Integrating Domain Terminology into Neural Machine Translation

Elise Michon, Josep Crego, Jean Senellart

SYSTRAN, 5 rue Feydeau, 75002 Paris, France
firstname.lastname@systrangroup.com

Abstract

This paper extends existing work on terminology integration into Neural Machine Translation, a common industrial practice to dynamically adapt translation to a specific domain. Our method, based on the use of placeholders complemented with morphosyntactic annotation, efficiently taps into the ability of the neural network to deal with symbolic knowledge to surpass the surface generalization shown by alternative techniques. We compare our approach to state-of-the-art systems and benchmark them through a well-defined evaluation framework, focusing on actual application of terminology and not just on the overall performance. Results indicate the suitability of our method in the use-case where terminology is used in a system trained on generic data only.

1 Introduction

High out-of-the-box quality for Neural Machine Translation (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 (Bulté and Tezcan, 2019; Xu et al., 2020), terminology handling (Hasler et al., 2018; Dinu et al., 2019), interactive translation (Peris and Casacuberta, 2019), post-editing modelling (Chatterjee et al., 2019) or dynamic adaptation (Farajian et al., 2017) are all different techniques to make machine translation part of real-life localization workflow.

In this work, we focus on integrating terminology as a quick way to dynamically specialize a translation to a specific domain. Terminology is a key high quality asset maintained by language specialists as part of a translation project: it is a way to guarantee language consistency, certify translation accuracy and define constraints to human translation. Terminologists are putting a lot of effort to describe terms, including their morphology, their syntax, the semantic context in which these terms apply, *etc.* 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 in a specific context.

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 "phrases" (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 one hand, 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.

In this work, we extend existing work on terminology adaptation, show similarity with translation memory, and propose a new approach and new benchmark through a well-defined evaluation framework focusing on actual application of terminology and not just on the overall performance.

Proceedings of the 28th International Conference on Computational Linguistics, pages 3925–3937
Barcelona, Spain (Online), December 8-13, 2020

2 Related Work

In recent years there has been significant work proposing methods to integrate such external specialized terminologies into NMT models. Mainstream techniques to tackle this challenge can be divided into three broad approaches, each showing different levels of performance when facing terminology injection issues, mainly inference overhead and generalization power. We illustrate their particularities on a common scenario using two English-Spanish terminology entries: *[precedent ⇝ precedente] (noun)* and *[to extend ⇝ ampliar] (verb)*, in the following translation:

These precedents can be extended.
Se pueden ampliar esos precedentes.

Placeholders incorporate non-terminal tokens into NMT systems, which requires 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). Following our example, source and translation terms appearing in the sentence pair are replaced by placeholders[1]:

These <term#1> can be <term#2> .
Se pueden <term#2> esos <term#1> .

The previous sentence pair is then used to feed the translation network, which learns to produce the target sentence with the corresponding placeholders. A similar workflow is applied in inference. Firstly, pre-processing work replaces found source terms and their morphological variants by placeholders (*precedents ⇝ <term#1>* and *extended ⇝ <term#2>*). Secondly, post-processing work is applied over the NMT output, where in turn translation terms replace placeholders (*<term#1> ⇝ precedentes* and *<term#2> ⇝ ampliar*)[2]. Note that the network loses any possibility to model the tokens in the terminology, since it has only access to placeholders. The method also lacks flexibility, as the model will always replace the placeholder with the same phrase irrespective of grammatical context[3]. In contrast, no computational overhead is applied at inference time by pre- and post-processing. The approach inherits from Luong et al. (2015) where words translated as out-of-vocabulary by the NMT network are post-processed using a dictionary.

Learning to apply constraints tackles the same problem by learning a copy behaviour of terminology at training time (Song et al., 2019; Dinu et al., 2019). 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 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 directly *replacing* the original term with the target one. Both alternatives obtain similar translation accuracy results. An additional input stream is also used to signal the switch between source text and target terminology to be copied. Additional factors contain three values: 0 for source words, 1 for source terms, and 2 for target terms[4]:

These$_0$ *precedents*$_1$ *precedentes*$_2$ *can*$_0$ *be*$_0$ *extended*$_1$ *ampliar*$_2$.$_0$
Se pueden ampliar esos precedentes .

This 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 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.

Recently, Bulté and Tezcan (2019) and Xu et al. (2020) followed a similar methodology where source sentences are augmented with entire translation sentences retrieved from translation memories, using

[1] Source and translation terms are usually required to be aligned to each other in the sentence pair.

[2] Unique placeholder indices are used to allow a correct placeholder identification in post-processing.

