Domain specialization: a post-training domain adaptation for Neural Machine Translation

Christophe Servan and **Josep Crego** and **Jean Senellart**
firstname.lastname@systrangroup.com
SYSTRAN / 5 rue Feydeau, 75002 Paris, France

Abstract

Domain adaptation is a key feature in Machine Translation. It generally encompasses terminology, domain and style adaptation, especially for human post-editing workflows in Computer Assisted Translation (CAT). With Neural Machine Translation (NMT), we introduce a new notion of domain adaptation that we call "specialization" and which is showing promising results both in the learning speed and in adaptation accuracy. In this paper, we propose to explore this approach under several perspectives.

1 Introduction

Domain adaptation techniques have successfully been used in Statistical Machine Translation. It is well known that an optimized model on a specific genre (litterature, speech, IT, patent...) obtains higher accuracy results than a "generic" system. The adaptation process can be done before, during or after the training process.

We propose to explore a new post-process approach, which incrementally adapt a "generic" model to a specific domain by running additional training epochs over newly available in-domain data.

In this way, adaptation proceeds incrementally when new in-domain data becomes available, generated by human translators in a post-edition context. Similar to the Computer Assisted Translation (CAT) framework described in (Cettolo et al., 2014).

Contributions The main contribution of this paper is a study of the new "specialization" approach, which aims to adapt generic NMT model without a full retraining process. Actually, it consist in using the generic model in a retraining phase, which only involves additional in-domain data. Results show this approach can reach good performances in a far less time than full-retraining, which is a key feature to adapt rapidly models in a CAT framework.

2 Approach

Following the framework proposed by (Cettolo et al., 2014), we seek to adapt incrementally a generic model to a specific task or domain. They show incremental adaptation brings new information in a Phrase-Based Statistical Machine Translation like terminology or style, which can also belong to the human translator. Recent advances in Machine Translation focuses on Neural Machine Translation approaches, for which we propose a method to adapt incrementally to a specific domain, in this specific framework.

The main idea of the approach is to specialize a generic model already trained on generic data. Hence, we propose to retrain the generic model on specific data, though several training iterations (see figure 2). The retraining process consist in re-estimating the conditional probability $p(y_1, \ldots, y_m | x_1, \ldots, x_n)$ where $(x_1, \ldots, x_n)$ is an input sequence of length n and $(y_1, \ldots, y_m)$ is its corresponding output sequence whose length m may differ from n. This is done without dropping the previous learning states of the Recurrent Neural Network.

The resulting model is considered as adapted or specialized to a specific domain.

3 Experiment framework

We create our own data framework described in the next section and we evaluate our results using the BLEU score (Papineni et al., 2002) and the TER (Snover et al., 2006).

The Neural Machine Translation system com-

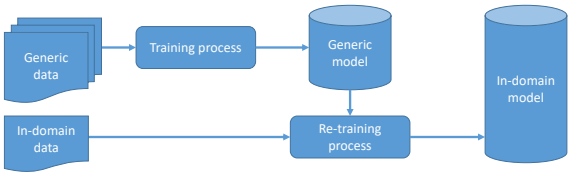

Figure 1: The generic model is trained with generic data, then the generic model obtained is retrained with in-domain data to generate an specialized model.

bines the attention model approach (Luong et al., 2015) jointly with the sequence-to-sequence approach (Sutskever et al., 2014).

According to our approach, we propose to compare several configurations, in which main difference is the training corpus. On one hand, we consider the generic data and several amounts of in-domain data for the training process. On the other hand, only the generic data are considered for the training process, then several amounts of in-domain data are used only for the specialization process in a retraining phase. The main idea behind these experiment is to simulate an incremental adaptation framework, which enables the adaptation process only when data are available (e.g.: translation post-editions done by a human translator.)

The approach is studied in the light of two experiments and a short linguistic study. The first experiment concerns the impact of "specialization" approach among several additional epochs; then, the second one, focuses on the amount of data needed to observe a significant impact on the translation scores. Finally, we propose to compare some translation examples from several outputs.

3.1 Training data

The table 1 presents all data used in our experiments. We propose to create a generic model with comparable amount of several corpora, which each of them belong to a specific domain (IT, literature, news, parliament). All corpora are available from the OPUS repository (Tiedemann, 2012).

