Improving Domain Adaptation for Machine Translation with Translation Pieces

Catarina Silva
Unbabel
catarina@unbabel.com

Abstract

Neural Machine Translation has achieved impressive results in the last couple years, in particular when aided by domain adaptation methods. However, it has well known caveats, and can sometimes generate inadequate content that appears fluent, but does not convey the meaning of the original sentence. In particular, for scarce in-domain data, these models tend to overfit, performing poorly on any content that differs slightly from the domain data. In this paper, we apply a recent technique based on translation pieces and show that it can work as a way to improve and stabilize domain adaptation. We present human evaluation results, with gains as high as 20 MQM points for single domains, and consistent gains in a multiple subdomain scenario of 3 MQM points for several language pairs.

1 Introduction

Neural Machine Translation (NMT) is a state-of-the-art technique to do machine translation. While NMT has proved to be efficient translating texts between multiple languages (Sennrich et al, 2016a) (Wang et al, 2017) in general settings, the ability of NMT technology to adapt to new domain has not attracted much focus from the research community. On the other hand, domain adaptation is a fundamental element of many industrial applications in which in-domain data is often scarce. In one of the most popular scenarios, large generic data is used to train an initial NMT system, obtaining a set of parameters that are then fine tuned on the much smaller in-domain corpus (Chu and Wang, 2018). The problem with this approach is that NMT often overfits to the target domain, which makes it less robust when it has to translate content which differs even slightly from the in-domain training data (Arthur et all, 2016) (Kaiser et al, 2017). Thus, the problem also holds when trying to generalize across more than one domain (Koehn and Knowles, 2017), where improving in one of the domains might sacrifice quality in others. In this paper, we present a possible solution to this issue, describing an application of the retrieved translation pieces (Zhang et al, 2018), which we show to help with cross-domain generalisation. We show that translation pieces improves the translation quality in a single domain, but also when combining multiple domains.

2 Experimental Setup

2.1 Neural Machine Translation

The aim of this work is to assess the impact of adding translation pieces to NMT models. For this purpose, we use the Marian framework[1] (Junczys-Dowmunt et al, 2018) to train models using the attention-based encoder–decoder architecture as described in Sennrich et al (2017).

For all experiments a standard preprocessing routine was applied, consisting of the following steps: entity replacement, tokenisation, truecasing and Byte-Pair Encoding (BPE) (Sennrich et al, 2016b) with 89,500 merge operations.

The process of entity replacement consists of identifying entities that should not be translated or that have a direct translation, usually defined

[1]https://marian-nmt.github.io/

as glossaries, and replacing them by placeholders that are put back in the text after translation. These placeholders are protected in the BPE step, so they are always considered as unique subwords.

2.2 Domain Adaptation

The domain adaptation process consists on a typical setting of training that uses a pretrained model:

- A model is trained with generic data and its best performing version is kept - using BLEU as the performance metric at validation;
- The generic model is passed as the `pretrained-model` flag in Marian;
- The initialized model is trained on in-domain data, with a lower validation frequency to achieve a validation per epoch .

Besides the validation frequency, for this experiment no other parameters were tuned in the model in the domain adaptation stage.

2.3 Translation Pieces Retrieval

For these experiments, we follow a process similar to the one proposed by Zhang et al (2018), where we assume we have a pool of pairs in our language pair – source and corresponding translation – that we can sample before-hand, from which we retrieve nearest neighbours with respect to the source sentences in our training data. Our retrieval process, unlike the base method, is based on sentence embeddings and not on a search engine. We then pick the retrieved sentences and compute a similarity measure that is used to score the pieces.

Translation Memories Retrieval: We get average sentence embeddings over all candidates and query sentences, through fastText word embeddings[2] (Joulin et al, 2016), and then run a nearest neighbour algorithm with FAISS (Johnson et al, 2017). This retrieves the neighbours by measuring the cosine similarity between the query embedding and candidate embeddings, as shown in Figure 1.

