SYSTRAN @ WMT 2021: Terminology Task

MinhQuang Pham[†‡], **Antoine Senellart**[†], **Dan Berrebbi**[†], **Josep Crego**[†], **Jean Senellart**[†]
[†] SYSTRAN, 5 rue Feydeau, 75005 Paris, France
firstname.lastname@systrangroup.com
[‡] LISN, Université Paris-Saclay 91405 Orsay, France
firstname.lastname@limsi.fr

Abstract

This paper describes SYSTRAN submissions to the WMT 2021 terminology shared task. We participate in the English-to-French translation direction with a standard Transformer neural machine translation network that we enhance with the ability to dynamically include terminology constraints, a very common industrial practice. Two state-of-the-art terminology insertion methods are evaluated based (i) on the use of placeholders complemented with morphosyntactic annotation and (ii) on the use of target constraints injected in the source stream. Results show the suitability of the presented approaches in the evaluated scenario where terminology is used in a system trained on generic data only.

1 Introduction

The high quality obtained by out-of-the-box neural machine translation (NMT) systems (Bojar et al., 2016) has boosted the adoption of automatic translation by the industry and invigorated the research and development on domain adaption and integration of technology in human translation workflows. For instance, combination with translation memories (Bulte and Tezcan, 2019; Xu et al., 2020; Pham et al., 2020), terminology handling (Hasler et al., 2018; Dinu et al., 2019; Michon et al., 2020; Bergmanis and Pinnis, 2021), interactive translation (Peris and Casacuberta, 2019), post-editing modelling (Chatterjee et al., 2020) or dynamic adaptation (Farajian et al., 2017) are all different techniques to make machine translation part of real-life localization workflow.

Terminology resources with all their sophistication have been the core building bricks and a continuous challenge to acquire in volume (Senellart et al., 2003) for rule-based engines. At the other extreme, they have been reduced to corpus or aligned "phrase pairs" (Schwenk et al., 2008) for Statistical Machine Translation approaches, missing most of their intrinsic linguistic properties. In contrast, neural machine translation operates on word and sentence representations in a continuous space so, on the one hand, it has access to deep actual linguistic knowledge (Conneau et al., 2018) and demonstrates a huge ability to generalize. But on the other hand, results are more difficult to interpret (Koehn and Knowles, 2017), and subsequently the translation process is far more complicated to control. Therefore, as for several other linguistic annotations, the challenge is how terminological information can be "passed" to the model. From a human perspective, even though presentation and usage of dictionaries have evolved from ontology (as found in paper dictionary) to corpus-based presentation, looking up terms in a dictionary is the ultimate point of reference for validating the correct term for a specific domain and context.

Inline with the conditions of the WMT 2021 terminology shared task, we present English-to-French NMT engines built from abundant generic (out-of-domain) training data. We evaluate several methods to enhance translation engines with the ability to integrate terminology as a quick way to dynamically specialize a translation to a particular domain, which in this case considers the new COVID-19 domain and the large efforts for translation of critical information regarding pandemic handling and infection prevention strategies. In-domain resources are limited to word- and phrase-level terminology entries created to guide professional translators to ensure both accuracy and consistency in translations. Our generic systems make only use of terminologies at inference time.

The remainder of the paper is organized as follows: Section 2 gives details of several terminology injection approaches considered in this work. In addition, we outline a grammatical error correction network that is applied over French translation hypotheses. The experimental framework is presented in Section 3. Results are discussed in Section 4. Finally, we draw conclusions in Section 5.

2 Terminology Injection

Terminology is typically defined as the technical or special terms used in a business, art, science, or special subject. A high quality asset maintained by language specialists as part of a translation project. It allows to guarantee language consistency, certify translation accuracy and define constraints to human translation.

In recent years there has been significant work proposing methods to integrate such external specialized terminologies into NMT models, each showing different levels of performance when facing terminology injection issues, mainly inference overhead and generalization power.

In this section we describe the two main methods employed for this shared task and illustrate the particularities of each on a common scenario using two English-French terminology entries: *Coronavirus* ⇝ *Coronavirus* and *pneumonia* ⇝ *pneumonie*, in the following translation:

Coronavirus can cause pneumonia
Les coronavirus peuvent causer des pneumonies

Table 1 illustrates training examples of each terminology injection methods evaluated in this work. First row shows the ***base*** configuration where no terminology is employed.