[3] In our example, we could assume that a plural noun was to be translated by a plural noun, but without sentence analysis, we could not have guessed that a past participle was to be translated by an infinitive form.

[4] The example illustrates the *append* alternative presented in Dinu et al. (2019).

fuzzy matching. Results show that the model acquires the ability to reuse the appended translations when producing its own hypotheses. The authors show impressive translation accuracy improvements when sufficiently large fuzzy matches exist in translation memories.

Constrained decoding enforces translation terms as decoding constraints applied at inference. Among others, Hokamp and Liu (2017) introduced grid beam search (GBS), an algorithm which employs a separate beam for each lexical constraint (translation term) aiming at ensuring the apparition of each given constraint in the translation hypothesis. The algorithm explores all possible constraints at each time-step, making sure not to generate a constraint that has already been generated in previous time-steps. The approach generates all the constraints in the final output. Other works (Hasler et al., 2018; Post and Vilar, 2018; Susanto et al., 2020) attempt to reduce the computational problem caused by using multiple beams in the inference, a well known weakness of this approach. Similar to the previous approach, constrained decoding does not consider target context when inserting translation terms, as it sets the target form and then produces a target context that fits this constraint. However, in a more realistic scenario, a source term may have multiple translation term inflections among which the MT engine should on-the-fly select the best one depending on the source and target context.

Previously, Chatterjee et al. (2017) proposed a guide mechanism to enhance an NMT network with the ability to prioritize translation options presented in the form of XML annotations of source words. The mechanism is applied at every inference time-step, where the beam search is influenced with external suggestions coming from the attention model. Similarly, Zhang et al. (2018) exploit a search engine to retrieve sentence pairs whose source sides are similar with the input sentence, from which they collect translation pieces. Then, the NMT model is modified to give an additional bonus to output sentences that contain the collected translation pieces.

Our contribution In this article, we compare several methods for domain terminology integration, seen as dynamic adaptation of a model trained on generic data to a specialized domain through terminology only. While results are expected to be lower than those obtained through fine-tuning (training more iterations with specialized parallel corpus), specializing with terminology only is a very frequent use case in industry, given that maintaining terminology lists make sense for experts to factorize the knowledge of frequently translated terms. We do not evaluate constrained decoding since comparison in Dinu et al. (2019) underlined that it did not outperform in-line terminology neither in BLEU nor in term usage rate, and its substantially increased decoding speed does not suit production environments.

3 Terminology Injection

This work builds on the placeholder method presented above. We extend the approach and adapt it to cover a wider variety of cases, and to control morphology to allow generalization power. To represent terminology we use several placeholders indicating part-of-speech (POS) and morphological information, both in source and target sides.[5] For each source-target term pair, we encode all possible inflections of the source and target word labelled with inflection type. 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. The model can then learn to translate a sequence of dedicated placeholders in source by a corresponding sequence of placeholders in target, this way providing the post-process with enough information to choose the right form among the multiple ones available for each translation term, thus ensuring the correct grammatical inflection in inference. Consider the previous example with extended placeholder annotations:

These <noun_ or_ adj#1> <plural_ masculine> can be <verb#2> <past_ participle> .
Se pueden <verb#2> <infinitive> esos <noun#1> <plural> .

A challenging case concerns homographs like the word *precedents* above. Source-side annotations indicate the homograph that can occur as a *noun* or an *adjective*, inflected in *plural* form. We also find it useful to convey in the source some information about the target word, namely that it is *masculine*,

[5]Linguistic annotations are obtained by an in-house toolkit.

for the model to better integrate it in translation (article, agreement,...). The second term *extended* is unambiguously a *verb* in *past participle*. Target-side annotations indicate that in the context of this example, the homograph *precedents* translates into Spanish as a *noun* in *plural* while the second term *extended* translates into a *verb* in *infinitive* tense. Annotations vary according to the language pair. For example, to control inflections in English-Spanish, we annotate the following properties of each POS category:

	in English (source)	in Spanish (target)
noun	- number: s/p - gender of the Spanish noun: m/f	- number: s/p - gender for some nouns like careers : m/f
adj	- whether the Spanish adjective precedes the noun (+LEFTADJ)[6]	- number: s/p - gender: m/f
verb	- tense: infinitive (W), present (P), ... - person: 3 - number: s[7]	- tense: infinitive (W), present (P), past part. (K), ... - person: 1/2/3 - number: s/p