Translation Pieces generation: To get the related translation pieces, we first go through query and candidate sentences and get the unedited words, this is, equal words appearing in both on the same order. We then run an aligner on the candidate's source and target and get the target pieces

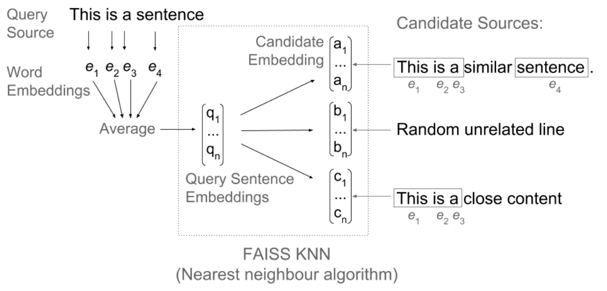

Figure 1: Procedure to obtain nearest neighbours with sentence embeddings

corresponding to the unedited regions. For these alignments, we used models trained with fast align (Dyer et al, 2013) based solely on generic data.

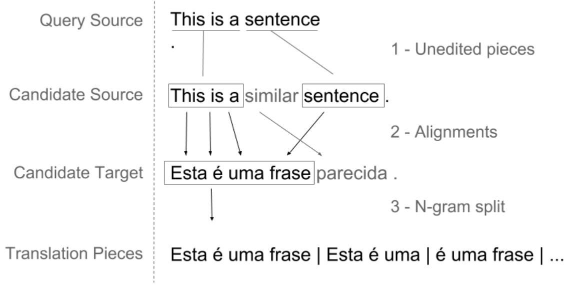

Figure 2: Process to obtain translation pieces with unedited words in nearest neighbours

We then merge the target regions obtained and generate as many translation pieces as possible, by breaking them into n-grams, up until a length of 4, as shown in Figure 2. Then, a score $s(X_q, X_c)$ is computed per sentence and associated with each generated piece, as shown in equation 1, where ed stands for the edit distance between the candidate and query sentences. We use the scoring method from (Zhang et al, 2018), and pick the maximum score $s(u)$ from all sentences featuring the translation piece u, as shown in equation 2. To apply the pieces in the beam search for a word w, we use a factor λ as an added weight, as shown in equation 3. For each hypothesis in the beam and each word, the set G consists of all pieces ending in word w.

$$s(X_q, X_c) = 1 - \frac{ed(X_q, X_c)}{max(|X_q|, |X_c|)} \tag{1}$$

$$s(u) = \max_{\{1<m<M, u \in X_c^m\}} s(X_q, X_c^m) \tag{2}$$

$$p_{beam}(w) = p_{beam}(w) + \lambda * \sum_{u \in G}(s(u)) \tag{3}$$

[2]English embeddings downloaded from wiki-news-300d-1M.vec.zip in https://fasttext.cc/docs/en/english-vectors.html

2.4 Implementation Details

The process of retrieving the translation pieces can be kept completely separate from the model. However, to integrate these in the generated translation, we require the model to be able to integrate these scores with its own.

Since in Marian, the produced scores from different scorers are combined at each time step t, before the beam is searched, we integrate the method with a new scorer module. This block can access the pieces and their scores for each sentence, and outputs the corresponding score delta for each word. It additionally allows for its weight to be used as the factor λ of the original method.

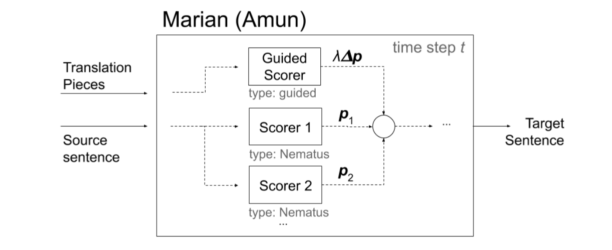

Figure 3: Amun implementation of translation pieces through a new scorer

Figure 3 provides a scheme of the implementation. Although we use only one scorer throughout the experiments presented, we show in this image a combination of several scorers of type `Nematus` to represent the possibilities of combining the implementation with other available features.