2.1 Placeholders

Our first method incorporates non-terminal tokens into NMT systems, which require modifying the pre-and post-processing of the data, and training the system with data that contains the same placeholders which occur in the test sets (Crego et al., 2016; Michon et al., 2020). Following our example, source and translation terms appearing in the sentence pair are replaced by placeholders adapted to cover a wider variety of cases, and to control morphology to allow generalization power.

The method presented in Michon et al. (2020) allows handling very challenging cases concerning homographs. This is, words (or phrases) that sharing the same form (*i.e. spread*) can occur with multiple meanings or different grammatical functions (*verb* or *noun*). The method predicts the part-of-speech of the target placeholder. Thus, solving the source homograph.

Given that the vast majority of the terminology released for this shared task consists of nouns (single words or phrases) we decided to use a simplified version of the method that only considers using noun placeholders.

Row ***mrk*** in Table 1 illustrates the use of placeholders for our previous training example. Each word form detected as terminology is replaced by two placeholders: The first indicates the part-of-speech of the terminology (in this work always a noun 'N') followed by a unique identifier; the second indicates the set of features conveying the morphology of the noun (masc/fem and sing/plur).

To predict the target morphology of the each term, the NMT model may find it useful to have access to the source word form. Thus, in a second version of the method we incorporate the terminology word form (*Coronaviruses* and *pneumonia*) in the source stream. We denote this version ***mrk+***. It is worth to notice that this second version only improves on the previous when the incorporated source term has sufficiently occurred in training.

Note also that Michon et al. (2020) do not require linguistic information in inference since ambiguities are not resolved in the source placeholders. In contrast, our implementation uses `SpaCy`[1] to obtain part-of-speeches and morphology features of input streams.

Target-side streams of methods *mrk* and *mrk+*, require post-processing to replace target-side placeholders by the final word forms. In practice, for each source-target term pair we encode all possible inflections of the source and target word labelled with the corresponding inflection type (placeholders). Not only does this analysis enable to lexically match any inflected form of the source term, but it can also produce any inflected form of the translation term, ensuring full flexibility in the inflection choice made by the neural network. Table 2 illustrates target word forms for the terminology *pneumonia* ⇝ *pneumonie*.

2.2 Learning to apply constraints

This approach tackles the same problem by learning a copy behaviour of terminology at training time (Song et al., 2019; Dinu et al., 2019; Bergmanis and Pinnis, 2021). The NMT model is trained to incorporate terminology translations when they are provided as additional input in the source sentence. Terminology translations are inserted as inline annotations, expecting the model to learn that such additional words must be copied in the target hypothesis. The authors insert terminology translations in the source sentence either by *appending* the target term to its source version, or by

[1] `https://spacy.io/`

Placeholders (tgt)	Word form
<N> <fem_sing>	pneumonie
<N> <fem_plur>	pneumonies

Table 2: Target word forms and associated placeholders for the term entry *pneumonia* ⇝ *pneumonie*.

directly *replacing* the original term with the target one. Example in row ***app*** of Table 1 illustrates the *append* alternative presented in Dinu et al. (2019).

The approach uses a generic NMT architecture which learns to use an external terminology provided at run-time, thus, showing no inference overhead. However, similarly to the preceding approach, it lacks generalization power as it simply "copies" the term found in the terminology base injected in the source sentence, irrespective of the target hypothesis context. Dinu et al. (2019) argue that in some cases the approach exhibits the ability to inflect translation terms.

Finally, a second version of the method is also illustrated in Table 1, denoted as ***app+***. The target term is injected using its lemma form. Thus, forcing the NMT model to produce the right inflection of the term observed in the source stream. In the example, *pneumonie* must be inflected in its plural form, *pneumonies*. Tokens <b>, <i> and <e> are used to inform the model of the source and target terminology boundaries. Note that, in contrast to placeholder methods, no additional post-processing is required.

2.3 Grammatical Error Correction

As previously stated, placeholder methods allow larger generalization power thanks to the flexibility of the inflection mechanism employed in the translation workflow. However, morphology choices made by the network do not take into account the actual word forms, which was observed to result in a higher number of inflection errors (Michon et al., 2020). To alleviate this problem we add a correction module that performs over the resulting translation hypotheses.

