NICE: Neural Integrated Custom Engines

Daniel Marín Buj[1], **Daniel Ibáñez García**[1], **Zuzanna Parcheta**[1], **Francisco Casacuberta**[2]
daniel.marin@cdt.europa.eu
{daniel.ibanez,zuzanna.parcheta}@ext.cdt.europa.eu
fcn@prhlt.upv.es

[1] Translation Centre for the Bodies of the European Union
[2] PRHLT Research Center, Universitat Politecnica de València

Abstract

In this paper, we present a machine translation system implemented by the Translation Centre for the Bodies of the European Union. The main goal of this project is to create domain-specific machine translation engines to support machine translation services and applications for the Translation Centre's clients. In this article, we explain the entire implementation process of NICE: Neural Integrated Custom Engines. We describe the problems identified and the solutions provided, and present the final results for different language pairs. Finally, we describe the work that will be done on this project in the future.

1 Project description

Set up in 1994, the Translation Centre for the Bodies of the European Union (CdT) delivers an average of 750,000 pages a year to over 60 European Union institutions, agencies and bodies across Europe. It has grown steadily, hand in hand with an increasing number of official European Union (EU) languages. To meet the needs of its clients and to cope with very specialised fields and growing translation volumes, the CdT has decided to enhance its services with state-of-the-art technologies such as neural machine translation (NMT) (Wu, 2016; Castilho, 2017).

The business goal of this project is to provide raw machine translation of source texts that enable translators to produce final translations that are indistinguishable from human translations with less effort than it would take to produce the same translations from scratch (Jia, 2019). Also, we aim to create engines that are fully integrated into CdT's translation management system and fine-tuned for specific needs, such as post-editing particular document types, which cannot be achieved by existing systems. Finally, the purpose is to keep maximum confidentiality in the inference process by assuring an adapted, on-premise infrastructure.

In this work, we focus on two different domains: intellectual property (IP) and public health (PH). 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.

The practice adopted for the development of machine translation engines included extensive preprocessing of data. After such data preparation, a generic model (GEN) was trained using data from all available domains. Then, the generic model was fine-tuned with in-domain data (IND). After training the IND model, we tested it using fixed test sets, and with five standard metrics. After the automatic evaluation against high-quality references, human translators assessed another set of representative samples by applying predefined metrics at a segment level, such as adequacy and fluency (Koehn, 2006), and by post-editing the raw output to measure the potential productivity gains (Levenshtein distance (Marg, 2016)). All steps of NMT engine creation will be explained in the following sections.

Martins, Moniz, Fumega, Martins, Batista, Coheur, Parra, Trancoso, Turchi, Bisazza, Moorkens, Guerberof, Nurminen, Marg, Forcada (eds.)
Proceedings of the 22nd Annual Conference of the European Association for Machine Translation, p. 329–338
Lisboa, Portugal, November 2020.

2 Data available

The available data for all language pairs belong to existing domains: IP, PH, other domains and generic material.

Within each domain, the data was split into an extendable number of sets depending on the quality for the purpose, ordered from the most suitable (1) to the less suitable (5), with each number reflecting the relative and presumed quality of the set, as follows:

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).

Each sentence pair extracted from the quality sets is linked to metadata labels indicating the date and the quality set number to which the pair belongs. This metadata was used in the preprocessing pipeline. Most of the data was parsed as TMX 1.4b; however, publicly available data was obtained in different formats such as plain text or other TMX versions.

For low-resource pairs with English as source language, we generated synthetic data to enlarge the training corpora. We consider low-resource language pairs when the IND dataset contains less than 150,000 bilingual sentences. The synthetic data consists of back translations of monolingual sets extracted from non-English language pairs, such as Croatian–French, being the Croatian the low-resource language in this case. As described in Koehn et al. (2017), successful applications of this idea used equal amounts of synthetic and true data to train the final system. However, generating this amount of synthetic data is not always possible. Besides, to generate synthetic data, a reverse translation system is required. Since most CdT machine translation engines are unidirectional from English, we used `eTranslation` platform for generating synthetic data via back-translation.[1]

[1] `https://ec.europa.eu/cefdigital/eTranslation`

`eTranslation` (Oravecz, 2019) is the European Commission's machine translation service, supported by the Connecting Europe Facility (CEF) and developed by the Directorate-General for Translation. Available engines can translate documents between all official EU languages and a few non-EU languages, providing quality machine translation in a secure system that protects privacy. As an EU body, the CdT contributes to its development and maintenance and has access to its platform. These engines were used for back-translation because of the high-quality output, as demonstrated in experiments to benchmark both `eTranslation` and our translation system.