The combined scores will then be used in the beam search, with the effect of guiding the probabilities towards the translation pieces suggested. The factor λ must then be tuned to avoid overusing the provided pieces. Additionally, in our implementation [3] we introduce an extra parameter to this scorer, a threshold for the similarity that will disregard low-similarity pieces.

2.5 Tuning process

As mentioned in the previous section, we have only two model parameters that control the translation pieces implementation - the weight λ and a similarity threshold. To tune them, we run a grid search on both parameters, and keep the best performing pair as evaluated by a set of automatic metrics in our validation set. For simplicity, this process is ran only at the end of the initial model training.

The chosen automatic metrics follow. We pick as main metric BLEU, since it is a standard metric in machine translation. We also pick OTEM (Over Translation Evaluation Metric) and UTEM (Under Translation Evaluation Metric), since they have shown strong correlation with human evaluation (Yang et al, 2018) (Malaviya et al, 2018). Moreover, we believe that one of the possible caveats of the translation pieces could be related to overusage of these pieces, and these metrics can help detect that.

3 Results

To understand the impact of the translation pieces method, we first pick three language pairs: EN→FR, EN→NL and EN→RO. We chose data from email customer support, which we will consider our domain, and use available translation memories as base for the translation pieces retrieval. This experiment focuses on the impact on one single domain, and its results are presented on section 3.1. We then run a second experiment for EN→FR, EN→DE, EN→ES, EN→IT and EN→PT, where we analyze the impact of the translation pieces when used to fine tune subdomains, which we present in section 3.2. Appendix A presents additional information on the generic models used to fine tune both experiments.

We present results on the previously mentioned metrics – BLEU, OTEM and UTEM. We also present human evaluation on a subset of the test set. For this purpose we compiled 15 documents, making up a sample of 150 to 200 lines of data per set. One professional linguist annotated the errors in the data for each language. This gave us a breakdown of the most common errors for each model, and also a Multidimensional Quality Metric (MQM) score (Lommel et al, 2014)[4].

We use for these experiments the model architecture described in Section 2.1, with the fine tuning process listed in Section 2.2. We then apply the translation pieces method as described in Sections 2.3 and 2.4. Finally, we use the tuning process mentioned in 2.5 to obtain thresholds. All results presented correspond to the domain adap-

[3]https://github.com/CatarinaSilva/marian/guided-translation-scoring. Currently available only for Amun for CPU, but will be developed in the future for GPU and marian-decoder

[4]A definition can be consulted in the following link; http://www.qt21.eu/mqm-definition/definition-2015-12-30.html

tation baseline and the evaluation result using the attained thresholds.

3.1 Domain Adaptation

For the first set of experiments we gathered the available in-domain lines, and queried our available Translation Memories for the same domain. We present the resulting number of lines for each language pair in table 1.

	Train	Dev	Test	TMs
$D_{EN \rightarrow FR}$	43.69K	1004	1002	1852
$D_{EN \rightarrow NL}$	82.71K	1002	1001	1217
$D_{EN \rightarrow RO}$	73.15K	1000	1006	337

Table 1: Number of in-domain lines available for the training of models in each language pair

We ran the first experiment for the EN→FR language pair, a language pair with a strong baseline – around 65 BLEU points and 78 MQM points. The results can be seen in Table 2.

	BLEU	OTEM	UTEM	MQM
DA	**65.21**	**0.74**	**32.97**	77.39
DA + TPs	64.31	0.77	35.22	**82.58**

Table 2: Comparison of domain adaptation (DA) with added translation pieces (DA + TPs) for EN-FR

It is possible to see that, even though the automatic metrics do not reflect that, and they seem to point to the existence of more over and under translation, the human evaluation score shows improvements over the baseline.

	Macro MQM	Micro MQM
DA	78.2	77.39
DA + TPs	**82.62**	**82.58**

Table 3: Macro and Micro MQM of evaluated jobs for EN-FR

In Table 3 we show both the micro and macro MQM scores. The first presents the calculated value over the full analyzed set, similar to considering the full annotated data as one single document. The later presents an average of MQM scores for each document. We can see that both are very close to each other, which leads us to believe there are no particular outliers pulling the average up or down in the macro average.

