On the Development of Customized Neural Machine Translation Models

Mauro Cettolo, Roldano Cattoni, Marco Turchi
Fondazione Bruno Kessler, Trento, Italy
{cettolo,cattoni,turchi}@fbk.eu

Abstract

Recent advances in neural modeling boosted performance of many machine learning applications. Training neural networks requires large amounts of clean data, which are rarely available; many methods have been designed and investigated by researchers to tackle this issue. As a partner of a project, we were asked to build translation engines for the weather forecast domain, relying on few, noisy data. Step by step, we developed neural translation models, which outperform by far Google Translate. This paper details our approach, that - we think - is paradigmatic for a broader category of applications of machine learning, and as such could be of widespread utility.

1 Introduction

The field of machine translation (MT) has experienced significant advances in recent years thanks to improvements in neural modeling. On the one hand, this represents a great opportunity for industrial MT, on the other it also poses the great challenge of collecting large amounts of clean data, needed to train neural networks. MT training data are parallel corpora, that is collections of sentence pairs where a sentence in the source language is paired with the corresponding translation in the target language. Parallel corpora are typically gathered from any available source, in most cases the web, without much guarantees about quality nor domain homogeneity.

Over the years, the scientific community has accumulated a lot of knowledge on ways to address the problem of the quantitative and qualitative inadequacy of parallel data necessary to develop translation models. Among others, deeply investigated methods are: corpus filtering (Koehn et al., 2020), data augmentation such as data selection (Moore and Lewis, 2010; Axelrod et al., 2011) and back-translation (Bertoldi and Federico, 2009; Sennrich et al., 2016), model adaptation (Luong and Manning, 2015; Chu and Wang, 2018). They should be the working tools of anyone who has to develop neural MT models for specific language pairs and domains.

This paper reports on the development of neural MT models for translating forecast bulletins from German into English and Italian, and from Italian into English and German. We were provided with in-domain parallel corpora for each language pair but not in sufficient quantity to train a neural model from scratch. Moreover, from the preliminary analysis of data, the English side resulted noisy (e.g. missing or partial translations, misaligned sentences, etc.), affecting the quality of any pair involving that language. For this very reason, we focus on one of the pairs involving English we had to cover, namely Italian-English.

An overview of the in-domain data and the description of their analysis are given in Section 2, highlighting the issues that emerged. Section 3 describes the previously listed methods together with their employment in our specific use-case. Developed neural translation models are itemized in Section 4, where their performance are compared and discussed; our best models outperform by far Google Translate and some examples will give a grasp of the actual translation quality.

We think that our approach to the specific problem we had to face is paradigmatic for a broader category of machine learning applications, and we hope that it will be useful to the whole NLP scientific community.

2 Data

We were provided with two *csv* files of weather forecast bulletins, issued by two different forecast services that from here on are identified with the acronyms BB and TT. Each row of the BB csv contains, among other things, the text of the original bulletin written in German and, possibly, its translation into Italian and/or English; in the TT csv rows, the Italian bulletin is paired with its translation into German and/or English.

2.1 Statistics

BB Bulletins were extracted from the BB csv file and paired for any possible combination of languages. Each bulletin is stored on a single line but split in a few dozen fields; the average length of each field (about 18 German words) is appropriate for MT systems, which process long sentences with difficulty. Table 1 shows statistics of the training and test sets for the it-en language pair.

site	task	set	#seg	#src w	#trg w
BB	it-en	trn-nsy	30,957	626,211	505,688
		tst-nsy	20,000	376,553	298,560
		tot	50,957	1,002,764	804,248

Table 1: Statistics of the BB it-en benchmark. The label *nsy* will be clear after reading Section 3.2.

TT Bulletins were extracted from the TT csv file and paired for each language combination. Differently than the BB case, each TT bulletin was stored on a single line without any field split; since bulletins are quite long for automatic processing (on average 30 Italian words) and are the concatenation of rather heterogeneous sentences, we decided to segment them by splitting on strong punctuation. This requires a re-alignment of source/target segments because in general they differ in number. The re-alignment was performed by means of the *hunalign* sentence aligner[1](Varga et al., 2005). Table 2 shows statistics of the training and test sets for the it-en language pair.

site	task	set	#seg	#src w	#trg w
TT	it-en	trn	5,177	78,834	73,763
		tst	1,962	30,232	28,135
		tot	7,139	109,066	101,898

Table 2: Statistics of the TT it-en benchmark.