To determine whether and to what extent synthetic data improves the quality of the PH engine, different experiments were conducted with the English–Croatian pair, using all available data and comparing the model trained with synthetic data (quality sets 1-5) against a baseline trained without any synthetic data (quality sets 1-4). The detailed amounts of data are shown in Table 1 (EN—HR). From Figure 1 we can appreciate that the GEN model obtains lower sacreBLEU (Post, 2018) with synthetic data. It can be due to the fact that the GEN model is large enough (1.4 millions of bilingual sentences) and the added 460,000 bilingual synthetic sentences are of lower quality than the original data from the generic model. The fact that the validation and test sets belong to IND makes the potential degradation of the GEN scores less relevant, as long as IND scores improve. In the case of the IND model, the sacreBLEU score of the model trained with synthetic data is around 0.5 points higher when using synthetic data in the GEN and IND models. The IND data contains only 83,000 sentences, so the addition of 33,000 bilingual synthetic sentences had a positive effect on the final score. The last experiment with synthetic data was the fine-tuning of the original GEN model (without synthetic data) with an IND dataset including synthetic data. The obtained sacreBLEU score was 52.4, which implies a reduction of 0.6 sacreBLEU points compared to 53 sacreBLEU points from the previous experiment. Therefore, the best approach found was to apply synthetic data to both GEN and IND models.

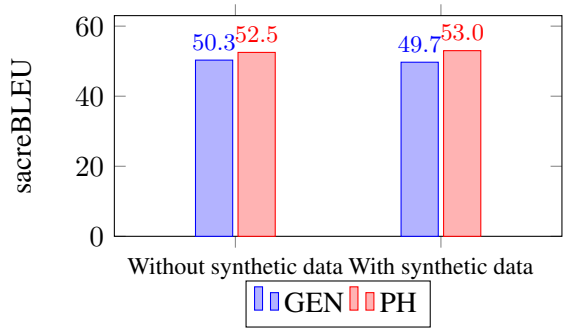

Figure 1: Comparison of the PH model output quality with and without synthetic data for English–Croatian using sacreBLEU.

3 Data preparation

In this section, we describe the entire data preparation process, which is as follows:

1. **Extraction of parallel sentences from TMX files**: we extracted translation units for the relevant languages from the available quality-graded sets.

2. **Cleaning of anomalous data**: we filtered out pairs according to different criteria, such as sentence pairs with identical source and target, sentences without words, or anomalous size ratios between source and target lengths.

3. **Deduplication**: we deduplicated pairs when the same source was translated in different ways more than once keeping the most recent translation with the highest quality. To keep the best pairs in the deduplication step, the quality labels described previously were used.

4. **Removal of oversized sentences**: to accommodate differences among languages, 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 sentences.

5. **Data normalisation**: we used regular expressions to protect numbers, URLs, emails, codes and certain acronyms, and replaced them with the corresponding token, e.g. numbers with ((NUMBER 0)), as described in Post et al. (2019). Where sentences contained several matches of the same pattern, we numbered each of them, e.g ((NUMBER 0)), ((NUMBER 1)).

6. **Vocabulary model training**: we trained a byte-pair encoding model using the `sentencepiece` sub-word tokeniser (Kudo, 2018), which omits the previously protected tokens.

7. **Training data encoding**: we tokenised training data with the `sentencepiece` model.

8. **Advanced data filtering**: we applied `fast_align` (Dyer, 2013) to train an alignment model on good 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. Once the alignment model was built, we scored the clean data and obtained a $z - score$ for each sentence. The scored bilingual sentences were normalised by the source length of each bilingual pair and the vector was standardised using Equation 1, where μ is the mean of scores and σ is its standard deviation.

$$z = \frac{x - \mu}{\sigma} \quad (1)$$

In Figure 2, the grey field represents the sentences below a fixed threshold that are filtered out; in this case, -1. Where the dataset was very small, we applied a lower threshold so the filtering would be more tolerant.

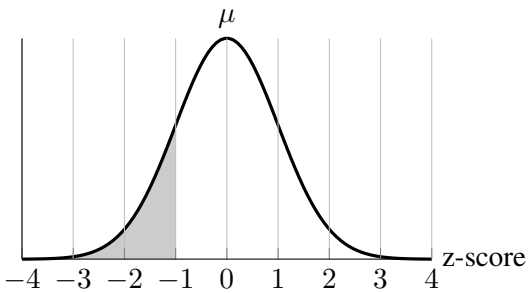

Figure 2: Z-score filtering.

We conducted experiments using `fast_align` data filtering method for English–Polish and English–German. Figure 3 shows the sacreBLEU scores and demonstrates the significant improvement in engine quality using `fast_align` for language pairs with many resources such as

English–German (EN–DE) and low-resource language such as English–Polish (EN–PL) pairs.