The second experiment performed considered the EN→NL language pair, which has a weaker baseline, related both to a weaker baseline model and to the lack of availability of the same amount of domain data.

	BLEU	OTEM	UTEM	MQM
DA	35.05	**2.21**	59.96	48.23
DA + TPs	**41.82**	3.22	**55.38**	**53.07**

Table 4: Comparison of domain adaptation (DA) with added translation pieces (DA + TPs) for EN-NL

In this experiment both BLEU and UTEM improved significantly, as shown in Table 4, as well as the human evaluation metric. Aditionally, looking into the errors tagged by the linguists, we found that in particular errors concerning grammatical register and named entity errors decreased significantly.

	Macro MQM	Micro MQM
DA	47.52	48.23
DA + TPs	**56.32**	**53.07**

Table 5: Macro and Micro MQM of evaluated jobs for EN-NL

Regarding micro and macro averages, we noticed a wider spread than for the previous language pair, with a few outliers in the distribution, for example the existence with jobs of negative MQM. This can happen typically when a job is small and has the presence of a few major or critical errors or when a job has a huge amount of errors. Figure 4 presents the distribution of documents for both models.

The results in Table 5 show that the macro average increased significantly more than the micro average. Through 4 we see that the worst-scoring jobs were pulled to higher MQM values, which explains this impact. The most visible example is the lowest scoring job, that went from a negative value to a positive one, with a difference of about 30 MQM points.

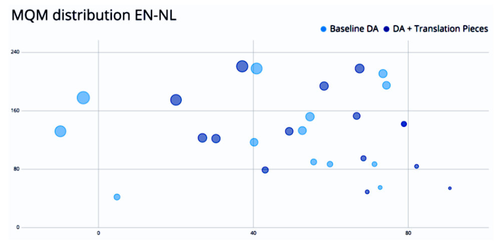

Figure 4: Distribution of MQM for EN→NL annotated jobs. The lighter dots are the jobs corresponding to the baseline, while the darker ones correspond to the model with translation pieces

The final language pair tested on this experiment was EN→RO, which is a language pair with lower resources, which leads to very low baselines, as we can see in tables 6 and 7.

	BLEU	OTEM	UTEM	MQM
DA	64.91	1.01	38.58	43.94
DA + TPs	**65.28**	**1.00**	**38.18**	**73.19**

Table 6: Comparison of domain adaptation (DA) with added translation pieces (DA + TPs) for EN-RO

For this language pair we see improvements in all metrics, with the human evaluation score raising 15 points of macro and 30 points of micro MQM. It is visible that the micro and macro averages are very different, which is linked to the existence of a lot of small jobs that resulted in very low MQMs, pulling the macro average down. Figure 5 shows the distribution of jobs, where it is possible to see that for the translation pieces the distribution is skewed to higher values, with only one exception.

	Macro MQM	Micro MQM
DA	16.25	43.94
DA + TPs	**31.86**	**73.19**

Table 7: Macro and Micro MQM of evaluated jobs for EN-RO

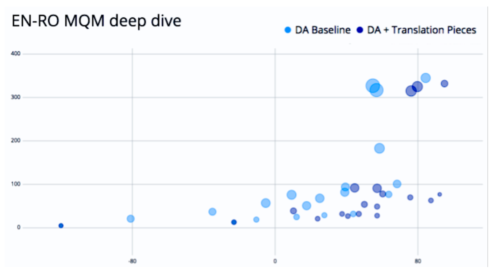

Figure 5: Distribution of MQM for EN→RO annotated jobs. The lighter dots are the jobs corresponding to the baseline, while the darker ones correspond to the model with translation pieces

Overall, for this experiment, the amount of jobs with less than 100 words had a big impact on macro MQMs, making it a bit hard to assess jobs comparatively or at a macro scale. However, all metrics suffered tremendous improvements, with significant drops in all errors, allowing us to clearly pick the translation pieces implementation over the baseline.