Homograph terms require morphological information for all possible POS categories. This is further illustrated here with the homograph *close*, for which the model, after seeing enough examples, will learn to disambiguate between *[close ⇝ final]* (*noun*), *[close ⇝ próximo]* (*adjective*) and *[close ⇝ cerrar]* (*verb*). The table below illustrates three English-Spanish translation examples with the word *close* assigned to a different POS category. Each example shows: the English sentence (a); after pre-processing, with source terms replaced by source-side placeholders (b); the Spanish translation with target-side placeholders (c) and the Spanish translation (d).

noun	(a)	*after close of business*
	(b)	*after <NNP_A_V#1> <s_m> <+LEFTADJ> <W> of business*
	(c)	*tras el <N#1> <s> de actividades*
	(d)	*tras el final de actividades*
adj	(a)	*values close to the level observed*
	(b)	*values <NNP_A_V#1> <s_m> <+LEFTADJ> <W> to the level observed*
	(c)	*valores muy <A#1> <mp> a los observados*
	(d)	*valores muy próximos a los observados*
verb	(a)	*close all pages*
	(b)	*<NNP_A_V#1> <s_m> <+LEFTADJ> <W> all pages*
	(c)	*<V#1> <P3s> todas las páginas*
	(d)	*cierra todas las páginas*

Note that our approach does not require performing any linguistic annotation in inference. All annotations are already compiled in the terminology base acquired from specialized data. Following with the example, the word *close* triggers the use of the terminology placeholders: *close ⇝ <NNP_A_V#1> <s_m> <+LEFTADJ> <W>*, indicating that *close* is considered in our specialized terminology either as a *noun*, a *verb* or an *adjective* (b). The NMT network then produces the target hypothesis solving the ambiguity in translation (c), and post-processing converts remaining placeholders[8] into word forms by means of a set of rules (d).

A potential disadvantage of this approach is that actual instances of injected terminology are completely hidden to the neural network, that only handles placeholders, whereas this information can be valuable, with the exceptions of rare words or OOVs. We thus propose a second alternative where the source term is left in the source sentence surrounded by placeholders:

These <NNP_ A#1> precedents <plural_ masculine> can be <V#2> extended <past_ participle> .
Se pueden <V#2> <infinitive> esos <N#1> <plural> .

[6] See *el próximo año/el año próximo* 'the following year' while most adjectives can only succeed to the noun *el año escolar* 'the scholar year'.

[7] Only 3rd person singular is discriminant in English conjugation.

[8] Consistency checks ensure an equal number of placeholders in source and target.

4 Experimental Framework

4.1 Corpora

Detailed statistics of the corpora used in this work are provided in Appendix B. Mainly, we use data coming from generic domains for both training and inference: Parallel Paragraphs crawled from the web (COMM); Proceedings of the European Parliament (EPPS); Legislative texts of the European Union (JRC); News Commentaries (NEWS). We use data from specialized domains for inference only: Documentation from the European Central Bank (ECB); Documents from the European Medicines Agency (EMEA); Localisation files (KDE4). All data is preprocessed using onmt-tokenize-text[9].

4.2 Terminology Bases

Terminology databases are automatically extracted from the training sections of each corpus used in this work (see Appendix B). Parallel data is first word aligned with fast_align[10] before extracting phrase pairs[11]. Pairs are kept as terminology entries when they follow a set of pre-defined POS patterns (see details in Appendix B) and only when pairs appear in the testset. Part-of-speech tagging is performed on both sides by TreeTagger[12]. Lemmatisation is carried out by an in-house linguistic analysis tool and frequency filtering is performed to select only the most relevant translation for each term in the domain. Interestingly enough, we observe that the same term is present in several terminologies with different translations according to the domain as can be seen in the next examples:

accordance (noun)
ECB/NEWS: conformidad (noun)
EMEA: acuerdo (noun)
EPPS/JRC: arreglo (noun)

move (verb)
COMM: migrar (verb)
ECB: pasar (verb)
KDE4: mover (verb)
NEWS: ascender (verb)

move (noun)
JRC: decisión (noun)
KDE4: movimiento (noun)
NEWS: maniobra (noun)

4.3 Neural Machine Translation

Our NMT models follow the state-of-the-art Transformer architecture described in Vaswani et al. (2017) implemented in the OpenNMT-tf[13] toolkit. Before learning, we train a 32K joint byte-pair encoding (Sennrich et al., 2016) not applying on introduced placeholders. Note that all models are learnt using a joint source and target vocabulary and shared word embeddings to allow the injection of target words in the source stream. This is only required by one configuration but it enables a fair comparison and does not harm the rest of models. Additional details of our translation networks are given in Appendix A.