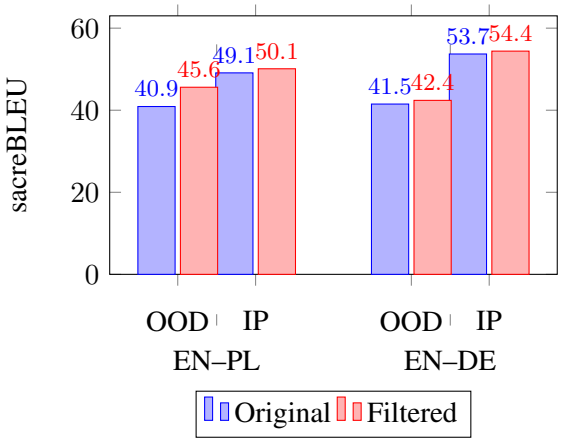

Figure 3: Quality comparison using English–Polish (EN–PL) and English–German (EN–DE) language pairs. 'Original' data means that the data was not cleaned and 'filtered' means that the data was filtered using `fast_align`. The sacreBLEU scores come from the evaluation of the normal test set.

Table 1: Example sizes of parallel corpora for a given language pair before and after data preparation.

Pair	Quality	Domain	Before	After
EN–ES	1	GEN	1.1M	460k
		PH	141k	90k
	2	GEN	649k	257k
		PH	57k	31k
	3	GEN	1M	590k
		PH	226k	144k
	4	GEN	13.5M	6.7M
		PH	1.5M	443k
	5	GEN	0	0
		PH	0	0
	Total	GEN	16.2M	8M
		PH	1.9M	708k
EN–HR	1	GEN	542k	266k
		PH	110k	60k
	2	GEN	268k	121k
		PH	37k	14k
	3	GEN	238k	132k
		PH	12k	8.7k
	4	GEN	1.9M	931k
		PH	0	0
	5	GEN	560k	460k
		PH	42k	33k
	Total	GEN	3.5M	1.9M
		PH	201k	116k

A large amount of data was discarded after the data preparation process. Table 1 shows the number of sentence pairs available for each quality set for a language pair with sufficient resources, such as English–Spanish (EN–ES), compared to a low-resource language pair, such as English–Croatian (EN–HR), before and after data preprocessing.

4 Training

All our engines are built with `OpenNMT-tf` (Klein, 2017), which is an open-source toolkit for NMT and neural sequence learning with a TensorFlow backend.

4.1 Architecture

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

The transformer is based on an encoder-decoder structure (Bahdanau, 2014; Cho, 2014). The encoder maps an input sequence of symbol representations $(x_1, \ldots, x_n)$ to a sequence of continuous representations $(z_1, \ldots, z_n)$. Given z, the decoder then generates an output sequence $(y_1, \ldots, y_m)$ of symbols one element at a time. At each step the model is auto-regressive (Graves, 2014), consuming the previously generated symbols as additional input when generating the next. The transformer follows this overall architecture using stacked self-attention and point-wise, fully connected layers for both the encoder and decoder.

4.2 Hyperparameters

During the training process, we used Adam as optimisation method (Kingma, 2014). 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 (Prechelt, 1998) with a patience value of five evaluations. Once the training was stopped, we averaged the five best stored models.

4.3 Instance description

The training was done in Amazon Elastic Compute Cloud (Amazon EC2).[2] EC2 is a cloud service enabling developers to instantiate machines, which can be configured on-demand in terms of speed, storage and mathematical calculation.

[2] `https://aws.amazon.com/ec2`

We used a 'p3.8xlarge' instance to train translation models. This instance type includes 4 NVIDIA Tesla V100 GPUs with 16 GiB of GPU memory each, which allows an in-domain model to be trained using a TransformerBig architecture in 7-12 hours depending on the language pair.

5 Evaluation

In this section, we describe the validation process and test set generation and which type of sentences each test set contains. We describe the final evaluation of the models by human translators and the benchmarking exercise against two state-of-the-art systems: `eTranslation` and `DeepL Pro`.[3]

5.1 Validation and test

A random test file containing 2,000 sentences was generated from the IND dataset. In addition, several test files were generated to check the quality of specific types of segments, such as very long or very short sentences, sentences with numbers, etc. Those files were produced just once, and were used for all experiments for a given domain and language pair. To generate the test files, we first prepared the training data and, from there, calculated several parameters to use when producing the test sets. These parameters included the threshold indicating when a sentence is considered too long. Each test set can be described as follows:

- Normal: 2,000 pairs; this set is extracted from the IND dataset and is used to check the quality of the trained model.
- Long: 1,000 pairs; it contains long sentences. A sentence is considered long when it is longer than 80% of the sentences in the training set.
- Short: 1,000 pairs; it contains short sentences to translate. The default length of short sentences is two full words but this parameter is configurable.
- Numbers: 1,000 pairs; it contains sentences with numbers, dates and codes, e.g. 578,850 euros, or 40%.
- Uppercase: 1,000 pairs; sentences that contain uppercase letters, e.g. entity names.

To validate the model during the training process, we used a validation set of 2,000 sentence pairs, which was extracted as a normal test set from the IND dataset. It was also generated once and the same validation file was used in all experiments for a given language pair.

