proceedings of the First Annual ACM-SIAM Symposium on Discrete Algorithms*, ser. SODA '90. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 1990, pp. 319–327. [Online]. Available: http://dl.acm.org/citation.cfm?id=320176.320218

[12] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, and A. Bordes, "Supervised learning of universal sentence representations from natural language inference data," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, Sept. 2017, pp. 670–680. [Online]. Available: https://www.aclweb.org/anthology/D17-1070

[13] J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with gpus," *arXiv preprint arXiv:1702.08734*, 2017.

[14] G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. Rush, "OpenNMT: Open-source toolkit for neural machine translation," in *Proceedings of ACL 2017, System Demonstrations*. Vancouver, Canada: Association for Computational Linguistics, 2017, pp. 67–72. [Online]. Available: http://aclweb.org/anthology/P17-4012

[15] R. Sennrich, B. Haddow, and A. Birch, "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)*. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 1715–1725. [Online]. Available: https://www.aclweb.org/anthology/P16-1162

[16] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "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, June 2007, pp. 177–180. [Online]. Available: https://www.aclweb.org/anthology/P07-2045

[17] J. Zeng, J. Su, H. Wen, Y. Liu, J. Xie, Y. Yin, and J. Zhao, "Multi-domain neural machine translation with word-level domain context discrimination," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, Oct.-Nov. 2018, pp. 447–457. [Online]. Available: https://www.aclweb.org/anthology/D18-1041

[18] M. A. Farajian, M. Turchi, M. Negri, and M. Federico, "Multi-domain neural machine translation through unsupervised adaptation," in *Proceedings of the Second Conference on Machine Translation*. Copenhagen, Denmark: Association for Computational Linguistics, Sept. 2017, pp. 127–137. [Online]. Available: https://www.aclweb.org/anthology/W17-4713

[19] A. Bapna and O. Firat, "Non-parametric adaptation 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)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 1921–1931. [Online]. Available: https://www.aclweb.org/anthology/N19-1191

[20] X. Li, J. Zhang, and C. Zong, "One sentence one model for neural machine translation," in *Proceedings of the 11th Language Resources and Evaluation Conference*. Miyazaki, Japan: European Language Resource Association, May 2018. [Online]. Available: https://www.aclweb.org/anthology/L18-1146

[21] C. Chu, R. Dabre, and S. Kurohashi, "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)*. Vancouver, Canada: Association for Computational Linguistics, 2017, pp. 385–391. [Online]. Available: http://aclweb.org/anthology/P17-2061

[22] C. Chu and R. Wang, "A survey of domain adaptation for neural machine translation," in *Proceedings of the 27th International Conference on Computational Linguistics*, ser. COLING 2018, Santa Fe, New Mexico, USA, 2018, pp. 1304–1319. [Online]. Available: http://aclweb.org/anthology/C18-1111

[23] R. C. Moore and W. Lewis, "Intelligent selection of language model training data," in *Proceedings of the ACL 2010 Conference Short Papers*. Uppsala, Sweden: Association for Computational Linguistics, 2010, pp. 220–224. [Online]. Available: http://aclweb.org/anthology/P10-2041

[24] A. Axelrod, X. He, and J. Gao, "Domain adaptation via pseudo in-domain data selection," in *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*. Edinburgh, Scotland, UK.: Association for Computational Linguistics, 2011, pp. 355–362. [Online]. Available: http://aclweb.org/anthology/D11-1033

[25] K. Duh, G. Neubig, K. Sudoh, and H. Tsukada, "Adaptation data selection using neural language models: Experiments in machine translation," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Sofia, Bulgaria: Association for Computational Linguistics, 2013, pp. 678–683. [Online]. Available: http://aclweb.org/anthology/P13-2119

[26] R. Wang, H. Zhao, B.-L. Lu, M. Utiyama, and E. Sumita, "Neural network based bilingual language model growing for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, 2014, pp. 189–195. [Online]. Available: http://aclweb.org/anthology/D14-1023