We propose to specialize the generic model using a last corpus, which is a corpus extracted from the European Medical Agency (*emea*). The corpus is composed of more than 650 documents, which are medicine manuals.

We took apart a $2K$ lines as test corpus, then, to simulate the incremental adding of data, we created four training corpora corresponding to several amount of documents: 500, $5K$, $50K$ and all the lines of the training corpus. These amount of data are corresponding roughly to 10% of a document, one document and ten documents, respectively.

Type	Domain	#lines	#src tokens	#tgt tokens
Train	*generic*	3.4M	73M	86M
	emea-0.5K	500	5.6K	6.6K
	emea-5K	5K	56.1K	66.4K
	emea-50K	50K	568K	670K
	emea-full	922K	10.5M	12.3M
dev.	*generic*	2K	43.7K	51.3K
test	*emea*	2K	35.6K	42.9K

Table 1: details of corpora used in this paper.

Models	BLEU	TER
generic	26.23	62.47
generic+emea-0.5K	26.48	63.09
generic+emea-5K	28.99	58.98
generic+emea-50K	33.76	53.87
generic+emea-full	41.97	47.07

Table 2: BLEU score of full trained systems.

3.2 Training Details

The Neural Machine Translation approach we use is following the sequence-to-sequence approach (Sutskever et al., 2014) combined with attentional architecture (Luong et al., 2015). In addition, all the generic and in-domain data are pre-processed using the *byte pair encoding* compression algorithm (Sennrich et al., 2016) with 30K operations, to avoid Out-of-Vocabulary words.

We keep the most frequent $32K$ words for both source and target languages with 4 hidden layers with 500-dimensional embeddings and 800 bidirectional Long-Short Term Memory (bi-LSTM) cells. During training we use a mini-batch size of 64 with dropout probability set to 0.3. We train our models for 18 epochs and the learning rate is set to 1 and start decay after epoch 10 by 0.5. It takes about 8 days to train the generic model on our NVidia GeForce GTX 1080.

The models were trained with the open-source toolkit `seq2seq-attn`[1] (Kim and Rush, 2016).

3.3 Experiments

As a baseline, we fully trained five systems, one with the generic data (*generic*) and the other with generic and various amount of in-domain data: 500 lines (*emea-0.5K*), $5K$ lines (*emea-5K*) and

[1] `https://github.com/harvardnlp/seq2seq-attn`

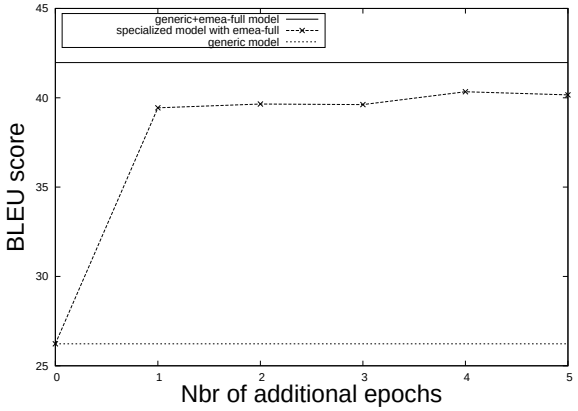

Figure 2: Curve of "specialization" performances among epochs.

Training corpus	Specialization corpus	BLEU	TER
generic	N/A	26.23	62.47
generic+emea-0.5K	N/A	26.48	63.09
generic+emea-5K	N/A	28.99	58.98
generic+emea-50K	N/A	33.76	53.87
generic+emea-full	N/A	41.97	47.07
generic	*emea-0.5K*	27.33	60.92
generic	*emea-5K*	28.41	58.84
generic	*emea-50K*	34.25	53.47
generic	*emea-full*	39.44	49.24

Table 3: BLEU and TER scores of the specialization approach on the in-domain test set.

Process	Corpus	#lines	#src tokens	#tgt tokens	Process time
Train	*generic*	3.4M	73M	86M	8 days
Specialization	*emea-0.5K*	500	5.6K	6.6K	<1 min
	emea-5K	5K	56.1K	66.4K	≈1 min
	emea-50K	50K	568K	670K	≈6 min
	emea-full	922K	10.5M	12.3M	105 min

Table 4: Time spent for each process, the training and the specialization process, according to the amount of data we have.

