NICE for Covid-19 translation task: Report

Zuzanna Parcheta, Daniel Marín Buj, and Anna Samiotou
name.first_surname@cdt.europa.eu

Translation Centre for the Bodies of the European Union

Abstract. In this document, we present a machine translation system developed by the Translation Centre for the Bodies of the European Union used for translation of covid-19-related texts from Covid-19 MLIA Eval. The developed translation system, previously described in Marín Buj et al. (2020), was adapted to generate prediction of texts related with Covid-19.

1 Project description

NICE: Neural Integrated Custom Engines was developed to provide raw machine translation of source texts that enable translators to produce final translations with much less effort than it would take to produce the same translations from scratch [3]. Also, the purpose is to keep maximum confidentiality in the inference process by assuring an adapted, on-premise infrastructure.

In Translation Centre we focus on four different domains: intellectual property (IP), public health (PH), legal and generic domain (GEN). Although the scope of the project includes all 24 official EU languages, each domain has its requirements in terms of language coverage. Thus, we targeted specific pairs for the development phase of the engines, with English being the common language for all models. To get more details about development of NICE, please consult Marían Buj et al. (2020) work [1].

2 Data available

In this section we describe data used for training of Covid-19 translation engines.

2.1 Constrained data

The shared task organisers provide Covid-19-related data which is described in Table 1.

Covid-19 MLIA @ Eval Initiative, http://eval.covid19-mlia.eu/.

Table 1. Constrained data provided by organisers. k denotes thousands of elements and M denotes millions of elements. $|S|$ stands for number of sentences.

Subset	$\|S\|$						
	de	el	es	fr	it	sv	ar
Train	1,5M	674k	2,8M	2,4M	1M	374k	424k
Dev	4k	4k	4k	4k	4k	4k	4k
Blind Test	4k	4k	4k	4k	4k	4k	4k

2.2 Unconstrained data

The Translation Centre owns data for all language pairs belong to existing domains: IP, PH, legal and generic material.

Within each domain, the data is organised depending of quality being 1 the most suitable with the following properties:

1. Validated translations from CdT translation memories.
2. Non-validated translations from CdT translation memories.
3. Verified sentence-based alignments from CdT legacy data.
4. Non-CdT data sources (public).
5. Synthetic data (CdT and non-CdT).

For low-resource pairs with English as source language, we generated synthetic data to enlarge the training corpora. The unconstrained data is described in Table2. GEN domain includes all Translation Centre data, and PH only sentences from PH domain.

Table 2. Unconstrained data used in experiments. k denotes thousands of elements and M denotes millions of elements. $|S|$ stands for number of sentences.

Subset	$\|S\|$						
	de	el	es	fr	it	sv	ar
GEN	20,5M	13,8M	27,6M	19,3M	16,3M	374k	-
PH	5M	1,6M	1,5M	1M	1,8M	1,4M	-

3 Data preparation

The data preparation process is as following:

1. Bilingual data for a given language pair is extracted from files.
2. The bilingual sentences where some anomaly was detected, such as sentence pairs with identical source and target, sentences without words, anomalous size ratios between source and target lengths etc. are removed from the dataset.

3. We deduplicate pairs when the same source was translated in different ways more than once keeping the most recent translation with the highest quality.
4. Too long sentences are removed. We used a parameter that indicates the percentage of sentences to keep by length. We applied a value of 0.99, which removes 1% of the longest sentences.
5. A byte-pair encoding model using the `sentencepiece` sub-word tokeniser [6] is trained and all data used for the training is tokenised using the trained model.
6. We make use of `fast_align` [2] to train alignment model trained on high-quality data (i.e. quality sets 1 and 2) for the corresponding language pair. `fast_align` allows an alignment model to be computed that contains negative log-likelihood between source and target words.
7. We apply clustering to clean sentences pairs which have to high distance to the cluster centre. The cluster centre is computed using sentences with the highest quality, usually 1.
8. Data noising techniques are applied to avoid over-fitting.