To summarize, for all language pairs, we see improvements in both macro and micro MQM, even when automatic metrics do not represent this improvement. We discuss this further in section 4.

3.2 Subdomain Distinction Impact

The second experiment aimed at comparing the usage of translation pieces of a particular domain to a direct adaptation in that particular subdomain. A subdomain can be seen as a smaller set of data inside the original domain that is more closely related. For example, if considering crisis data as a domain, more specific medical crisis data could be a subdomain inside of the first. In these, experiments, we keep the wide domain of email customer support, and consider different companies as subdomains.

For the first experiment, we reused the data from experiment 3.1, which we will consider as our subdomain (SD). We ran the domain adaptation process for it and for its parent domain (D) and ran the translation pieces code on top of both. The results are shown in Table 8. Interestingly, the baseline with the wider domain performs better in all automatic metrics, even though the human evaluation does not corroborate that.

	BLEU	OTEM	UTEM	MQM
D	**69.34**	**0.67**	**28.64**	76.46
D + TPs	68.14	1.55	31.37	81.07
SD	65.21	0.74	32.97	77.39
SD + TPs	64.31	0.77	35.22	**82.58**

Table 8: Comparison of translation pieces on top of Domain Adaptation (D) and subdomain adaptation (SD) for EN→FR

In fact, the results show that using translation pieces directly on the subdomain performs better as measured by human evaluation. Additionally, using a wider domain with translation pieces already surpasses the baseline for the subdomain. We consider the latter result of great interest since, if it holds for other subdomains, it would mean that the same baseline model can perform better than several subdomain models. Table 9 discriminates the micro and macro scores for MQM, both supporting the previous observations.

We then completed this set of experiments by picking several subdomains inside a known domain and using translation pieces on each. These domains vary in size, but neither performed better when adapting directly on the subdomain than the wider domain model. The goal was to understand if the previous behaviour holds, that is, if we can make a domain model perform better in its

	Macro MQM	Micro MQM
D	76.8	76.46
D + TPs	80.55	81.07
SD	78.2	77.39
SD + TPs	**82.62**	**82.58**

Table 9: Macro and Micro MQM of evaluated jobs for a domain and subdomain in EN-FR

subdomains through translation pieces. Table 10 presents the average micro and macro MQM variation attained in each Language Pair. The break down of these results can be seen in appendix B.

	ΔMacro-MQM	ΔMicro-MQM
EN$\rightarrow$DE	-0.74	+ 0.84
EN$\rightarrow$FR	+ 0.47	+ 1.77
EN$\rightarrow$ES	+ 2.92	+ 2.77
EN$\rightarrow$IT	+ 2.90	+ 1.85
EN$\rightarrow$PT	+ 3.78	+ 3.51

Table 10: Average difference across experiment subdomains for different language pairs: Δ = MQM$_{TP} - MQM_{Base}$

The results show a positive trend over most language pairs. We consider this an encouraging result that seems to support our previous hypothesis. However, even though we seem to attain there are slight variations over different domains in the tested language pairs. We leave as future work a more thorough analysis of the baseline quality of our translation memories and its relation with these variations, as discussed in the next section.

4 Discussion

Overall the experiments show that the use of translation pieces brings benefit to domain adaptation. In particular, the results for subdomain distinction are very promising, opening an easier path for a wider model to improve over its contained subdomains. We further discuss some caveats and future work regarding these experiments.

We used BLEU, OTEM and UTEM as metrics for both tuning and evaluating the presented method. However, we see throughout several experiments that these metrics seem to either contradict or under represent the improvements seen with human evaluation. If this is the case, a possible caveat is that we are tuning our threshold with sub-optimal metrics.

We hypothesise that these metrics might suffer from the fact that they are single reference. This might clash with the fact that our domain data contains a high number of repetition on the source side, thus presenting a lot of different variations of the same translation. Since our test set is just a slice of this pool of jobs, we might be overweighting a specific variation present in the test set of the sources, penalizing good variations produced by the different models.