$50K$ lines (*emea-50K*). The evaluation is done on the in-domain test (*emea-tst*) and presented in the table 2. Without surprises, the more the model is trained with in-domain data the more BLEU and TER scores are improved. These models are baselines in incremental adaptation experiments.

3.3.1 Performances among training iterations

The first study aims to evaluate the approach among additional training iterations (also called "epochs"). Figure 2 presents the curve of performances when the specialization approach is applied to the *generic* model by using all in-domain data (*emea-full*).

We compare the results with two baselines: on the top of the graphic, we show a line which corresponds to the score obtained by the model trained with both generic and in-domain data (noted *generic+emea-full*); On the bottom, the line is associated to the generic model score, which is trained with only generic data (noted *generic*). The curve is done with the generic model specialized with five epochs additional epochs on all in-domain data (noted *specialized model with emea*). In the graphic, we can observe that a gap obtained with the first additional epoch with more than 13 points, but then the BLEU score improves around 0.15 points with each additional epoch and tend to stall after 10 epochs (not shown).

So far, the specialization approach does not replace a full retraining, while the specialization curve does not reach the *generic+emea-full* model. But, the retraining time of one additional epoch with all in-domain data is around 1 hour and 45 minutes, while a full retraining would takes more than 8 days.

In our CAT framework, even 1 hour and 45 minutes is too much, the adaptation process need to be performed faster with smaller amount of data like a part of a document (500 lines) or a full document ($5K$ lines). Considering the time constraint, the approach tends to be performed though one additional epoch.

3.3.2 Performances among data size

The second experiment concerns the observation of specialization performances when we vary the amount of data. Using the data presented Table 1, we apply the specialization process on the generic corpus by taking $0.5K$, $5K$, $50K$ and all the in-domain data (as presented in section 3.1). According to our previous study (see section 3.3.1), we focuses on the results obtained with only one additional epoch.

We can observe that with only 500 lines, the improvements reaches more than 1 BLEU points and 2 TER points. Then, with 10 time more additional data, BLEU and TER scores improved the baseline of 2 and nearly 4 points, respectively. With more additional data (10 documents), improvements reach 8 points of BLEU and 9 points of TER. Finally with all the in-domain data available, the specialization increase the baseline of 13 points of both BLEU and TER scores.

Comparing the approach with retraining all the generic data, with the same amount of in-domain

data, it appears our approach reaches nearly the same results. Moreover, with $50K$ of in-domain data, the specialization approach performs better of 0.5 of BLEU and TER points. But, when we have much more in-domain data available, the specialization approach does not outperforms the full retraining (39.44 against 41.97 BLEU points).

3.4 Discussion

Focussing on the time constraint of the CAT framework, the table 4 presents the time taken to process our specialization approach. It goes from less than one minutes to more than 1 hour and 45 minutes. If we compare this table with the table 3, we observe that this approach enables to gain 1 BLEU point in less than 1 minute, 2 points in 1 minute and more than 6 BLEU points in 6 minutes. The ratio of "time spent" to "score gained" seems impressive.

The table 5 shows an example of the outputs obtained with the specialization approach. We compare the generic model compared to the specialized models with respectively $0.5K$, $5K$ and $50K$ lines of in-domain data.

We can clearly see the improvements obtained on the translation outputs. Even if the last one does not stick strictly to the reference, the translation output can be considered as a good translation (syntactically well formed and semantically equivalent).

This specialization approach can be seen as an optimization process (like in classical Phrase-Based approach), which aims to tune the model (Och, 2003).

4 Related work

Last years, domain adaptation for machine translation has received lot of attention and studies. These approaches can be processed at three levels: the pre-processing, the training, the post-processing. In a CAT framework, most of the approaches focuses on the pre-processing or on the post-processing to adapt models.

Such pre-processing approaches like data selection introduced by (Lü et al., 2007) and improved by (Gao and Zhang, 2002) and many others (Moore and Lewis, 2010; Axelrod et al., 2011) are effective and their impact studied (Lambert et al., 2011; Cettolo et al., 2014; Wuebker et al., 2014). But, the main draw back of these approaches is they need a full retrain to be effective.