5.2 Metrics

We used several standard metrics to evaluate the test files described above. We evaluated each file at document and sentence level based on the following metrics: sacreBLEU (Post, 2018), NIST (Doddington, 2002), TER (Snover, 2006), CHARCUT (Lardilleux, 2017) and METEOR (Denkowski, 2011). Even though sacreBLEU's purpose is to evaluate whole documents, we also used it to evaluate each sentence. The sentence evaluation is only used for our internal records, to collect data for future experiments and to facilitate deeper analysis of the sentences.

5.3 Human evaluation

The final validation of the engines was done by translators. This assessment was carried out with a minimum of one in-house translator (worst-case scenario) and up to three professional linguists, depending on the language pair and the domain. The results for engines in the PH and IP domains are described separately below.

The human assessment focused on the following categories and metrics:

- Fluency: assesses to what extent a translated text is grammatically informed, whether it contains spelling errors, and how it is perceived by a native speaker. It is manually entered by the translator according to a scale of 1 (lowest mark) to 4 (highest mark).
- Adequacy: assesses to what extent the meaning in the source text is expressed in the translation. It is manually entered by the translator according to a scale of 1 (lowest mark) to 4 (highest mark).
- Productivity: computed automatically at a segment level by comparing the raw machine translation against the post-edited version as the normalised edit distance. It considers the minimum number of character edits (i.e.

[3] `https://www.deepl.com`

insertions, deletions or substitutions) that are required to transform the original string into the final version of the same string. Scientific research (Marg, 2016) suggests a strong correlation between edit distance and post-editing productivity metrics.

The acceptance threshold for quality criteria (fluency and adequacy) was set at 2.75 by a consensus among the specialists involved in the project. Also, the acceptance criteria for productivity were as follows: a maximum of 25% of text should be classified as 'Re-translation required', and a minimum of 50% of text should be classified as 'Acceptable as is' or 'Little post-editing needed'. The mapping between these categories and the normalised edit distances was decided by a consensus among the specialists of the project. The results for IP and PH domains are reported separately below.

IP domain: models created for the IP domain cover eight language pairs: {DE, ES, FR, IT}–EN and EN–{DE, ES, FR, IT}. Figure 4 shows the quality of IP documents in eight different languages in terms of fluency and adequacy. The post-editing effort of documents from the IP domain is shown in Figure 5.

The results presented reflect the quality of the first builds that yielded acceptable ratings during the human evaluation, resulting in eight out of eight language pairs deemed fit for purpose according to the fluency and adequacy marks of 1,567 segments. All models were considered as fit for assimilation and post-editing.

PH domain: models created for the PH domain cover seven language pairs: EN–{BG, DA, DE, ES, FR, PL, SV}. Human evaluation was done at a subdomain level, each corresponding to a specialised EU agency (EMA,[4] EU-OSHA,[5] ECDC,[6] EMCDDA[7]), by CdT translators evaluating linguistic quality and productivity aspects. The main difference between the subdomains is that EMA document types can be considered technical, i.e. medical prospects, reports or scientific documentation. Other subdomain texts (EU-OSHA, ECDC, EMCDDA) are of an informative or educational nature, such as web articles, press releases or content for the general public. In total, 1,533 sentences were evaluated by human translators; the number of sentences evaluated from each subdomain is shown in Table 2. Figure 6 shows the quality of EMA documents for seven different language pairs. Figure 7 shows the productivity results for the EMA subdomain.

Table 2: Number of segments evaluated per subdomain.

Subdomain	Segments evaluated
ECDC	161
EMA	854
EMCDDA	168
EU-OSHA	350
Grand Total	1533

The productivity evaluation between different agencies is shown in Figure 8.

Following the acceptance criteria for productivity, six out of seven languages pairs evaluated from the EMA subdomain were considered as fit for assimilation and post-editing. EMCDDA and EU-OSHA results were fit for assimilation and post-editing only for a limited set of language pairs. ECDC subdomain results were not fit for assimilation and post-editing. The language pairs from different subdomains that do not meet the acceptance requirements are shown in Table 3.

Table 3: Subdomains failing at quality and/or productivity.

Subdomain	Pair	Quality	Productivity
EMA	EN–SV	✓	✗
ECDC	EN–FR	✓	✗
	EN–DE	✓	✗
	EN–SV	✓	✗
EU-OSHA	EN–PL	✗	✗
	EN–DE	✓	✗
	EN–DA	✓	✗
	EN–SV	✗	✗
EMCDDA	EN–FR	✓	✗
	EN–DA	✓	✗
	EN–BG	✓	✗
	EN–DE	✗	✗
	EN–ES	✗	✗
	EN–SV	✓	✗

In total, 14 out of 28 cases (language pair plus PH subdomain) did not meet the acceptance requirements in terms of productivity.

In terms of fluency and adequacy, the non-technical documents received much lower scores than technical documents (EMA).