4.4 Experiments

We evaluate the following configurations:

- **app**: the target inflected term is appended to the source term. We use an additional parallel stream (factor) to indicate if each word is a term to inject and its respective belonging to source or target. Word embeddings are built after concatenating both factor embeddings (Dinu et al., 2019):

These$_0$ *precedents*$_1$ *precedentes*$_2$ *can*$_0$ *be*$_0$ *extended*$_1$ *ampliar*$_2$.$_0$
Se pueden ampliar esos precedentes .

- **mrk**: source and target inflected terms are analysed and replaced by marks representing their POS and morphological information:

These <NNP_A#1> <p_m> can be <V#2> <K> .
Se pueden <V#2> <W> esos <N#1> <p> .

[9] https://github.com/OpenNMT/Tokenizer
[10] (Dyer et al., 2013) https://github.com/clab/fast_align
[11] https://github.com/moses-smt/mosesdecoder/tree/master/phrase-extract
[12] https://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger
[13] (Klein et al., 2017) https://github.com/OpenNMT/OpenNMT-tf

- **mrk+**: source and target inflected term are similarly analysed and annotated by marks, with the source term still present in the sentence:

These <NNP_A#1> precedents <p_m> can be <V#2> extended <K> .
Se pueden <V#2> <W> esos <N#1> <p> .

It is worth mentioning that all models are trained from the same data and injected the same terminology, respectively at train and test time. For every testset, we evaluate each model under four different annotation conditions:

- NONE: *no injection* of terms, to control for the performance of models trained with annotations when no annotation is injected in inference.
- MANY: injection of a *large quantity* of terms, to evaluate the ability of each configuration to handle multiple terms in a single sentence.
- ALREADY: injection of a *reasonable quantity* of terms already well translated in the baseline.
- IMPROVE: injection of a *reasonable quantity* of terms not already well translated in the baseline.

We evaluate each terminology injection configuration in equal and separable conditions, to better understand how each term of customer terminology, usually a mix of already well translated terms and terms benefiting from specialized translation, can contribute to translation improvement. To be able to evaluate terminology injection and influence on BLEU score for existing corpora, we place ourselves in a setting where injected terms are necessarily present in the reference. While we acknowledge that it does not fully reproduce a real scenario where there is usually no guarantee about the coverage of customer specialized terminology in the content to translate, however this experimental setting is, compared to the situation evaluated in Dinu et al. (2019):

- closer to applied use cases by evaluating generic models on technical testsets and terminologies,
- more controlled in the term match as it uses morphological analysis instead of approximate match, necessary to match forms such as *sigue* 'follows' from the verb *seguir* 'to follow' and
- more complete as our terminologies cover not only fully inflected nouns, adjectives and verbs, but also noun phrases, verb phrases and homographs, recognizing the role of all these categories to specialize translation.

5 Results

Results in terms of BLEU score (Papineni et al., 2002) computed by `multi-bleu.perl`[14] are reported in Table 1. The NONE condition checks that, when no term is injected and trained for the same number of iterations, all three models trained with annotations (**app**, **mrk**, **mrk+**) reach a performance only slightly lower than the baseline (**tok**). In the case of **mrk** and **mrk+**, we hypothesize that they actually use less rich data during training since the placeholders are not lexicalized.

In the MANY condition, when we inject a high number of terms, the **app** score makes a significant jump in specialized domains only, while scores of the models based on morphological marking (**mrk**, **mrk+**) suffer a substantial decrease in both generic and specialized domains, of higher importance for **mrk** and specialized domains. When we inject a "reasonable" quantity of terms, results highly depend on the nature of the injected terms. In the ALREADY condition, when terms are already well produced by the baseline, terminology injection creates a small drop for all models compared to the baseline, a drop that gets more important for **mrk** and specialized domains. These results indicate that models using morphological marking suffer from not having access to lexical instances, in particular when too many terms are injected, reflecting the limits of these models.

[14] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/multi-bleu.perl

However, in the IMPROVE condition, when injected terms where not already present in the baseline, terminology injection induces a considerable gain, in particular for specialized domains: for **app** (+0.38 in generic domains, +1.25 in specialized domains in average), and of larger magnitude for **mrk** (+0.50 generic, +2.15 specialized) and **mrk+** (+0.71 generic, +2.08 specialized).