The post-training family concerns methods which aims to update the model or to optimize the model to a specific domain. Our approach belongs to this category.

This approach is inspired by (Luong and Manning, 2015), they propose to train a generic model and, then, they further a training over a dozen of epochs on a full in-domain data (the TED corpus). We do believe this approach is under estimated and we propose to study its efficiency in a specific CAT framework with a few data. On one hand, we propose to follow this approach by proposing to use a fully trained generic model. But, on the other hand, we propose to train further only on small specific data over a few additional epochs (from 1 to 5). In this way, our approach is slightly different and can be equated to a tuning process (Och, 2003).

5 Conclusion

In this paper we propose a study of the "specialization" approach. This domain adaptation approach shows good improvements with few in-domain data in a very short time. For instance, to gain 2 BLEU points, we used 5K lines of in-domain data, which takes 1 minute to be performed.

Moreover, this approach reaches the same results as a full retraining, when 10 documents are available. Within a CAT framework, this approach could be a solution for incremental adaptation of NMT models, and could be performed between two rounds of post-edition. In this way, we propose as future work to evaluate our approach in a real CAT framework.

References

[Axelrod et al.2011] Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 355–362, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.

[Cettolo et al.2014] Mauro Cettolo, Nicola Bertoldi, Marcello Federico, Holger Schwenk, Loic Barrault, and Chrstophe Servan. 2014. Translation project adaptation for mt-enhanced computer assisted translation. *Machine Translation*, 28(2):127–150, October.

[Gao and Zhang2002] Jianfeng Gao and Min Zhang. 2002. Improving language model size reduction using better pruning criteria. In *Proceedings of*

the 40th Annual Meeting on Association for Computational Linguistics, ACL '02, pages 176–182, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Kim and Rush2016] Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Austin, Texas, November.

[Lambert et al.2011] Patrik Lambert, Holger Schwenk, Christophe Servan, and Sadaf Abdul-Rauf. 2011. Investigations on translation model adaptation using monolingual data. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 284–293, Edinburgh, Scotland, July. Association for Computational Linguistics.

[Lü et al.2007] Yajuan Lü, Jin Huang, and Qun Liu. 2007. Improving statistical machine translation performance by training data selection and optimization. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 343–350.

[Luong and Manning2015] Minh-Thang Luong and Christopher D. Manning. 2015. Stanford neural machine translation systems for spoken language domains. In *IWSLT2015*, Da Nang, Vietnam, December.

[Luong et al.2015] Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal, September.

[Moore and Lewis2010] R.C. Moore and W. Lewis. 2010. Intelligent Selection of Language Model Training Data. In *ACL (Short Papers)*, pages 220–224, Uppsala, Sweden, July.

[Och2003] Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, pages 160–167.

[Papineni et al.2002] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July.

[Sennrich et al.2016] Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. 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)*, pages 86–96, Berlin, Germany, August. Association for Computational Linguistics.

[Snover et al.2006] Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of association for machine translation in the Americas*.

[Sutskever et al.2014] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *NIPS*.

[Tiedemann2012] Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC'2012)*.

[Wuebker et al.2014] Joern Wuebker, Hermann Ney, Adrià Martínez-Villaronga, Adrià Giménez, Alfons Juan, Christophe Servan, Marc Dymetman, and Shachar Mirkin. 2014. Comparison of data selection techniques for the translation of video lectures. In *AMTA*.

Source:	What benefit has SonoVue shown during the studies ?
Reference:	Quel est le bénéfice démontré par SonoVue au cours des études ?
generic model	Quel avantage SSonVue a-t-il montré pendant les études ?
specialization *emea-0.5K*	Quel bénéfice SSonVue a-t-il montré lors des études ?
specialization *emea-5K*	Quel bénéfice SSonVue a-il montré pendant les études ?
specialization *emea-50K*	Quels est le bénéfice démontré par SonoVue au cours des études ?

Table 5: Example of translation output of the generic model and the specialized models with different amount of in-domain data. Red, blue and green are, respectively, *bad*, *acceptable* and *good* translations.