[4] https://www.ema.europa.eu
[5] https://osha.europa.eu
[6] https://www.ecdc.europa.eu
[7] http://www.emcdda.europa.eu

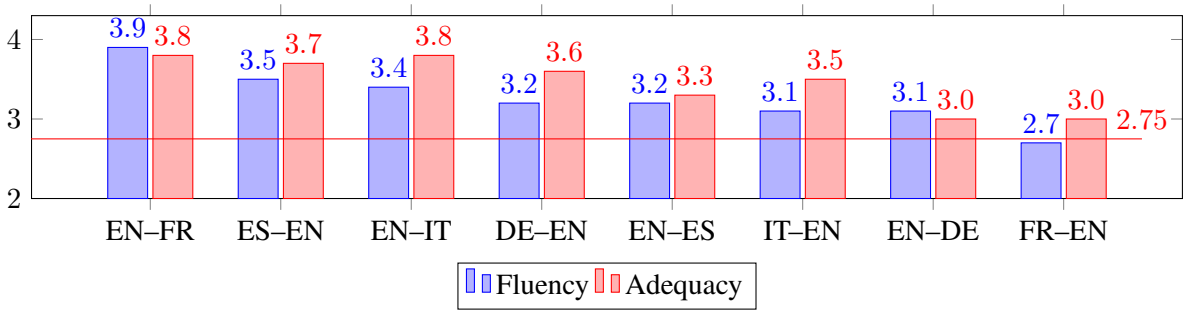

Figure 4: Fluency and adequacy, weighted by segment length in the IP domain.

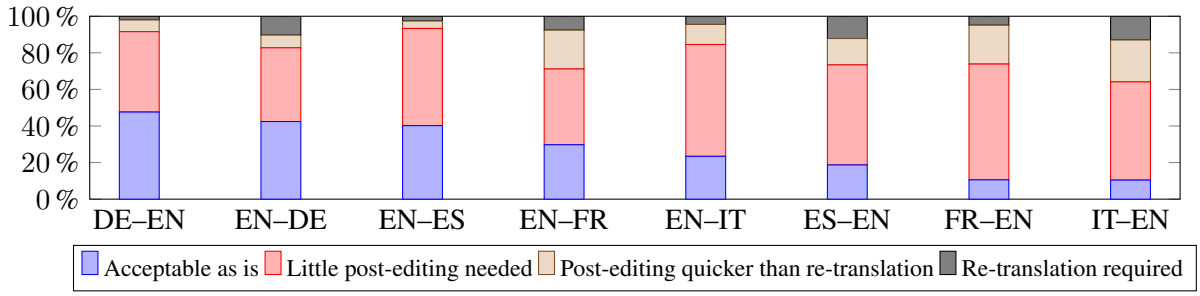

Figure 5: Post-editing effort per segment, weighted by segment length. Based on the calculation of the normalised edit distance at segment level weighted by the segment length in a total of 3,726 segments (85,577 words) post-edited by professional linguists.

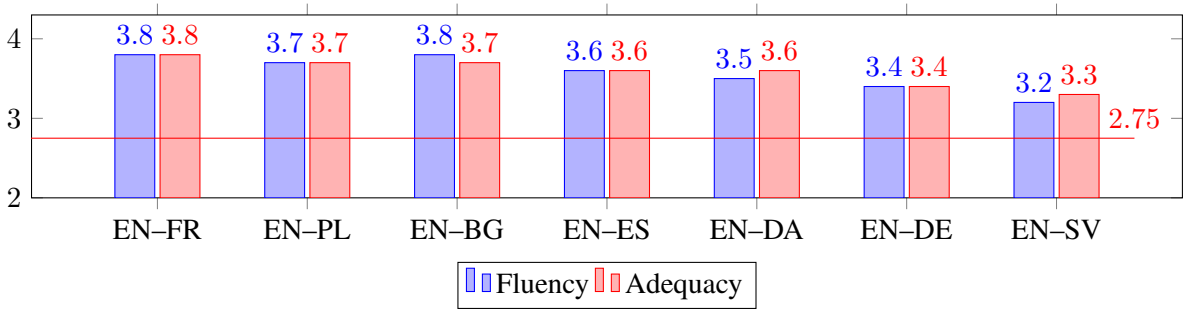

Figure 6: Fluency and adequacy, weighted by segment length from EMA documents.

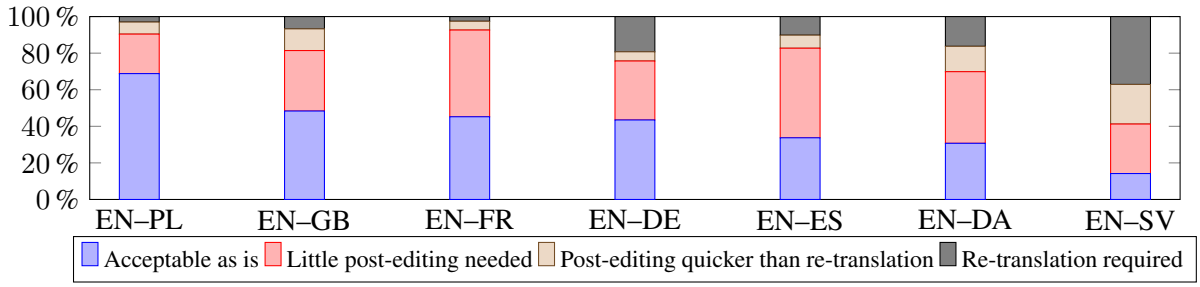

Figure 7: Post-editing effort per segment, weighted by segment length. Based on the calculation of the normalised edit distance at segment level weighted by the segment length in a total of 854 segments from EMA documents post-edited by CdT linguists.