		NONE				MANY			ALREADY			IMPROVE		
		tok	**app**	**mrk**	**mrk+**	**app**	**mrk**	**mrk+**	**app**	**mrk**	**mrk+**	**app**	**mrk**	**mrk+**
Generic	COMM	38.28	38.31	38.46	38.40	38.47	37.84	38.06	38.23	38.32	38.18	38.57	38.77	**38.78**
	EPPS	44.08	44.14	43.98	44.20	44.66	41.90	43.31	44.06	43.67	43.99	44.83	44.64	**44.88**
	JRC	53.85	53.59	53.69	53.56	53.49	49.57	51.19	53.25	53.11	53.20	53.91	54.01	**54.13**
	NEWS	44.73	44.27	44.70	45.01	44.84	42.43	43.75	44.25	44.25	44.64	45.12	45.50	**45.96**
	Avg	45.23	45.08	45.21	45.29	45.37	42.94	44.08	44.95	44.84	45.00	45.61	45.73	**45.94**
Specialized	ECB	40.89	40.48	40.82	40.80	40.79	33.82	36.86	39.92	39.18	39.60	41.41	41.74	**41.97**
	EMEA	37.98	37.13	37.60	37.58	38.39	37.14	36.26	36.96	36.65	36.94	39.47	**41.46**	41.10
	KDE4	37.25	36.82	37.14	36.96	40.90	33.42	35.31	36.69	35.81	36.12	39.00	**39.37**	39.28
	Avg	38.71	38.14	38.52	38.45	40.03	34.79	36.14	37.86	37.21	37.55	39.96	**40.86**	40.78

Table 1: BLEU results of different model configurations over generic (up) and specialized (down) test sets and according to different terminology conditions.

5.1 In-depth Evaluation

In parallel to translation quality scores measured by BLEU, we now examine the correct term use rate, as well as the distribution of the different types of errors concerning term integration in the hypothesis, illustrated in Table 2.

Error Type	Source	Reference	Translation
Case	As can be seen from **Chart** 4 , [...]	Como se desprende del **gráfico** 4 , [...]	Como se puede ver en el **Gráfico** 4 , [...]
Inflection	In addition , spot transactions which have been contracted but which have not yet been **settled** should be included in the data .	Deben declararse , además , las operaciones al contado perfeccionadas pero aún no **liquidadas** .	Además , las operaciones puntuales que se hayan contraído pero que aún no se hayan **liquidado** deben incluirse en los datos .
Homography	**Driving** and using machines Ciprofloxacin Kabi can reduce your attention .	**Conducción** y uso de máquinas Ciprofloxacino Kabi puede disminuir su atención .	**Conducir** y usar máquinas Ciprofloxacino Kabi puede reducir su atención .
Absence	Mixtard 10 NovoLet 100 IU / ml suspension for injection in a **pre-filled pen** .	Mixtard 10 NovoLet , 100 UI / ml suspensión inyectable en una **pluma precargada**	Mezcla 10 NovoLet 100 IU / ml de suspensión para inyección en un prellenado bolígrafo .

Table 2: Examples of error types for terminology injection.

We identify the following types of error:

- **Case**: the term is integrated with a different casing (*gráfico* Vs. *Gráfico*) than in the reference.
- **Inflection**: the term is integrated with a different inflection than in the reference (includes number, gender and verb form errors). Note that the sentence stays perfectly grammatical as the model integrates the chosen term with a different but correct inflection (*liquidadas* Vs. *liquidado*).
- **Homography**: the integrated term is not the one in the reference, but corresponds to an homograph in source. This error does not necessarily make the sentence nonsensical, in the example translating a gerund *driving* by an infinitive verb *conducir* instead of a noun *conducción*.
- **Absence**: the term does not appear in the translation, with any difference of case, inflection or a form corresponding to an homograph, which means that the model has chosen to ignore the annotation to build its translation.

Table 3 summarizes statistics of the error types observed in test sets. In MANY and ALREADY conditions, for both generic and specialized domains, **app** presents the highest term use rate, higher than the baseline and the **mrk** models. However, a closer look indicates that the errors made by **mrk** models are mostly due to case (around 6%) and inflection (10%), errors that may not necessarily make the translation worse for the human evaluator (see examples [1] and [2] in Table 4), while most errors of **app** come from its non-injection of the desired term (5%). We also verify that in the MANY condition, introducing too many terms does not help the models to generate consistent translations as it blurs the sequence of words in the sentence [3].