We use a correction module based on Gector (Omelianchuk et al., 2020) with a pretrained multilingual BERT to correct grammatically incorrect French words. The model predicts grammatical features for each word in the translated sentence, allowing only for 3 types of edits:

- Transformation of gender/number

  le [Fem] → la
  le [Plur] → les

- transformation of tense/person of verbs

  avez [3_Plur] → avons
  avez [Ind_Imp] → aviez

- Elision

  le [ELISION] → l'

Table 9 in Appendix B illustrates the vocabulary of tags considered by the model. Once the model predicts whether a word needs to be corrected (and which correction), the final word form is found using a dictionary and the predicted tag. Table 3 illustrates examples of translation hypotheses produced by the NMT model (Hyp) predicted tags for each word (Pred) and corrected hypotheses (Corr). Tag ✓ is used to indicate that no transformation is required.

base	Coronaviruses can cause pneumonia Les coronavirus peuvent causer des pneumonies
mrk	*<N#1> <masc_plur> can cause <N#2> <fem_sing>* *Les <N#1> <masc_plur> peuvent causer des <N#2> <fem_plur>*
mrk+	*<N#1> Coronaviruses <masc_plur> can cause <N#2> pneumonia <fem_sing>* *Les <N#1> <masc_plur> peuvent causer des <N#2> <fem_plur>*
app	*<b> Coronaviruses <i> coronavirus <e> can cause <b> pneumonia <i> pneumonies <e>* *Les coronavirus peuvent causer des pneumonies*
app+	*<b> Coronaviruses <i> coronavirus <e> can cause <b> pneumonia <i> pneumonie <e>* *Les coronavirus peuvent causer des pneumonies*

Table 1: Examples of training streams for the same sentence pair using terms *Coronaviruses* ⇝ *Coronavirus* and *pneumonia* ⇝ *pneumonie* according to each injection method evaluated in this work.

Hyp	...	**le**	épidémie	rapidement	propagée	aux	villes	...
Pred	...	**ELISION**	✓	✓	✓	✓	✓	...
Corr	...	**l'**	épidémie	rapidement	propagée	aux	villes	...
Hyp	...	avec	**le**	fièvre	à	peu	près	...
Pred	...	✓	**Fem_Sing**	✓	✓	✓	✓	...
Corr	...	avec	**la**	fièvre	à	peu	près	...
Hyp	...	atteintes	à	la	mise	en	**quarantaines**	...
Pred	...	✓	✓	✓	✓	✓	**Fem_Sing**	...
Corr	...	atteintes	à	la	mise	en	**quarantaine**	...
Hyp	...	cas	de	COVID-19	**confirmées**	en	laboratoire	...
Pred	...	✓	✓	✓	**Masc_Plur_Past_Part**	✓	✓	...
Corr	...	cas	de	COVID-19	**confirmés**	en	laboratoire	...

Table 3: Examples of word edits performed by the correction model.

3 Experimental Framework

3.1 Corpora

Table 7 in Appendix A provides some statistics on the parallel corpora employed for training our models. It is important to note that all corpora used are out-of-domain. We first filtered out longer sentences and sentences with a significant difference in the number of words between the source and the corresponding translation. All data is pre-processed using the OpenNMT tokenizer[2].

In order to train our correction (GeC) model with additional data, we also use the monolingual (French) corpora made available for the shared task. See Table 8 in Appendix A for detailed statistics of monolingual data.

3.2 Terminology

Table 4 illustrates some examples of the terminology entries released by the organisers of the shared task.

English	French
contagious	contagieux
active cases	cas actifs
confirmed cases	cas confirmés

Table 4: English-French terminology examples.

We note that most terminology entries are composed of several words. Indeed 54.8% of terms are groups of two words, 22.3% contains more than three words and only 22.9% are single words as measured in the source side.

3.3 NMT Engines

All our NMT engines follow the Transformer architecture (Vaswani et al., 2017) implemented by the `OpenNMT-tf`[3] toolkit (Klein et al., 2017). More precisely, we use: Word embedding size: 1024; Number of layers: 6; Number of heads in multi-head self-attention layer: 16; Inner dimension of feedforward layer: 4096; Dropout rate: 0.1; In addition, we use shared embeddings for both the input and output layers. The encoder and decoder use the same BPE units (Sennrich et al., 2016) learned from source and target corpora. We train our MT models using Noam schedule (Vaswani et al., 2017) with 4000 warm-up iterations. To balance between the domains of the training corpora, we use the following sampling distribution over the training corpora:

$$\lambda_\alpha(d) = \frac{q_d^\alpha}{\sum_{d=1}^{n_d} q_d^\alpha}, \quad (1)$$

where q_d is the size of d^{th} corpora, the scalar $\alpha \in [0, +\infty]$ changes the sampling distribution as low α upsamples small corpora and downsamples large corpora while high α favors large corpora over small corpora. In the training of our MT systems, we use $\alpha = 0.5$. Learning is performed over 8 GPUs during $300K$ steps with a batch size of $32K$ tokens per step. During training, we filtered out sentences larger than 250 tokens. We applied label smoothing to the cross-entropy loss with a rate of 0.1. Resulting models are built after averaging the last ten checkpoints of the training process. In inference, we apply a length penalty rate of 0.6.

[2] https://github.com/OpenNMT/Tokenizer

[3] https://github.com/OpenNMT/OpenNMT-tf

3.4 Training NMT

Terminology injection approaches implemented for this evaluation rely on NMT models with the ability to translate input streams with target terms (*app* and *app+*) and using placeholders (*mrk* and *mrk+*). Thus, a key step for our models is the availability of training data with such annotations.

To identify terminology pairs in our training database we :

- Analyse English and French using `SpaCy` to produce part-of-speeches, morphology features, noun phrases and lemmas. Only NPs (single words or phrases) are considered.
- Word align English and French parallel corpora using the `fast_align`[4] toolkit (Dyer et al., 2013).

Terminology pairs are only considered when English and French sides consist of noun phrases and when words in a term are only aligned to words in its counterpart.

Words of the terminology entries identified are replaced by the corresponding tokens (depending on the approach). See Table 1 for examples of sentence pairs with terminology entries. We make sure that a given sentence pair does not exceed 5 terminology entries.

3.5 Training GeC

We used all available French corpora to train our GeC network. To include errors in the French streams we replace some words by any inflection of its base form (lemma). The resulting corpora is then tokenized using wordpiece and passed to the BERT language model for embedding. Error detection and tagging are then performed by the network from subword embeddings. Grammatical features, part-of-speeches and lemmas are performed by the `SpaCy` toolkit. Table 3 illustrates examples of word error correction by our model.

3.6 Terms with Multiple Translations

Note that terms released by the shared task organisers may have multiple translation options (*i.e. quarantine ⇝ quarantaine/mise en quarantaine*). Thus, the right translation must be predicted and injected in the translation hypothesis.

The translation workflow implemented for this evaluation considers the injection of each translation option into the input sentence. This is, when a matched term is built with n different translation options, the original input sentence is copied n times and each translation is injected into one copy. Once all copies have been translated, the one showing the lowest perplexity is selected as measured by the pretrained BERT French language model detailed in section 2.3.

4 Results

Table 5 indicates BLEU[5] (Post, 2018) accuracy results of our NMT systems implementing different terminology injection methods before (second column) and after (third column) grammatical error correction.

System	NMT	`+corr`
`base`	**44.9**	44.8
`mrk`	42.3	**42.7**
`mrk+`	44.9	**45.1**
`app`	45.9	**46.0**
`app+`	**45.9**	**45.9**

Table 5: BLEU score of our NMT systems before (NMT) and after the correction model (`+corr`) measured over the development set.

As it can be seen, the methods that learn to apply constraints (*app* and *app+*) obtain the best performance. Overall, the GeC model succeeds in fixing grammatically incorrect French words. However, a benefit barely reflected by BLEU.

We now evaluate the performance of matching terminology entries over the development input sentences. Note that the same matching method is always applied, detailed in Section 2.1, where input sentences are matched against all possible inflections of source terms. Table 6 illustrates the accuracy of recognized terms. 73 percent of the unrecognized terms are verbs which we choose to not process. We recognized also 234 terms that are not highlighted in the development set (FP), most of them do not interfere with translations.

Accuracy	FN	FP
0.97	0.03	0.21

Table 6: Matching rates of terminology entries measured over the development set. FN and FP scores stand respectively for false negatives (terms not identified) and false positives (wrong terminology identifications).

[4] https://github.com/clab/fast_align

[5] https://github.com/mjpost/sacrebleu

5 Conclusions

We presented SYSTRAN English-to-French submissions to WMT 2021 terminology shared task. All our systems follow the Transformer network architecture enhanced with the ability to dynamically include terminology constraints. Several terminology injection methods were evaluated, showing their ability to effectively injecting terms while producing highly accurate translations.