5.4 Benchmarking

The benchmarking was done against `eTranslation` and `DeepL Pro`, which are top-quality machine translation platforms in the industry. However, the system comparison may not be representative enough since the samples used for benchmarking had never been seen before by `NICE`, while this could not be guaranteed in the case of `DeepL Pro` and `eTranslation`. Therefore, results are only indicative and cannot be used to draw any conclusion. The systems for each domain are compared separately below.

IP benchmarking: Figure 9 shows the comparison of quality in terms of fluency and adequacy against `eTranslation` and `DeepL`.

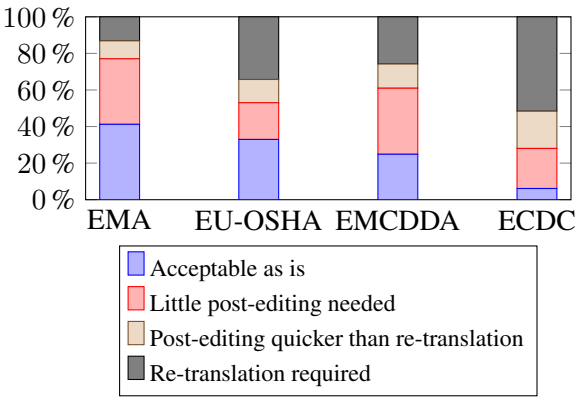

Figure 8: Post-editing effort per segment, weighted by segment length. Based on the calculation of the normalised edit distance at segment level weighted by the segment length in a total of 854 segments from documents of all subdomains from the PH domain post-edited by CdT linguists.

The productivity of benchmarked systems is shown in Figure 10. All three systems show comparable results. Both quality and productivity pass the acceptance threshold.

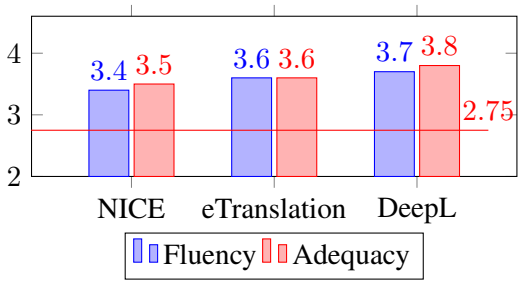

Figure 9: Fluency and adequacy ratings per segment, weighted by segment length for the IP domain.

PH benchmarking: Figure 12 shows the comparison of quality in terms of fluency and adequacy against `eTranslation` and `DeepL Pro`. The productivity of benchmarked systems is shown in Figure 11. The quality of all compared systems is acceptable in terms of quality and fluency. In terms of productivity, only `NICE` meets the requirements. Although `NICE` fails to meet the acceptance requirements for some language pairs from different subdomains, it gets the best score compared to other benchmarked systems.

6 Deployment

For deployment, weight pruning (See, 2016; Zhu, 2017) was applied to accelerate prediction. Weight pruning has several advantages: 1) the inference time is much lower; 2) the model size is reduced. The pruning experiments were done with

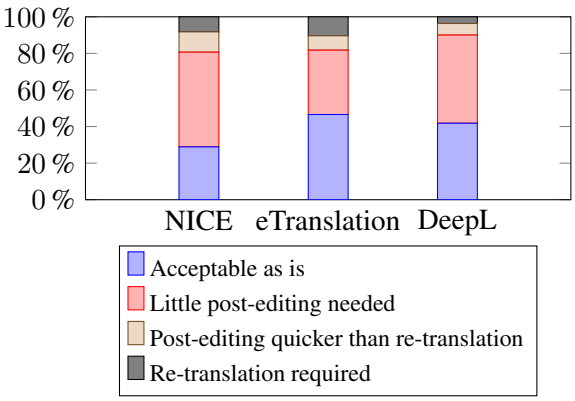

Figure 10: Post-editing effort per segment, weighted by segment length. Based on the calculation of the normalised edit distance at segment level weighted by the segment length in a total of 854 segments from the IP domain post-edited by CdT linguists.