4 Training

The architecture used to train our models is TransformerBig, a large transformer network based on Vaswani et al. (2017) [8].

All our engines are built with `OpenNMT-tf` [5], which is an open-source toolkit for NMT and neural sequence learning with a TensorFlow backend.

During the training process, we used Adam as optimisation method [4]. A dropout layer of 30% probability was applied and a weight decay value of 10^{-4}. We calculated the number of validation steps based on the size of the training data and considering a buffer of 500,000 shuffled sentences, including at least two validation cycles per epoch. That way, the validation steps depend on the corpora and the batch size, which is usually of 64 examples. We stored the last ten checkpoints and applied early stopping [7] with a patience value of five evaluations. Once the training was stopped, we averaged the five best stored models.

5 Implemented systems

To follow our existing preprocessing pipelines, we included the data provided by organisers into our own data. As the data is external, we included it into the data set of PH domain of quality 4. For Covid-19 shared tasks purpose, We followed 3 different strategies to generate predictions.

1. Train engine using constrained data: As the used data belongs exclusively to the quality 4, the preprocess pipeline skips the alignments ans clustering cleaning steps.
2. Generate translation using available Center's engines: During the engines creation process, we didn't owe covid provided data. The purpose of this system is to test our engines with external data.

3. Train generic engine using constrained+unconstrained GEN data and fine-tune on constrained + PH data: First, we train generic model using all available data and after, fine-tune the generic model with PH data.

6 Results

6.1 System 1

We submitted predictions using engines trained only with constrained data for English – {Spanish, Italian, Greek, French, Swedish, Arabic}. Results of generated predictions are shown in Table 3.

Table 3. Results of predictions for system 1.

source	en					
target	es	it	el	fr	sv	ar
BLEU	55.4	37.9	32.9	56.9	20.3	15.9
TER	34.1	51.9	56.6	34.6	75.3	77.9
BEER	74.6	62.5	59.0	74.5	46.5	48.7

6.2 System 2

We provided English – {Spanish, German, Italian, Greek, French, Swedish} translations. Results are shown in Table 5.

Table 4. Results of predictions for system 2.

source	en					
target	es	de	it	el	fr	sv
BLEU	51.4	34.9	45.2	37.5	49.7	21.3
TER	37.0	53.1	43.3	50.0	40.0	72.7
BEER	72.9	64.3	68.8	63.7	71.3	48.7

6.3 System 3

The system 3 required train from scratch engines using all available data and fine-tune it with PH data. Unfortunately, we only was able to train engine for English – Italian pair of languages before the deadline.

Table 5. Results of predictions for system 3.

source	en
target	it
BLEU	49.0
TER	39.9
BEER	70.5

References

[1] Buj, D.M., Ibáñez García, D., Parcheta, Z., Casacuberta, F.: NICE: Neural integrated custom engines. In: Proceedings of the 22nd Annual Conference of the European Association for Machine Translation, pp. 329–338, European Association for Machine Translation, Lisboa, Portugal (Nov 2020), URL https://aclanthology.org/2020.eamt-1.35

[2] Dyer, C., Chahuneau, V., Smith, N.A.: A simple, fast, and effective reparameterization of ibm model 2. In: Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 644–648 (2013)

[3] Jia, Y., Carl, M., Wang, X.: How does the post-editing of neural machine translation compare with from-scratch translation? a product and process study. The Journal of Specialised Translation pp. 60–86 (01 2019)

[4] Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)

[5] Klein, G., Kim, Y., Deng, Y., Senellart, J., Rush, A.M.: Opennmt: Open-source toolkit for neural machine translation. arXiv preprint arXiv:1701.02810 (2017)

[6] Kudo, T., Richardson, J.: Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. arXiv preprint arXiv:1808.06226 (2018)

[7] Prechelt, L.: Early stopping-but when? In: Neural Networks: Tricks of the trade, pp. 55–69, Springer (1998)

[8] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems, pp. 5998–6008 (2017)