Acknowledgements

The work presented in this paper was partially supported by the European Commission under contract H2020-787061 ANITA.

This work was granted access to the HPC resources of [TGCC/CINES/IDRIS] under the allocation 2020- [AD011011270] made by GENCI (Grand Equipement National de Calcul Intensif).

References

Toms Bergmanis and Mārcis Pinnis. 2021. Facilitating terminology translation with target lemma annotations. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3105–3111, Online. Association for Computational Linguistics.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. 2016. Findings of the 2016 conference on machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 131–198, Berlin, Germany. Association for Computational Linguistics.

Bram Bulte and Arda Tezcan. 2019. Neural fuzzy repair: Integrating fuzzy matches into neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1800–1809, Florence, Italy. Association for Computational Linguistics.

Rajen Chatterjee, Markus Freitag, Matteo Negri, and Marco Turchi. 2020. Findings of the WMT 2020 shared task on automatic post-editing. In *Proceedings of the Fifth Conference on Machine Translation*, pages 646–659, Online. Association for Computational Linguistics.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136, Melbourne, Australia. Association for Computational Linguistics.

Josep Crego, Jungi Kim, Guillaume Klein, Anabel Rebollo, Kathy Yang, Jean Senellart, Egor Akhanov, Patrice Brunelle, Aurelien Coquard, Yongchao Deng, Satoshi Enoue, Chiyo Geiss, Joshua Johanson, Ardas Khalsa, Raoum Khiari, Byeongil Ko, Catherine Kobus, Jean Lorieux, Leidiana Martins, Dang-Chuan Nguyen, Alexandra Priori, Thomas Riccardi, Natalia Segal, Christophe Servan, Cyril Tiquet, Bo Wang, Jin Yang, Dakun Zhang, Jing Zhou, and Peter Zoldan. 2016. Systran's pure neural machine translation systems. *CoRR*, abs/1610.05540.

Georgiana Dinu, Prashant Mathur, Marcello Federico, and Yaser Al-Onaizan. 2019. Training neural machine translation to apply terminology constraints. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3063–3068, Florence, Italy. Association for Computational Linguistics.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. 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*, pages 644–648, Atlanta, Georgia. Association for Computational Linguistics.

M. Amin Farajian, Marco Turchi, Matteo Negri, and Marcello Federico. 2017. Multi-domain neural machine translation through unsupervised adaptation. In *Proceedings of the Second Conference on Machine Translation*, pages 127–137, Copenhagen, Denmark. Association for Computational Linguistics.

Eva Hasler, Adrià de Gispert, Gonzalo Iglesias, and Bill Byrne. 2018. Neural machine translation decoding with terminology constraints. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 506–512, New Orleans, Louisiana. Association for Computational Linguistics.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada. Association for Computational Linguistics.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver. Association for Computational Linguistics.

Elise Michon, Josep Crego, and Jean Senellart. 2020. Integrating domain terminology into neural machine translation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3925–3937, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem Chernodub, and Oleksandr Skurzhanskyi. 2020. GECToR – grammatical error correction: Tag, not rewrite. In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 163–170, Seattle, WA, USA â†' Online. Association for Computational Linguistics.

Álvaro Peris and Francisco Casacuberta. 2019. A neural, interactive-predictive system for multimodal sequence to sequence tasks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 81–86, Florence, Italy. Association for Computational Linguistics.

Minh Quang Pham, Jitao Xu, Josep Crego, François Yvon, and Jean Senellart. 2020. Priming neural machine translation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 516–527, Online. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Holger Schwenk, Jean-Baptiste Fouet, and Jean Senellart. 2008. First steps towards a general purpose French/English statistical machine translation system. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 119–122, Columbus, Ohio. Association for Computational Linguistics.

Jean Senellart, Christian Boitet, and Laurent Romary. 2003. SYSTRAN new generation: the XML translation workflow. In *Proceedings of Machine Translation Summit IX: Papers*, New Orleans, USA.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany.