[1] github.com/danielvarga/hunalign

2.2 Analysis and Issues

As a good practice before starting the creation of MT models, data have been inspected and analyzed looking for potential problems. Several critical issues emerged, which are described in the following paragraphs.

Non-homogeneity of data - Since data originated from two distinct weather forecast services (BB and TT), first of all it must be established whether they are linguistically similar and, if so, to what extent. For this purpose, focusing on the languages of the it-en benchmarks, we measured the perplexity of the BB and TT test sets on n-gram language models (LMs) estimated on the BB and TT training sets:[2] the closer the perplexity values of a given text on the two LMs, the greater the linguistic similarity of BB and TT training sets. Table 3 reports values of perplexity (PP) and out-of-vocabulary rates (%OOV) for all test sets vs. LMs combinations.[3]

		LM trained on			
		BB trn		TT trn	
		PP	%OOV	PP	%OOV
it	BB tst	10.8	0.22	92.0	12.07
	TT tst	42.4	0.60	10.3	0.41
en	BB tst	8.9	0.14	80.1	8.49
	TT tst	65.6	2.05	12.7	0.51

Table 3: Cross comparison of BB and TT texts.

Overall, we can notice that the PP of the two test sets significantly varies when computed on in- and out-of-domain data. The PP of any given test set is 4 (42.4 vs. 10.8) to 9 (92.0 vs. 10.3) times higher when measured on the LM estimated on the text of the other provider than on the text of the same provider. These results highlight the remarkable linguistic difference between the bulletins issued by the two forecast services.

In-domain data scarcity - Current state-of-the-art MT neural networks (Section 4.1) have dozens to hundreds million parameters that have to be estimated from data. Unfortunately, the amount of provided data does not allow an effective estimation from scratch of such a huge number of parameters, as we will empirically prove in Section 4.3.

[2] 3-gram LMs with modified shift beta smoothing were estimated using the IRSTLM toolkit (Federico et al., 2008).

[3] In order to isolate the genuine PP of the text, the dictionary upperbound to compute OOV word penalty was set to 0; the OOV rates are shown for this very reason.

BB English side - BB data have a major problem on the English side. In fact, looking at csv file, we realized that many German bulletins were not translated at all into English. Moreover, in the English side there are 20% fewer words than in the corresponding German or Italian sides, a difference that is not justified by the morpho-syntactic variations between languages. In fact, it happens that entire portions of the original German bulletins are not translated into English, or that a definitely more compact form is used, as in:

de: *Der Hochdruckeinfluss hält bis auf weiteres an.*
en: *High pressure conditions.*

This critical issue affects both training and test sets, as highlighted by figures in Table 1; as such, it negatively impacts both the quality of the translation models, if trained/adapted on such noisy data, and the reliability of evaluations, if run on such distorted data. A careful corpus filtering is therefore needed, as discussed in Section 3.2.

3 Methods

3.1 MT Model Adaptation

A standard method for facing the in-domain data scarcity issue mentioned in Section 2.2 is the so-called *fine-tuning*: given a neural MT model trained on a large amount of data in one domain, its parameters are tuned by continuing the training using a small amount of data from another domain (Luong and Manning, 2015; Chu and Wang, 2018). Though effective on the new in-domain data supplied for model adaptation, fine-tuning typically suffers from performance drops on unseen data (test set), unless proper regularization techniques are adopted (Miceli Barone et al., 2017). We avoid overfitting by fine-tuning our MT models with dropout (set to 0.3) (Srivastava et al., 2014) and performing only a limited number of epochs (5) (Miceli Barone et al., 2017).