We propose that in future work, the usage of multiple references in evaluation should be studied (Fomicheva and Specia, 2016) (Dreyer and Marcu, 2012). We believe this might lead to more reliable scores, and align better with human evaluation.

Another important factor that we do not present on these experiments is the quality of the translation pieces and its direct link to the quality of the results. We suspect that by having a better control of the quality of the pool of translation memories used, even if reducing its size, the performance of the method should improve even further, and at most should have as lower bound the quality of the baseline model.

In future work, we want to assess the quality of the used translation memories and compare the method in a scenario where we use only subdomains with available quality data. In this setting, the expected behavior would be for these subdomains to improve, without hurting other subdomains that might lack the same amount or quality of data. For this purpose, an analysis by professional linguists is necessary, both to produce a baseline value that can be related to the presented results, but also as a data selection procedure so that we can run experiments on gold data.

Finally, we want to extend the analysis to the reasons that cause the translation quality to improve with translation pieces. We suspect that the method is improving the translation of rare words, but also increasing the agreement and consistency of the translations, in particular with specific terminology. We plan to investigate this hypothesis to gain better understanding of this method.

Acknowledgements

We would also like to thank the support provided by the European Union in the context of the PT2020 project: 038510. We would additionally like to thank André Martins and Maxim Khalilov from Unbabel AI and the Unbabel Quality Team for their help with the human evaluation.

References

Arthur, Philip and Neubig, Graham and Nakamura, Satoshi 2016 Incorporating discrete translation lexicons into neural machine translation. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing 15571567 Association for Computational Linguistics

Chu, Chenhui and Wang, Rui 2018 In Proceedings of of the 27th International Conference on Computational Linguistics 1304-1319

Dreyer, Markus and Marcu, Daniel 2012 Hyter: Meaning-equivalent semantics for translation evaluation. *In 2012 Conference of the North American Chapter of the ACL: Human Language Technologies* 162171

Dyer, Chris and Chahuneau, Victor and Smith, Noah A. 2013 A simple, fast, and effective reparameterization of IBM Model 2 In Proc. NAACL

Fomicheva, Marina and Specia, Lucia 2016 Reference Bias in Monolingual Machine Translation Evaluation *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* 77–82 Association for Computational Linguistics https://www.aclweb.org/anthology/P16-2013

Johnson, Jeff and Douze, Matthijs and Jégou, Hervé 2017 Billion-scale similarity search with GPUs *CoRR* abs/1702.08734

Joulin, Armand and Grave, Edouard and Bojanowski, Piotr and Douze, Matthijs and Jégou, Hérve and Mikolov, Tomas 2016 FastText.zip: Compressing text classification models arXiv:1612.03651

Junczys-Dowmunt, Marcin and Grundkiewicz, Roman and Dwojak, Tomasz and Hoang, Hieu and Heafield, Kenneth and Neckermann, Tom and Seide, Frank and Germann, Ulrich and Fikri Aji, Alham and Bogoychev, Nikolay and Martins, André F. T. and Birch, Alexandra 2018 Marian: Fast Neural Machine Translation in C++ *Proceedings of ACL 2018, System Demonstrations* 116–121 Association for Computational Linguistics

Kaiser, Łukasz and Nachum, Ofir and Roy, Aurko and Bengio, Samy 2017 Learning to remember rare events

Koehn, Philipp and Knowles, Rebecca 2017 Six challenges for neural machine translation *In Proceedings of the First Workshop on Neural Machine Translation* 2839

Lommel, Arle and Uszkoreit, Hans and Burchardt, Aljoscha 2014 Multidimensional Quality Metrics (MQM): A Framework for Declaring and Describing Translation Quality Metrics

Malaviya, Chaitanya and Ferreira, Pedro and Martins, André F. T. 2018 Sparse and Constrained Attention for Neural Machine Translation *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* 370–376