Kai Song, Yue Zhang, Heng Yu, Weihua Luo, Kun Wang, and Min Zhang. 2019. Code-switching for enhancing NMT with pre-specified translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 449–459, Minneapolis, Minnesota. Association for Computational Linguistics.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, Istanbul, Turkey. European Language Resources Association (ELRA).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Jitao Xu, Josep Crego, and Jean Senellart. 2020. Boosting neural machine translation with similar translations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1580–1590, Online. Association for Computational Linguistics.

A Corpora Statistics

We experiment with English-French corpora made available via the shared task organisers[6] (Tiedemann, 2012), corresponding to texts from: News commentaries (*news*), European Parliament proceedings (*epps*), United Nations official records and documents (*unpc*), web crawling (*ccrawl*, *pcrawl* and *giga*), In addition we also used the next monolingual French data sets: News Commentary 2019 (*news.19*) and News Commentary 2020 (*news.20*).

Table 7 shows statistics of the parallel corpora used for learning NMT models. Statistics computed after a lightly tokenization (to split-off punctuation). Data sets were previously filtered to discard very long sentences (> 80 words) and with very different number of tokens on either side (fertility > 6 words).

Corpus	Sents (M)	Words (M)		Vocab (K)	
		En	Fr	En	Fr
news	0.2	4.4	5.5	70	75
epps	1.5	41.3	47.7	111	127
ccrawl	1.4	28.6	33.2	486	496
giga	9.4	212.2	259.9	1467	1376
unpc	11.8	256.7	330.3	739	622
pcrawl	92.2	1898.6	2237.2	8110	7757

Table 7: Statistics of parallel corpora used for training NMT. Number of sentences and words are given in millions, and vocabularies in thousands.

[6]Freely available from http://opus.nlpl.eu

Table 8 shows statistics of the monolingual (French) corpora used for learning our GeC model. Statistics computed after a lightly tokenization (to split-off punctuation).

Corpus	Sents (M)	Words (M)	Vocab (K)
news.19	10.2	247.9	955
news.20	9.3	232.5	912

Table 8: Statistics of monolingual corpora used for training GeC. Number of sentences and words are given in millions, and vocabularies in thousands.

B Vocabulary of GeC

Table 9 illustrates the vocabulary of tags considered by our GeC model.