3.2 Corpus Filtering

Machine learning typically requires large sets of clean data. Since rarely large data sets are also clean, researchers devoted much effort to data cleaning, the automatic process to identify and remove errors from data. The MT community is no exception. Even, WMT - the conference on machine translation - in 2018, 2019 and 2020 editions organized a Shared Task on Parallel Corpus Filtering. Koehn et al. (2020) provide details on the task proposed in the more recent edition, on participants, their methods and results. For reference purposes, organizers set up a competitive baseline based on LASER (Language-Agnostic SEntence Representations)[4] (Schwenk and Douze, 2017) multilingual sentence embeddings. The underlying idea is to use the cosine distance between the embeddings of the source and the target sentences to measure their parallelism. In a similar way we cleaned the BB noisy benchmark, filtering with a threshold of 0.9; statistics of the resulting bi-text are given in Table 4.

site	task	set	#seg	#src w	#trg w
BB	it-en	trn-cln	1,673	37,629	40,256
		tst-cln	1,011	20,280	21,657
		tot	2,684	57,909	61,913

Table 4: Stats of the filtered BB it-en benchmark.

The filtered bi-text does not suffer anymore from the imbalance number of words but it is 20 times smaller than the original one.

3.3 Data Augmentation

Since the corpus filtering discussed in the previous section removes most of the original data, further exacerbating the problem of data scarcity, we tried to overcome this unwanted side effect by means of data augmentation methods.

3.3.1 Data Selection

A widespreadly adopted data augmentation method is *data selection*. Data selection assumes the availability of a large general domain corpus and a small in-domain corpus; in MT, the aim is to extract parallel sentences from the large bilingual corpus that are most relevant to the target domain as defined by the small corpus.

On the basis of the bilingual cross-entropy difference (Axelrod et al., 2011), we sorted the sentence pairs of the OPUS collection,[5] used as general domain large dataset, according to their relevance to the domain determined by the concatenation of the BB and TT training sets. To establish the optimal size of the selection, we trained LMs - created in the same setup described in *non-homogeneity of data* paragraph of Section 2.2 - on increasing amounts of selected data and computed the PP of BB and TT test sets, separately for each side. Figure 1 plots the curves; the straight lines on

[4] github.com/facebookresearch/LASER
[5] opus.nlpl.eu

the bottom correspond to the PP of the same test sets on LMs built on the in-domain training sets.

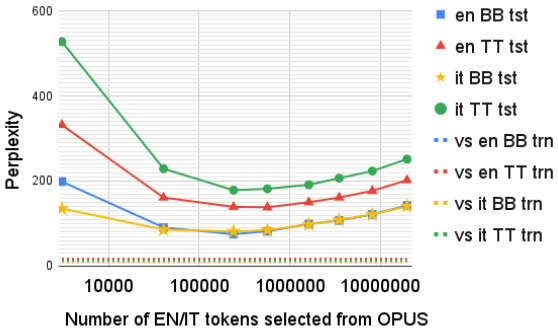

Figure 1: Perplexity of test sets on LMs estimated on increasing amounts of selected data.

The form of curves is convex, as usual in data selection. In our case, the best trade-off between the pertinence of data and its amount occur when something more than a million words is selected; therefore, we decided to mine from OPUS the bilingual text whose size is given in row *DS* of Table 5. Anyway, note that the lowest PP for selections is at least one order of magnitude greater than on LMs trained on in-domain training sets.

task	set	#seg	#src w	#trg w
it-en	DS	206,990	1,352,623	1,312,068
	BT	30,957	482,398	505,688

Table 5: Stats of selected and back translated data.

3.3.2 Back Translation

Another well-known data-augmentation method, which somehow also represents an alternative way to corpus filtering for dealing with the BB English side issue, is *back-translation*. Back-translation (Bertoldi and Federico, 2009; Sennrich et al., 2016; Edunov et al., 2018) assumes the availability of an MT system from the target language to the source language and of target monolingual data. The MT system is used to translate the target monolingual data into the source language. The result is a parallel corpus where the source side is the synthetic MT output while the target is human text. The synthetic parallel corpus is then used to train or adapt a source-to-target MT system. Although simple, this method has been shown to be very effective. We used back-translation to generate a synthetic, but hopefully cleaner, version of the BB training set. The translation into Italian of the 31k English segments of the training set (Table 1) was performed by an in-house generic en-it MT engine (details in Appendix A.1 of (Bentivogli et al., 2021)). Row *BT* of Table 5 shows the statistics of this artificial bilingual corpus; similarly to what happened with the filtering process, the numbers of Italian and English words are much more compatible than they are in the original version of the corpus.