[27] ——, "Connecting phrase based statistical machine translation adaptation," in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. Osaka, Japan: The COLING 2016 Organizing Committee, 2016, pp. 3135–3145. [Online]. Available: http://aclweb.org/anthology/C16-1295

[28] R. Sennrich, B. Haddow, and A. Birch, "Improving neural machine translation models with monolingual data," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, 2016, pp. 86–96. [Online]. Available: http://aclweb.org/anthology/P16-1009

[29] M. Chinea-Rios, Á. Peris, and F. Casacuberta, "Adapting neural machine translation with parallel synthetic data," in *Proceedings of the Second Conference on Machine Translation*. Copenhagen, Denmark: Association for Computational Linguistics, Sept. 2017, pp. 138–147. [Online]. Available: https://www.aclweb.org/anthology/W17-4714

[30] M. Freitag and Y. Al-Onaizan, "Fast domain adaptation for neural machine translation," *CoRR*, vol. abs/1612.06897, 2016.

[31] B. Chen, C. Cherry, G. Foster, and S. Larkin, "Cost weighting for neural machine translation domain adaptation," in *Proceedings of the First Workshop on Neural Machine Translation*. Vancouver: Association for Computational Linguistics, 2017, pp. 40–46. [Online]. Available: http://aclweb.org/anthology/W17-3205

[32] Ç. Gülçehre, O. Firat, K. Xu, K. Cho, L. Barrault, H.-C. Lin, F. Bougares, H. Schwenk, and Y. Bengio, "On using monolingual corpora in neural machine translation," *arXiv e-prints*, vol. abs/1503.03535, Mar. 2015. [Online]. Available: https://arxiv.org/abs/1503.03535

[33] D. Vilar, "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)*. New Orleans, Louisiana: Association for Computational Linguistics, June 2018, pp. 500–505. [Online]. Available: https://www.aclweb.org/anthology/N18-2080

[34] P. Koehn and J. Senellart, "Convergence of Translation Memory and Statistical Machine Translation," in *Proceedings of AMTA Workshop on MT Research and the Translation Industry*, Denver, 2010, pp. 21–31. [Online]. Available: http://homepages.inf.ed.ac.uk/pkoehn/publications/tm-smt-amta2010.pdf

[35] J. M. Crego, A. Max, and F. Yvon, "Local lexical adaptation in machine translation through triangulation: SMT helping SMT," in *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*. Beijing, China: Coling 2010 Organizing Committee, Aug. 2010, pp. 232–240. [Online]. Available: https://www.aclweb.org/anthology/C10-1027

[36] M. van der Wees, A. Bisazza, and C. Monz, "Dynamic data selection for neural machine translation," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, Sept. 2017, pp. 1400–1410. [Online]. Available: https://www.aclweb.org/anthology/D17-1147

[37] R. Wang, A. Finch, M. Utiyama, and E. Sumita, "Sentence embedding for neural machine translation domain adaptation," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Vancouver, Canada: Association for Computational Linguistics, July 2017, pp. 560–566. [Online]. Available: https://www.aclweb.org/anthology/P17-2089

[38] J. Zhang, M. Utiyama, E. Sumita, G. Neubig, and S. Nakamura, "Guiding neural machine translation with retrieved translation pieces," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, June 2018, pp. 1325–1335. [Online]. Available: https://www.aclweb.org/anthology/N18-1120

[39] B. Bulte and A. Tezcan, "Neural fuzzy repair: Integrating fuzzy matches into neural machine translation," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 1800–1809. [Online]. Available: https://www.aclweb.org/anthology/P19-1175

[40] J. Gu, Y. Wang, K. Cho, and V. O. K. Li, "Search engine guided neural machine translation," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, S. A. McIlraith and K. Q. Weinberger, Eds. AAAI Press, 2018, pp. 5133–5140. [Online]. Available: https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17282

[41] R. Sennrich, H. Schwenk, and W. Aransa, "A multi-domain translation model framework for statistical machine translation," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Sofia, Bulgaria: Association for Computational Linguistics, Aug. 2013, pp. 832–840. [Online]. Available: https://www.aclweb.org/anthology/P13-1082