		tok	**app**	**mrk**	**mrk+**	**tok**	**app**	**mrk**	**mrk+**	**tok**	**app**	**mrk**	**mrk+**
		MANY (4963 TERMS)				ALREADY (546 TERMS)				IMPROVE (575 TERMS)			
Generic	BLEU gain		+0.13	-2.30	-1.15		-0.28	-0.39	-0.23		+0.38	+0.50	+0.71
	Correct use	83	92	83	85	93	96	86	87	3	64	80	80
	Case	1	1	7	7	1	1	5	5	0	1	4	3
	Inflection	4	2	9	7	5	2	9	8	0	2	14	13
	Homography	0	0	0	0	0	0	0	0	1	0	1	1
	Absence	12	5	1	1	1	1	0	0	96	33	1	3
		MANY (27744 TERMS)				ALREADY (5194 TERMS)				IMPROVE (5127 TERMS)			
Specialized	BLEU gain		+1.32	-3.91	-2.56		-0.85	-1.49	-1.15		+1.25	+2.15	+2.08
	Correct use	72	87	78	81	87	93	83	84	2	62	75	76
	Case	3	1	6	6	3	1	5	5	0	1	9	8
	Inflection	6	3	12	10	8	3	11	10	0	2	13	12
	Homography	0	0	1	0	0	0	0	0	1	0	2	1
	Absence	19	9	3	2	1	3	0	1	96	35	2	2

Table 3: BLEU gain and linguistic analysis of injected terms in translation (distribution in %).

Coming to the IMPROVE condition, case and inflection errors persist at a similar rate for **mrk** models, but the rate of absent terms for **app** is growing to reach a noticeable level (34%): in all these cases, **app** prefers to ignore the information about the terminology it has been given to favor its own translation [4], sometimes identical to the baseline **tok**. Critically, this freedom leads **app** to fail injecting an irregular verb form, making the sentence ungrammatical [5], and complete drug names, making the translation much less secure to use [6,7]. With respect to **mrk** and **mrk+** models, they have comparably high term injection rates but **mrk** offers slightly higher BLEU in specialized domains, and more control over the injected term: in particular for multiple-word terms, we have observed that **mrk+** could erroneously repeat part of a compound [7], but **mrk**, that is blind to the injected terms being single or multiple words, can integrate both seamlessly.

6 Conclusion

Our major finding is that, in a context where the terminology introduces specialized terms that were not already well translated by the baseline, the **app** model – appending terminology as inline annotations in the source text – fails to inject terms at 34% and therefore does not guarantee the presence of expected terms in translations. This can be highly critical in a real setting when the user wants terminology to enforce the use of certified brands, product names, acronyms, but also business concepts, such as noun phrases and verbs. With the constraints that ones need to curate highly detailed linguistic resources and that the quantity of injected terms needs to be limited, the **mrk** models – representing expected terms by their morphological analysis – offer further guarantee of term injection with an absence rate of only 2%: when the exact term cannot be injected, the model usually injects a case or inflection variation that fits the translation. Additionally, the model can handle intricate patterns that are part of a vast majority of languages such as irregular forms, complex noun or verbal phrases, as well as multi-part and contextual entries. In contrast with the **app** model, that simply learns a copy behaviour from source to target agnostic to the context, the **mrk** models leverage the inner language knowledge of the neural network to perform morphological and syntactic analysis of the source, and more seamlessly generate the target.

[1] **mrk** models restore source case, different from reference

src	Tacrolimus may prolong the **QT interval** but at this time lacks substantial evidence ...
ref	Tacrolimus puede prolongar el intervalo QT , sin embargo en este momento falta evidencia sustancial ...
tok	Tacrolimus puede prolongar el intervalo de QT , pero en este momento carece de pruebas sustanciales ...
app	Tacrolimus puede prolongar el intervalo QT pero en este momento carece de pruebas sustanciales ...
mrk	Tacrolimus puede prolongar el Intervalo QT , pero en este momento carece de pruebas sustanciales ...
mrk+	Tacrolimus puede prolongar el Intervalo QT , pero en este momento carece de pruebas sustanciales ...

[2] source term is ambiguous, no model manages to get it correct

src	Tunnel device is missing , creating it has failed : **stop** .
ref	Falta el dispositivo de túnel , falló la creación : detenido .
tok	El dispositivo de túnel no está , lo que lo ha creado ha fallado : parada .
app	El dispositivo de túnel no está presente , lo que lo ha fallado : la detención .
mrk	Faltan dispositivos de túnel , lo que la ha fallado : detenga .
mrk+	El dispositivo de túnel falta , lo que lo ha fallado : detener .