	#segments	#src w	#trg w
it-en	32.0M	339M	352M

Table 6: Stats of the parallel generic training sets.

4 Experimental Results

4.1 MT Engine

The MT engine is built on the ModernMT framework[6] which implements the Transformer (Vaswani et al., 2017) architecture. The original generic model is *Big* sized, as defined in (Vaswani et al., 2017) by more than 200 million parameters. For training, bi-texts were downloaded from the OPUS repository[5] and then filtered through the already mentioned data selection method (Axelrod et al., 2011) using a general-domain seed. Statistics of the resulting corpus are provided in Table 6. Training was performed in the setup detailed in (Bentivogli et al., 2021).

The same Big model and its smaller variants, the *Base* with 50 million parameters and the *Tiny* with 20 million parameters, were also trained on in-domain data only for the sake of comparison.

4.2 MT Models

We empirically compared the quality of translations generated by various MT models: two generic, three genuine in-domain of different size and several variants of our generic model adapted (Section 3.1) on in-domain data resulting from the presented methods: filtering (Section 3.2), data selection (Section 3.3.1) and back-translation (Section 3.3.2). Performance was measured on the BB and TT test sets in terms of BLEU (Papineni et al., 2002), TER (Snover et al., 2006) and CHRF (Popović, 2015) scores computed by means of SacreBLEU (v1.4.14) (Post, 2018), with default

[6] github.com/modernmt/modernmt

MT model	BB						TT		
	noisy test set			clean test set			test set		
	%BLEU↑	%TER↓	CHRF↑	%BLEU↑	%TER↓	CHRF↑	%BLEU↑	%TER↓	CHRF↑
Generic models:									
GT⋆	11.45	106.61	.3502	32.59	51.72	.6104	32.20	61.56	.6315
FBK (Transformer_big)	7.43	113.07	.3833	19.68	63.68	.5229	23.45	70.46	.5525
Pure in-domain models trained on *BBtrn-nsy+TTtrn*:									
Transformer_tiny	23.34	83.86	.4882	35.80	61.05	.5808	42.19	51.79	.6488
Transformer_base	18.39	93.41	.4590	22.06	85.91	.5237	29.17	64.73	.5351
Transformer_big	20.45	95.76	.4755	24.73	89.26	.5330	28.01	68.42	.5193
FBK model adapted on:									
BBtrn-nsy	21.21[1]	80.82[2]	.4785[2]	37.91[3]	46.91[3]	.6172	13.77	79.14	.4007
BBtrn-cln	10.67	108.86	.4195	31.57	52.54	.5950	27.68	65.05	.5912
TTtrn	10.44	107.48	.4241	28.64	54.20	.5800	39.61	52.64	.6702
DS	10.82	109.71	.4255	30.11	54.86	.5873	29.76	63.68	.6099
BT	12.50	106.85	.4507	34.85	49.78	.6339	32.71	58.95	.6372
BBtrn-nsy+TTtrn	19.30[3]	79.29[1]	.4449	32.81	52.38	.5680	40.51[3]	51.97[3]	.6579
BBtrn-nsy+TTtrn+DS+BT	19.36[2]	86.33[3]	.4792[1]	41.17[1]	44.67[1]	.6488[2]	40.69[2]	51.84[2]	.6734[3]
BBtrn-cln+TTtrn	12.39	105.36	.4450	37.02	47.40	.6365[3]	40.34	52.16	.6755[2]
BBtrn-cln+TTtrn+DS+BT	13.75	104.59	.4619[3]	40.09[2]	45.28[2]	.6617[1]	41.16[1]	51.01[1]	.6803[1]

Table 7: BLEU/TER/CHRF scores of MT models on it-en test sets. [1], [2] and [3] indicate the "podium position" among the adapted models of each column. (⋆) Google Translate, as it was on 14 Sep 2021.