Mikolov, Tomas and Grave, Edouard and Bojanowski, Piotr and Puhrsch, Christian and Joulin, Armand 2018 Advances in Pre-Training Distributed Word Representations *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*

Sánchez-Gijón, Pilar and Moorkens, Joss and Way, Andy 2019 Post-editing neural machine translation versus translation memory segments *Machine Translation* 1–29 Association for Computational Linguistics

Sennrich, Rico and Firat, Orhan and Cho, Kyunghyun and Birch, Alexandra and Haddow, Barry and Hitschler, Julian and Junczys-Dowmunt, Marcin and Läubli, Samuel and Miceli Barone, Antonio Valerio and Mokry, Jozef and Nadejde, Maria 2017 Nematus: a Toolkit for Neural Machine Translation *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics* 65–68

Sennrich, Rico and Haddow, Barry and Birch, Alexandra 2016 Neural Machine Translation of Rare Words with Subword Units *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* 1715–1725 Association for Computational Linguistics https://www.aclweb.org/anthology/P16-1162

Sennrich, Rico and Haddow, Barry and Birch, Alexandra 2016 Edinburgh Neural Machine Translation Systems for WMT 16 *Proceedings of the First Conference on Machine Translation* 371–376 Association for Computational Linguistics https://www.aclweb.org/anthology/W16-2323

Wang, Yuguang and Cheng, Shanbo and Jiang, Liyang and Yang,Jiajun and Chen,Wei and Li, Muze and Shi, Lin and Wang, Yanfeng and Yang, Hongtao 2017 Sogou neural machine translation systems for wmt17 *In Proceedings of the Second Conference on Machine Translation* 410415

Yang, Jing and Zhang, Biao and Qin, Yue and Zhang, Xiangwen and Lin, Qian and Su, Jinsong 2018 Otem&Utem: Over- and Under-Translation Evaluation Metric for NMT *NLPCC*

Zhang, Jingyi and Utiyama, Masao and Sumita, Eiichro and Neubig, Graham and Nakamura, Satoshi 2018 Guiding Neural Machine Translation with Retrieved Translation Pieces *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)* 1325–1335 Association for Computational Linguistics https://www.aclweb.org/anthology/N18-1120

Appendix A: Additional information on generic data

Below follows a list of datasets used to compile the generic models for the presented experiments in this work:

- Books, DGT, ECB, EMEA, EUbookshop, Europarl, EUConst, giga-fren, GlobalVoices, GNOME, JRC-Acquis, KDE4, MultiUN, News-Commentary, SETTIMES, Tanzil, Tatoeba, TED2013, Ubuntu and Wikipedia (from http://opus.nlpl.eu/)

- CommonCrawl (from https://www.statmt.org/wmt13/translation-task.html)

- Paracrawl (from https://paracrawl.eu/releases.html)

- Rapid Corpus of EU press releases (from https://www.statmt.org/wmt18/translation-task.html) – Rapid

These corpora do not hold the same amount of data for all language pairs. The specific sets can be consulted in the links provided. Tables 11 and 12 present the corpora used for each language pair.

EN→	FR	DE	ES	PT
EUbookshop	x	x	x	x
DGT		x		x
Europarl	x	x	x	x
JRC-Acquis		x		x
EMEA		x		x
ECB		x		x
MultiUN	x	x	x	
GNOME				x
KDE4				x
GlobalVoices		x	x	x
News-Commentary	x	x	x	
Books		x	x	x
Rapid		x		
Ubuntu				x
TED2013		x	x	x
Tanzil				x
Wikipedia			x	x
Tatoeba			x	x
EUConst	x		x	x
CommonCrawl	x			x
giga-fren v2	x			

Table 11: Used corpora for training of generic models

EN→	IT	NL	RO
EUbookshop	x	x	x
DGT	x	x	x
Europarl	x	x	x
JRC-Acquis	x	x	x
EMEA	x	x	x
ECB	x	x	
GNOME	x		
KDE4	x		
GlobalVoices	x		
News-Commentary	x	x	
Books	x	x	
Ubuntu	x		
TED2013	x	x	x
Tanzil	x		
Wikipedia			x
Paracrawl			x
SETTIMES			x

Table 12: Used corpora for training of generic models

Table 13 presents the resulting number of lines, after joining all datasets presnted in the aforementioned tables.