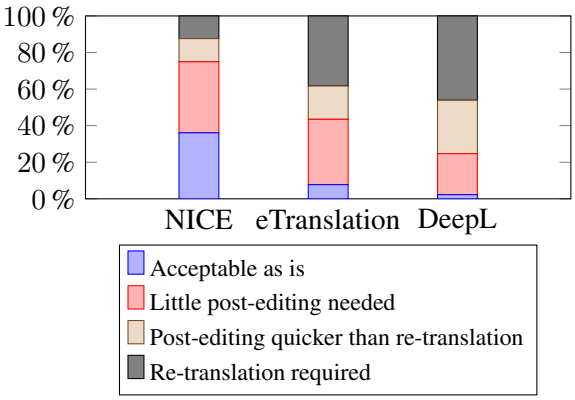

Figure 11: Post-editing effort per segment, weighted by segment length. Based on the calculation of the normalised edit distance at segment level weighted by the segment length in a total of 1 533 segments from documents of all agencies from the PH domain post-edited by CdT linguists.

the `CTranslate2` tool, which is an optimised inference engine for `OpenNMT-py` and `OpenNMT-tf` models supporting both CPU and GPU execution.[8] This library is geared towards an efficient serving of standard translation models, but is also a place for experimentation around model compression and inference acceleration. Table 4 shows the different experiments. The CPU used in the experiment from Figure 4 was i7-7800X CPU 3.5GHz*1.2 and the GPU GeForce GTX 1080 Ti 11GB. Finally, models were pruned using `CTranslate2` and the inference executed on CPU. In our view, the loss of quality is not significant and the inference speed is fast enough for the project purposes.

The final goal of this project is to integrate the custom neural engines into the workflows of the

[8] `https://github.com/OpenNMT/CTranslate2`

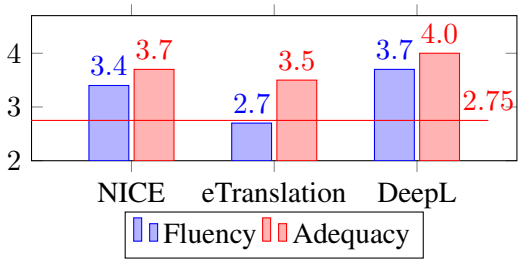

Figure 12: Fluency and adequacy ratings per segment, weighted by segment length for the PH domain.

CdT's advanced translation management system via a web service, allowing for optimised and more efficient translation services.

7 Conclusions and future work

This paper describes the implementation of NICE: Neural Integrated Custom Engines, which was developed by CdT in collaboration with the European Union Intellectual Property Office (EUIPO). The system described in this article includes a sophisticated data preprocessing pipeline. Different techniques for data filtering were applied with satisfactory results. Depending on the language pair, we also applied data augmentation techniques, which improved the output quality.

In this work, we focused on two domains: IP and PH. Nevertheless, the system has been designed to allow the rapid implementation of new EU-related domains, such as legal or finance.

Of the 36 samples evaluated in both domains, 22 were fit for post-editing purposes. NICE produced very satisfactory results for the IP domain and technical documents from the PH domain.

NMT development is an iterative process and it can be assumed that quality will improve over time. The database of high-quality translations produced by CdT contractors and in-house translators is growing day by day. As the CdT collects more data from revised and post-edited translations, incremental learning will be applied (Peris, 2017; Peris, 2019).

Still, there is room for improvement by other means, such as named entity recognition (NER) (Kai, 2019). For example, NER can be applied in the PH domain, where the use of names of medicines and active substances is very frequent, and for which sentencepiece does not manage well, tending to create new words. Another technique under development is the neural quality estimation for translation hypothesis selection (Shah, 2014), which allows the translation quality to be rated without references at run-time.

Finally, another technique is the application of advanced domain adaptation methods to enlarge IND datasets, mainly for languages with fewer resources. We are working to adapt a state-of-the-art classifier from Parcheta et al. (2019) for domain adaptation. The goal is to select more suitable pairs of sentences from the GEN dataset and include them in IND.

Soon, other custom engines will be implemented for other domains, such as the legal domain. We are working on collecting data.

The final step in this project will be to implement a simple web service that seamlessly integrates custom engines into the CdT's advanced translation workflows, allowing translators to work directly with our state-of-the-art, in-domain NMT technology NICE.

Table 4: Comparison of inference time and quality using weight pruning on CPU/GPU.

Hardware	cTranslate2	Model size	sentences/sec	sacreBLEU
CPU	NO	4GB	0.0267	57.9
CPU	YES	1.5GB	3.33	57.1
GPU	NO	4GB	30	57.9
GPU	YES	1.5GB	50	57.1

Acknowledgements

This work has been carried out under a cooperation programme between the CdT and the EUIPO. The authors would like to thank the valuable contributions and support received from Rafael Sáez Mendoza (EUIPO) and the different departments and persons involved in this programme in both organisations.

References

Bahdanau D., Cho K. and Bengio Y. 2015. Neural machine translation by jointly learning to align and translate. *Proc. of 3rd ICLR*

Castilho S., Moorkens J., Gaspari F., Calixto I., Tinsley J. and Way A. 2017. Is neural machine translation the new state of the art? *The Prague Bulletin of Mathematical Linguistics* Vol. 108, Nº 1, 109–120.