signatures.[7]

4.3 Results and Comments

Scores are collected in Table 7. First, as expected (*in-domain data scarcity* paragraph of Section 2.2), it is not feasible to properly train a huge number of parameters with few data; in fact, the best performing pure in-domain model is the smallest one (*Transformer_tiny*). Instead, the naive application of the MT state-of-the-art would have led to simply train a *Transformer_big* model on the original in-domain data. This model would not have been competitive with *GT* on TT data (28.01 vs. 32.20 BLEU); it would have been on BB data if we had only considered the noisy test set (20.45 vs. 11.45) resulting in an important misinterpretation of the actual quality of the two systems; conversely, our preliminary analysis allowed us to discover the need of cleaning BB data, which guarantees a reliable assessment (24.73 vs. 32.59).

Data augmentation methods (*DS*, *BT*) are both effective in making available additional useful bitexts; for example, the BLEU score of the model *BBtrn-cln+TTtrn* increases by 3 absolute points (from 37.02 to 40.09) when DS and BT data are added to the adaptation corpus.

The fine-tuning of a *Transformer_big* generic model to the weather forecast domain turned out to be more effective than any training from scratch using original in-domain data only: the top performing model - *BBtrn-cln+TTtrn+DS+BT* - definitely improves the *Transformer_tiny* with respect to all metrics on the BB clean test set (40.09/45.28/.6617 vs 35.80/61.05/.5808), and to two metrics out of three on the TT test set (TER: 51.01 vs. 51.79, CHRF: .6803 vs. .6488). Moreover, all its scores are a lot better than those of Google Translate.

4.4 Examples

To give a grasp of the actual quality of automatic translations, Table 8 collects the English text generated by some of the tested MT models fed with a rather complex Italian source sentence. The manual translations observed in BB data are shown as well: their number, their variety, some questionable/wrong lexical choices in them ("high" instead of "upper-level currents", "South-western" instead of "Southwesterly") and one totally wrong ("Weak high pressure conditions.") prove the difficulty of learning from such data and the need to pay par-

[7] BLEU+case.mixed+numrefs.1+smooth.exp+tok.13a, TER+tok.tercom-nonorm-punct-noasian-uncased, chrF2+numchars.6+space.false

`Italian source sentence:`	
	Le correnti in quota si disporranno da sudovest avvicinando masse d'aria più umida alle Alpi.
`Manual English translations found in BB bulletins:`	
	Weak high pressure conditions.
	The high currents will turn to south-west and humid air mass will reach the Alps.
	Southwesterly currents will bring humid air masses to South Tyrol.
	South-western currents will bring humid air masses to the Alps.
	South-westerly upper level flow will bring humid air masses towards our region.
	More humid air masses will reach the Alps.
	Humid air reaches the Alps with South-westerly winds.
`Automatic English translations generated by some MT models:`	
GT	The currents at high altitudes will arrange themselves from the southwest, bringing more humid air masses closer to the Alps.
FBK	Currents in altitude will be deployed from the southwest, bringing wet air masses closer to the Alps.
Transformer_tiny	South-westerly upper level flow will bring humid air masses towards the Alps.
BBtrn-cln+TTtrn+DS+BT	The upper level flow will be arranged from the southwest approaching more humid air masses to the Alps.

Table 8: Examples of manual and automatic translations.

ticular attention to the evaluation phase. Concerning translations, *GT* is able to keep most of the meaning of the source text but the translation is too literal to result in fluent English. *FBK* only partially transfers the meaning from the source and generates a rather bad English text. *Transformer_tiny* provides a very good translation both from a semantic and a syntactic point of view, losing only the negligible detail that the "air masses" are "more humid", not simply "humid". Finally, *BBtrn-cln+TTtrn+DS+BT*, the model that on the basis of our evaluations is the best one, on this specific example works very well at the semantic level but rather poorly on the grammatical level.

This example shows that pure in-domain models, as expected, are "more in-domain" than generic models, though adapted, showing greater adherence to domain-specific language. On the other hand, according to scores in Table 7, adapted models should be better in generalization. Only subjective evaluations involving meteorologists can settle the question of which model is the best.