	Train	Dev	Test
$EN \rightarrow FR$	32.42M	1500	1500
$EN \rightarrow DE$	18.38M	1500	1500
$EN \rightarrow ES$	35.97M	1500	1500
$EN \rightarrow IT$	13.97M	1500	1500
$EN \rightarrow PT$	14.03M	1500	1500
$EN \rightarrow NL$	13.38M	1500	1500
$EN \rightarrow RO$	6.97M	1999	1999

Table 13: Number of lines used for training of generic models

Appendix B: Break down of subdomain results

	Baseline	TPs	Δ
Domain A	72.47	78.76	+ 6.29
Domain B	81.27	77.36	- 3.91
Domain C	90.53	85.93	-4.6
Overall	81.42	80.68	-0.74

Table 14: Macro MQM subdomain evaluation for EN→DE

	Baseline	TPs	Δ
Domain A	67.92	76.41	+ 8.49
Domain B	79.61	78.93	- 0.68
Domain C	89.35	84.07	- 5.28
Overall	78.96	79.80	+ 0.84

Table 15: Micro MQM subdomain evaluation for EN→DE

	Baseline	TPs	Δ
Domain A	88.48	85.12	- 3.36
Domain B	57.62	57.61	- 0.01
Domain C	89.12	93.89	+ 4.77
Overall	78.41	78,87	+ 0.47

Table 16: Macro MQM subdomain evaluation for EN→FR

	Baseline	TPs	Δ
Domain A	87.24	86.38	- 0.86
Domain B	60.62	60.02	- 0.60
Domain C	86.16	92.93	+ 6.77
Overall	78.01	79.78	+ 1.77

Table 17: Micro MQM subdomain evaluation for EN→FR

	Baseline	TPs	Δ
Domain A	34.07	46.94	+ 12.87
Domain B	67.31	70.56	+ 3.25
Domain C	61.22	57.62	- 3.6
Domain D	54.54	53.71	- 0.83
Overall	54.29	57.21	+ 2.92

Table 18: Macro MQM subdomain evaluation for EN→IT

	Baseline	TPs	Δ
Domain A	34.36	45.88	+ 11.52
Domain B	61.04	64.0	+ 2.96
Domain C	59.62	58.46	-1.16
Domain D	55.69	53.47	-2.22
Overall	52.33	55.45	+ 2.77

Table 19: Micro MQM subdomain evaluation for EN→IT Micro MQM

	Baseline	TPs	Δ
Domain A	67.09	66.07	- 1.02
Domain B	52.62	60.20	+ 7.58
Domain C	49.92	52.07	+ 2.15
Overall	56.54	59.45	+ 2.9

Table 20: Macro MQM subdomain evaluation for EN→ES

	Baseline	TPs	Δ
Domain A	74.50	72.22	- 2.28
Domain B	52.24	60.39	+ 8.15
Domain C	54.64	54.96	+ 0.32
Overall	60.46	62.52	+ 1.85

Table 21: Micro MQM subdomain evaluation for EN→ES

	Baseline	TPs	Δ
Domain A	71.45	72.90	+ 1.45
Domain B	60.45	64.89	+ 4.44
Domain C	88.65	87.79	- 0.86
Domain D	58.48	68.55	+ 10.07
Overall	69.76	73.53	+ 3.78

Table 22: Macro MQM subdomain evaluation for EN→PT

	Baseline	TPs	Δ
Domain A	76.58	73.90	- 2.68
Domain B	55.1	60.19	+ 5.09
Domain C	83.75	83.60	- 0.15
Domain D	56.65	68.42	+ 11.77
Overall	68.02	71.53	+ 3.51

Table 23: Micro MQM subdomain evaluation for EN→PT