[3] translations can become inconsistent when injected too many terms

src	**Increase** in **cholesterol** and triglyceride levels , **hyponatremia**
ref	Aumento de las concentraciones de colesterol y de triglicéridos , hiponatremia
tok	Aumento de los niveles de colesterol y triglicéridos , hipoponatremia
app	Aumento del colesterol y de los niveles de ataque , hiponatremia
mrk	Aumento en los niveles de colesterol y de la férula , hiponatremia
mrk+	Aumentar en los niveles de colesterol y de los sistemas de retención , hiponatremia

[4] **app** fails to inject the correct term contrary to **mrk** models (but *jeringa* is in vocabulary)

src	Keep the **syringe** in the outer carton in order to protect from light .
ref	Mantener la jeringa en el embalaje exterior para protegerla de la luz .
tok	Mantenga la aguja en el cartón exterior para protegerse de la luz .
app	Mantenga la munición en el cartón exterior para proteger de la luz .
mrk	Mantenga la jeringa en el cartón exterior para proteger de la luz .
mrk+	Mantenga la jeringa en el cartón exterior para proteger de la luz .

[5] **app** fails to correctly inflect an irregular verb (*indujo* is OOV)

src	Tenecteplase **induced** total litter deaths during the mid-embryonal period .
ref	La tenecteplasa indujo la muerte total de la descendencia durante el periodo embrionario medio .
tok	La Tenecteplasa indució la muerte total de la basura durante el período embrionario medio .
app	La Tenecteplasa indució las muertes totales de despojos durante el período de mitad embrional .
mrk	La tenecteplase indujo la muerte total de la camada durante el periodo medio embrionario .
mrk+	La tenecteplasa indujo la muerte total de la camada durante el periodo de la mitad de embriones .

[6] **app** fails to inject a drug name (*TYSABRI* is OOV)

src	Use of **TYSABRI** has been associated with an increased risk of PML .
ref	El uso de TYSABRI se ha asociado a un incremento del riesgo de LMP .
tok	El uso de TYSABIRON ha sido asociado con un mayor riesgo de PML .
app	El uso de la TYSALine se ha asociado con un mayor riesgo de PML .
mrk	El uso de TYSABRI se ha asociado con un mayor riesgo de PML .
mrk+	El uso de TYSABRI se ha asociado con un mayor riesgo de PML .

[7] **app** fails to inject a multi-word drug name, **mrk+** repeats part of it

src	- tell you when you may need to use a higher or lower dose of **Insuman Infusat** ,
ref	- le indicará cuándo puede necesitar inyectarse una dosis más alta o más baja de Insuman Infusat .
tok	- decirle cuándo puede necesitar utilizar una dosis más alta o más baja de Infusat ,
app	- le indique cuándo puede necesitar utilizar una dosis mayor o menor de Infusat de Seguro ,
mrk	- le informarán cuando necesite utilizar una dosis superior o inferior de Insuman Infusat ,
mrk+	- indicarle cuándo puede necesitar utilizar una dosis mayor o menor de Insuman Infusat Infusat ,

Table 4: Examples of translations with terminology injection (in bold **source**, in blue expected terms; in red injection errors: case [1], inflection [2], word choice [4-7] ; in green bad translations [3, 5]).

Acknowledgements

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

References

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, et al. 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.

Bram Bulté 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, July. Association for Computational Linguistics.

Rajen Chatterjee, Matteo Negri, Marco Turchi, Marcello Federico, Lucia Specia, and Frédéric Blain. 2017. Guiding neural machine translation decoding with external knowledge. In *Proceedings of the Second Conference on Machine Translation*, pages 157–168, Copenhagen, Denmark, September. Association for Computational Linguistics.

Rajen Chatterjee, Christian Federmann, Matteo Negri, and Marco Turchi. 2019. Findings of the wmt 2019 shared task on automatic post-editing. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 11–28.

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, July. 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, July. 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, June. 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, September. 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, June. Association for Computational Linguistics.

Chris Hokamp and Qun Liu. 2017. Lexically constrained decoding for sequence generation using grid beam search. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1535–1546, Vancouver, Canada, July. Association for Computational Linguistics.

Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR Conference Track Proceedings*, San Diego, CA, USA, may.

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, August. Association for Computational Linguistics.

Thang Luong, Ilya Sutskever, Quoc Le, Oriol Vinyals, and Wojciech Zaremba. 2015. Addressing the rare word problem in neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 11–19, Beijing, China, July. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July. Association for Computational Linguistics.

Alvaro Peris and Francisco Casacuberta. 2019. Online learning for effort reduction in interactive neural machine translation. *Computer Speech & Language*, 58:98–126.

Matt Post and David Vilar. 2018. Fast lexically constrained decoding with dynamic beam allocation for neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1314–1324, New Orleans, Louisiana, June. 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.

Jean Senellart, Jin Yang, Anabel Rebollo, et al. 2003. Systran intuitive coding technology. *MT Summit IX: Proceedings of the ninth machine transaltion summit, New Orleans, USA*, pages 346–353.