5 Conclusions

In this paper we described the development process that led us to build competitive customized translation models. Given the provided in-domain data, we started by analyzing them under several perspectives and discovered that they are few, noisy and heterogeneous. We faced these issues by exploiting a number of methods which represent established knowledge of the scientific community: adaptation of neural models, corpus filtering and data augmentation techniques such as data selection and back-translation. In particular, corpus filtering allowed us to avoid the misleading results observed on the original noisy data, while adaptation and data augmentation proved useful in effectively taking advantage of out-of-domain resources.

References

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain Adaptation via Pseudo In-Domain Data Selection. In *Proc. of EMNLP*, pages 355–362, Edinburgh, Scotland, UK.

Luisa Bentivogli, Mauro Cettolo, Marco Gaido, Alina Karakanta, Alberto Martinelli, Matteo Negri, and Marco Turchi. 2021. Cascade versus Direct Speech Translation: Do the Differences Still Make a Difference? In *Proc. of ACL/IJCNLP (Volume 1: Long Papers)*, pages 2873–2887, Bangkok, Thailand.

Nicola Bertoldi and Marcello Federico. 2009. Domain Adaptation for Statistical Machine Translation with Monolingual Resources. In *Proc. of WMT*, pages 182–189, Athens, Greece.

Chenhui Chu and Rui Wang. 2018. A Survey of Domain Adaptation for Neural Machine Translation. In

Proc. of COLING, pages 1304–1319, Santa Fe, US-NM.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding Back-Translation at Scale. In *Proc. of EMNLP*, pages 489–500, Brussels, Belgium.

Marcello Federico, Nicola Bertoldi, and Mauro Cettolo. 2008. IRSTLM: An Open Source Toolkit for Handling Large Scale Language Models. In *Proc. of Interspeech*, pages 1618–1621, Brisbane, Australia.

Philipp Koehn, Vishrav Chaudhary, Ahmed El-Kishky, Naman Goyal, Peng-Jen Chen, and Francisco Guzmán. 2020. Findings of the WMT 2020 Shared Task on Parallel Corpus Filtering and Alignment. In *Proc. of WMT*, pages 726–742, Online.

Minh-Thang Luong and Christopher D. Manning. 2015. Stanford Neural Machine Translation Systems for Spoken Language Domains. In *Proc. of IWSLT*, pages 76–79, Da Nang, Vietnam.

Antonio Valerio Miceli Barone, Barry Haddow, Ulrich Germann, and Rico Sennrich. 2017. Regularization Techniques for Fine-tuning in Neural Machine Translation. In *Proc. of EMNLP*, pages 1489–1494, Copenhagen, Denmark.

Robert C. Moore and William Lewis. 2010. Intelligent Selection of Language Model Training Data. In *Proc. of ACL (Short Papers)*, pages 220–224, Uppsala, Sweden.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proc. of ACL*, pages 311–318, Philadelphia, US-PA.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proc. of WMT*, pages 392–395, Lisbon, Portugal.

Matt Post. 2018. A Call for Clarity in Reporting BLEU Scores. In *Proc. of WMT*, pages 186–191, Belgium, Brussels.

Holger Schwenk and Matthijs Douze. 2017. Learning Joint Multilingual Sentence Representations with Neural Machine Translation. In *Proc. of RepL4NLP*, pages 157–167, Vancouver, Canada.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving Neural Machine Translation Models with Monolingual Data. In *Proc. of ACL (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany.

Matthew Snover, Bonnie Dorr, Rich Schwartz, Linnea Micciulla, and John Makhoul. 2006. A Study of Translation Edit Rate with Targeted Human Annotation. In *Proc. of AMTA*, pages 223–231, Cambridge, US-MA.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958.

Dániel Varga, Péter Halácsy, András Kornai, Nagy Viktor, Nagy Laszlo, N. László, and Tron Viktor. 2005. Parallel Corpora for Medium Density Languages. In *Proc. of RANLP*, pages 590–596, Borovets, Bulgaria.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All You Need. In *Proc. of NIPS*, pages 5998–6008, Long Beach, US-CA.