Vocabulary	Example
✓	Ø
<Gender=Masc_Number=Sing>	chiennes → chien
<Gender=Fem_Number=Sing>	chiens → chienne
ELISION	le → l’
<VerbForm=Inf>	avons → avoir
<Mood=Ind_Number=Sing_Person=3_Tense=Pres_VerbForm=Fin>	avoir → a
<Gender=Masc_Number=Plur>	chienne → chiens
<Gender=Masc_Number=Sing_Tense=Past_VerbForm=Part>	avoir → eu
<Number=Plur>	homme → hommes
<Gender=Fem_Number=Plur>	chien → chiennes
<Number=Sing>	hommes → homme
<Mood=Ind_Number=Plur_Person=3_Tense=Pres_VerbForm=Fin>	avoir → ont
<Gender=Masc_Number=Sing_Tense=Past_VerbForm=Part_Voice=Pass>	avoir → eu
<Tense=Pres_VerbForm=Part>	avoir → ayant
<Mood=Ind_Number=Sing_Person=3_Tense=Imp_VerbForm=Fin>	avoir → avait
<Gender=Masc>	chienne → chien
<Gender=Fem_Number=Sing_Tense=Past_VerbForm=Part>	avoir → eue
<Mood=Ind_Number=Sing_Person=3_Tense=Fut_VerbForm=Fin>	avoir → aura
<Gender=Fem_Number=Sing_Tense=Past_VerbForm=Part_Voice=Pass>	avoir → eue
<Gender=Masc_Number=Plur_Tense=Past_VerbForm=Part>	avoir → eus
<Mood=Ind_Number=Sing_Person=1_Tense=Pres_VerbForm=Fin>	avoir → ai
<Gender=Masc_Number=Plur_Tense=Past_VerbForm=Part_Voice=Pass>	avoir → eus
<Gender=Masc_Tense=Past_VerbForm=Part>	avoir → eu
<Mood=Ind_Number=Plur_Person=1_Tense=Pres_VerbForm=Fin>	avoir → avons
<Gender=Fem_Number=Plur_Tense=Past_VerbForm=Part_Voice=Pass>	avoir → eues
<Mood=Cnd_Number=Sing_Person=3_Tense=Pres_VerbForm=Fin>	avoir → aurais
<Gender=Fem_Number=Plur_Tense=Past_VerbForm=Part>	avoir → eues
<Mood=Ind_Number=Plur_Person=3_Tense=Fut_VerbForm=Fin>	avoir → auront
<Mood=Ind_Number=Plur_Person=3_Tense=Imp_VerbForm=Fin>	avoir → avaient
<Gender=Masc_NumType=Ord_Number=Sing>	cents → cent
<Mood=Ind_Number=Sing_Person=3_Tense=Past_VerbForm=Fin>	avoir → eut
<Gender=Fem_NumType=Ord_Number=Sing>	cents → cent
<Tense=Past_VerbForm=Part>	avoir → eu
<Mood=Ind_Number=Plur_Person=2_Tense=Pres_VerbForm=Fin>	avoir → avez
<Mood=Sub_Number=Sing_Person=3_Tense=Pres_VerbForm=Fin>	avoir → ait
<NumType=Ord_Number=Sing>	cents → cent
<Gender=Masc_Tense=Past_VerbForm=Part_Voice=Pass>	avoir → eu
<Gender=Fem>	chien → chienne
<Mood=Cnd_Number=Plur_Person=3_Tense=Pres_VerbForm=Fin>	avoir → auraient
<Gender=Masc_NumType=Card_Number=Plur>	quatrième → quatrièmes
<Gender=Masc_NumType=Ord_Number=Plur>	cent → cents
<Mood=Imp_Number=Plur_Person=2_Tense=Pres_VerbForm=Fin>	avoir → ayez
<Mood=Ind_Number=Sing_Person=1_Tense=Imp_VerbForm=Fin>	avoir → avais
<Gender=Fem_NumType=Ord_Number=Plur>	cent → cents
<Mood=Sub_Number=Plur_Person=3_Tense=Pres_VerbForm=Fin>	avoir → aient
<Mood=Ind_Number=Plur_Person=1_Tense=Fut_VerbForm=Fin>	avoir → aurons
<Mood=Ind_Number=Plur_Person=1_Tense=Imp_VerbForm=Fin>	avoir → avions
<Gender=Masc_NumType=Card_Number=Sing>	premières → premier
<Mood=Cnd_Number=Sing_Person=1_Tense=Pres_VerbForm=Fin>	avoir → aurais
<Mood=Ind_Number=Plur_Person=3_Tense=Past_VerbForm=Fin>	avoir → eurent
<Mood=Cnd_Number=Plur_Person=1_Tense=Pres_VerbForm=Fin>	avoir → aurais
<Mood=Ind_Number=Plur_Person=2_Tense=Fut_VerbForm=Fin>	avoir → aurez
<Mood=Ind_Number=Sing_Person=1_Tense=Fut_VerbForm=Fin>	avoir → aurai
<Number=Plur_Tense=Past_VerbForm=Part_Voice=Pass>	avoir→ eus
<Number=Sing_Tense=Past_VerbForm=Part>	avoir → eu
<Mood=Sub_Number=Sing_Person=1_Tense=Pres_VerbForm=Fin>	avoir → aie
<Mood=Ind_Person=3_Tense=Pres_VerbForm=Fin>	neiger → neige
<Tense=Past_VerbForm=Part_Voice=Pass>	avoir → eu
<Mood=Sub_Number=Sing_Person=3_Tense=Past_VerbForm=Fin>	avoir → eu
<Mood=Cnd_Number=Plur_Person=2_Tense=Pres_VerbForm=Fin>	avoir → auriez
<Number=Sing_Tense=Past_VerbForm=Part_Voice=Pass>	avoir → eu
<Mood=Imp_Number=Plur_Person=1_Tense=Pres_VerbForm=Fin>	avoir → ayons
<Number=Plur_Tense=Past_VerbForm=Part>	avoir → eus
<Mood=Ind_Number=Plur_Person=2_Tense=Imp_VerbForm=Fin>	avoir → aviez
<Mood=Imp_Tense=Pres_VerbForm=Fin>	avoir → aie
<Mood=Sub_Number=Plur_Person=1_Tense=Pres_VerbForm=Fin>	avoir → avons
<Mood=Ind_Number=Sing_Person=2_Tense=Imp_VerbForm=Fin>	avoir → avais
<Mood=Sub_Number=Plur_Person=2_Tense=Pres_VerbForm=Fin>	avoir → avez

Table 9: Vocabulary of tags of our GeC model.