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, August. Association for Computational Linguistics.

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, June. Association for Computational Linguistics.

Raymond Hendy Susanto, Shamil Chollampatt, and Liling Tan. 2020. Lexically constrained neural machine translation with Levenshtein transformer. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3536–3543, Online, July. 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, May. 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, July. Association for Computational Linguistics.

Jingyi Zhang, Masao Utiyama, Eiichro Sumita, Graham Neubig, and Satoshi Nakamura. 2018. Guiding neural machine translation with retrieved translation pieces. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1325–1335, New Orleans, Louisiana, June. Association for Computational Linguistics.

A NMT Configuration

The next table gives details of the network configuration used in our experiments:

N	d	d_{ff}	h	V	$batch$	$Optimization$	$Updates$	$beam$
6	512	2,018	8	32,000	2,048	lazy Adam	300,000	5

where N is the number of layers; d is the size of both word embeddings and hidden layers; d_{ff} is the size of inner feed forward layer; h is the number of heads; V is the length of the joint vocabulary used; $batch$ is the learning batch size (in number of tokens) and $beam$ indicates the inference beam size. For Adam (Kingma and Ba, 2015) optimization we set warm-up steps to $4,000$ and update learning rate for every 8 iterations. For training and inference we use a single NVIDIA P100 GPU.

B Corpora Statistics

Tables 5 and 6 respectively illustrate statistics of the different corpora used in this work[15] and the number of injected terms according to POS patterns.

Corpora are randomly split, keeping 500 sentences for validation, $2,000$ (or $8,000$) for testing and the rest for training.

Corpora		Generic				Specialized		
		COMM	EPPS	JRC	NEWS	ECB	EMEA	KDE4
Train sets (K)								
Sentences		1,158	723	847	273			
Words	en	21,437	14,805	19,620	6,033			
	es	22,889	15,283	22,125	7,107			
Vocab	en	717	155	293	184			
	es	821	216	313	215			
Test sets								
Sentences		2,000	2,000	2,000	2,000	8,000	8,000	8,000
Words	en	36,557	40,731	45,267	44,619	191,316	125,843	69,752
	es	39,038	42,039	50,663	52,447	214,411	141,871	78,773
Vocab	en	11,543	7,580	8,770	11,560	17,380	18,247	12,505
	es	12,499	8,897	9,834	12,220	21,058	20,654	13,322
OOV	en	642	103	305	402	4,553	9,860	2,420
	es	788	134	298	456	4,687	10,036	2,351

Table 5: Statistics over train/test corpora after splitting-off punctuation. Training figures are given in thousands (K), en and es stand for English and Spanish respectively. Out-of-vocabulary words (OOV) are computed over all Generic train corpora.

[15] Freely available from `http://opus.nlpl.eu` (Tiedemann, 2012)

Term type	Generic				Specialized		
	COMM	EPPS	JRC	NEWS	ECB	EMEA	KDE4
Train sets							
A	37,161	30,553	32,922	18,409			
A-V	2,225	2,276	1,294	855			
N	209,537	148,315	217,981	68,785			
NNP-A	3,631	4,092	3,361	1,497			
NNP-V	6,264	5,002	4,369	2,175			
NP	3,561	1,808	2,176	1,452			
V	31,214	19,354	16,694	9,000			
Test sets (IMPROVE)							
A	19	48	13	49	316	232	218
A-V	-	1	-	-	12	20	8
N	164	250	302	393	3,889	4,813	2,704
NNP-A	2	-	1	1	13	15	8
NNP-V	-	12	2	14	373	122	239
NP	6	6	3	15	118	179	32
V	25	69	45	103	373	597	1,088
Test sets (ALREADY)							
A	23	38	11	45	249	257	177
A-V	-	2	-	2	9	9	17
N	154	282	276	361	4,043	4,594	2,762
NNP-A	-	-	3	-	71	11	12
NNP-V	1	3	-	4	181	108	250
NP	13	27	8	55	148	302	20
V	27	47	19	79	559	765	1,032
Test sets (MANY)							
A	120	383	182	425	1,730	1,232	952
A-V	-	20	-	3	62	43	69
N	883	2,349	3,349	3,372	25,671	21,242	14,320
NNP-A	2	-	14	4	333	59	45
NNP-V	3	32	14	63	1,265	562	1,276
NP	78	127	59	414	883	1,119	141
V	159	386	217	682	3,522	3,219	5,431

Table 6: Terminology statistics (English) according to POS patterns and data sets annotated on the English-side of the corpora.