Cho K., Van Merrienboer B., Gülçehre Ç., F. Bougares, Schwenk H. and Bengio Y. 2014. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *Proc. of EMNLP* 1724–1734

Denkowski M. and Lavie A. 2011. Meteor 1.3: Automatic metric for reliable optimization and evaluation of machine translation systems. *Proc. of 6th WMT*. Edinburgh, Scotland. 85–91.

Doddington G. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. *Proc. of the second Int. Conf. on Human Language Technology Research* San Diego, California, USA. 138–145.

Graves A. 2013. Generating sequences with recurrent neural networks. *arXiv:1308.0850v5*

Hakala K. and Pyysalo S. 2019. Biomedical Named Entity Recognition with Multilingual BERT. *Proc. of BioNLP*. Hong Kong, China. 56–61.

Jia Y., Carl M., and Wang X. 2017. How does the post-editing of neural machine translation compare with from-scratch translation? A product and process study. The Journal of Specialised Translation Vol. 31, 61–86.

Kingma D. P. and Ba J. 2014. Adam: A Method for Stochastic Optimization. *Proc. of 3rd ICLR*

Klein G., Kim Y.,Deng Y., Senellart J. and Rush A. M. 2017. OpenNMT: Open-Source Toolkit for Neural Machine Translation. *arXiv:1701.02810v2*

Koehn P. 2017. Neural Machine Translation. Available at `http://mt-class.org/jhu/assets/nmt-book.pdf`

Koehn P. and Monz C. 2017. Manual and automatic evaluation of machine translation between european languages. *Proceedings on the Workshop on Statistical Machine Translation* 102–121.

Kudo T. and Richardson J. 2018. SentencePiece: A simple and language independent subword tokenizer. *arXiv:1808.06226v1*

Lardilleux A. and Lepage Y. 2017. CHARCUT: Human-Targeted Character-Based MT Evaluation with Loose Differences. *Proc. of IWSLT*. Tokyo, Japan. 146–153.

Lui M. and Baldwin T. 2012. A Simple, Fast, and Effective Reparameterization of IBM Model 2. *Proc. of the NAACL 2013: Human Language Technologies*. Atlanta, USA. 644–648.

Marg L. 2016. The Trials and Tribulations of Predicting Post-Editing Productivity. *Proc. of LREC*. Portorož, Slovenia. 146–153.

McCallum A. and Nigam K. 1998. A comparison of event models for naive Bayes text classification. *AAAI-98 Workshop on learning for text categorization* 41–48.

Oravecz C., Bontcheva K., Lardilleux A., Tihanyi L. and Eisele A. 2019, eTranslation's Submissions to the WMT 2019 News Translation Task. *proc. WMT*. Florence, Italy. 320–326.

Papineni K., Roukos S., Ward T. and Wei-Jing Z. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. *Proc. of the 40th Annual Meeting on ACL*. Philadelphia, Pennsylvania, USA. 311–318.

Parcheta Z., Sanchis-Trilles G., Casacuberta F. and Redahl R. 2019. Multi-input CNN for Text Classification in Commercial Scenarios. *Proc. of the IWANN*. Gran Canaria, Spain. 596–608.

Peris Á. and Casacuberta F. 2019. Online learning for effort reduction in interactive neural machine translation. 98–126. *Computer Speech & Language*

Peris Á., Cebrián L. and Casacuberta F. 2017. Online learning for neural machine translation post-editing. *arXiv:1706.03196v1*

Post M. 2018. A call for clarity in reporting BLEU scores. 186–191. *Proc. of WMT*.

Post M., Ding S., Martindale M. and Wu W. 2019. An Exploration of Placeholding in Neural Machine Translation, *Proc. of MT Summit XVII. Research Track, Volume 1*, Dublin, Ireland. 182–192.

Prechelt L. 1998. Early stopping – but when? *Neural Networks: Tricks of the trade*. 55–69.

See A. , Luong M.-T. and Manning C. D. 2016. Compression of neural machine translation models via pruning. *CoNLL* 291—301.

Shah K. and Specia L. 2014. Quality Estimation for Translation Selection. *Proc. of EAMT*. Dubrovnik, Croatia. 109–116.

Snover M. , Dorr B., Schwartz R., Micciulla L. and Makhoul J. 2006. A study of translation edit rate with targeted human annotation. *Proc. of AMTA*. Cambridge, Massachusetts, USA. Vol. 200, N^{o} 6, 186–191.

Wu Y., Schuster M., Chen Z., Le Q. V, et al. 2017. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv*

Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A. N., Kaiser L. and Polosukhin I. 2017. Attention Is All You Need. *Advances in neural information processing systems* 5998–6008.

Zhu M. and Gupta S. 2018. To prune, or not to prune: exploring the efficacy of pruning for model compression. *Proc. of ICLR*