Terminology-Constrained Neural Machine Translation at SAP

Miriam Exel Bianka Buschbeck Lauritz Brandt
SAP SE
Dietmar-Hopp-Allee 16, 69190 Walldorf
Germany
firstname.lastname@sap.com

Simona Doneva[*]
University of Mannheim
68131 Mannheim
Germany
sdoneva@mail.uni-mannheim.de

Abstract

This paper examines approaches to bias a neural machine translation model to adhere to terminology constraints in an industrial setup. In particular, we investigate variations of the approach by Dinu et al. (2019), which uses inline annotation of the target terms in the source segment plus source factor embeddings during training and inference, and compare them to constrained decoding. We describe the challenges with respect to terminology in our usage scenario at SAP and show how far the investigated methods can help to overcome them. We extend the original study to a new language pair and provide an in-depth evaluation including an error classification and a human evaluation.

1 Introduction

With over one billion words per year, SAP deals with a huge translation volume; covering product localization and translation of documentation, training materials or support instructions for up to 85 languages. With a wide range of product lines in different industries, translation settings are diverse. There are over 100 active translation domains for which we maintain translation resources such as translation memories and terminologies. At SAP we usually train multi-domain neural machine translation (NMT) engines, whose input consists of a multitude of data sources including the contents of the company-internal translation memories from various domains. The resulting NMT system produces high-quality technical translations, but has difficulties generating appropriate and coherent terminology in specific contexts. Given the great importance of correct and consistent terminology in technical translation, this is a nuisance for the translators that work in a post-editing scenario as well as for users consuming machine translation (MT) in a self-service scenario.

In our translation environment, translators are assigned projects along with the relevant translation domain's terminology. To achieve term consistency, SAP maintains SAPterm[1], a large terminology database which also specifies viable term translations. Translators can easily select target terms from SAPterm in a computer-assisted translation (CAT) environment, but applying terminology constraints in NMT is a challenge. As we do not have reliable term recognition or morphological inflection generation tools for all our productive languages at our disposal, we require an approach that not only enforces the correct terminology but also learns its contextually appropriate inflections.

To that end, we investigate the approach presented in Dinu et al. (2019), which combines inline annotation with source factors (Sennrich and Haddow, 2016), that provide an additional input stream with terminology annotation, to show how domain-specific terminology can be enforced in a multi-domain NMT model. The approach should be capable of handling unseen terminology while retaining NMT's ability to produce fluent output sequences without the need for additional resources such as morphological generators and without drastically reducing decoding speed. We will present results for variations of this approach which were not investigated in Dinu et al. (2019),

[*] Employed as a working student at SAP during this project.

© 2020 The authors. This article is licensed under a Creative Commons 3.0 licence, no derivative works, attribution, CC-BY-ND.

[1] http://www.sapterm.com/

but could be of interest to users of NMT who plan to implement that approach in a productive system.

While the WMT news translation task that Dinu et al. (2019) evaluate on is a viable test bed for new methods, we aim to validate that the method is also applicable to other scenarios, such as the translation of texts from the business and IT context of SAP, when constraining it with entries from SAPterm. We furthermore extend the original study to a new language pair (English–Russian) and provide an in-depth evaluation including a human assessment. Our study yields very promising results, amongst others improvements of up to 11 BLEU points on terminology data, and paves the way to the customization of NMT at SAP: a selected SAPterm glossary can be applied directly when producing MT proposals for a translation project. This yields better translation quality, helps to reduce post-editing costs and eases translators' frustration with correcting terms.

2 Related Work

Several approaches to make NMT adapt to a domain-specific terminology have been proposed in the literature. Fine-tuning on in-domain training data on-the-fly (Farajian et al., 2018; Huck et al., 2019) is shown to improve translation quality and term accuracy but creates additional technological challenges for model management and increases infrastructure costs. Additionally, terminology constraints cannot be specified on a sentence or document level, but instead need to be distinctly present in the available training data, which often is not the case in a productive scenario. The latter argument also holds for domain-aware MT (Kobus et al., 2017), where a multi-domain model distinguishes the translation domains using a domain tag, which is prepended to the source segment.

Since terminology databases are available in most translation environments, integrating them into NMT at run-time to enable domain-specific translation is an ongoing research topic. Early approaches use placeholder tokens for source and target (for example (Crego et al., 2016)). Placeholder approaches often suffer from disfluency as the NMT model does not have access to the term and therefore has difficulties creating a fluent and morphologically sound translation.

Constrained decoding is one of the most prominent approaches to enforcing terminology in NMT. The decoder is subject to a set of constraints that are strictly enforced during decoding (Hokamp and Liu, 2017; Chatterjee et al., 2017). Some issues with constrained decoding have already been addressed, such as better positioning of target terms by exploiting the correspondence between source and target terms (Hasler et al., 2018), and improving performance for the base approach (Post and Vilar, 2018; Hu et al., 2019). Nevertheless, the increase in decoding time compared to unconstrained decoding is still considerable (cf. Section 5). Also the output surface form is enforced exactly as provided by the constraint and no morphological adaptation is applied by the decoder. This leads to misplaced constraints and broken sentences (Burlot, 2019) as well as special cases where surface form variants of an enforced term are being produced by the decoder but not picked up by the constraint, leading to a duplication as the constraint produces the terminology again (Dinu et al., 2019).

Dinu et al. (2019) offer a different approach to applying terminology constraints in NMT. The target terms are inserted into the source string during training and decoding, and thus the model learns a copying behavior. An indication of which words are source terms, target terms or no terms is provided to the model via an additional input stream. This input is encoded as source factors, in the same way that linguistic features can be encoded (Sennrich and Haddow, 2016). For the English–German WMT 2018 news translation task, moderate improvements in BLEU and term accuracies >90% are reported. The zero-shot nature of this approach enables the application of unseen terminology at test time. Furthermore, Dinu et al. (2019) report cases of generating morphological variants of terminology entries in the output, while decoding times are not increased compared to the base model. As the ability to apply terminology constraints is *trained* into the NMT model *by* either appending the target term to the source term or *by* replacing it, Dinu et al. (2019) refer to their models as *train-by* models, and we will continue doing so.

Many commercial providers of MT offer an option to upload a user dictionary in order to customize the NMT output to enforce a certain terminology.[2] This is a feature that users became ac-

[2] Accessed on February 21st, 2020:
Amazon Translate: https://aws.amazon.com/blogs/machine-learning/introducing-amazon-translate-custom-terminology/
Google Translate: https://cloud.google.com/translate/docs/advanced/glossary
Microsoft Translator: https://docs.microsoft.com/en-us/azure/cognitive-services/translator/dynamic-dictionary

customed to in rule-based and statistical MT, and consequently they expect a similar functionality for NMT as well. Naturally, the commercial providers usually leave us in the dark about the technology that is used for the implementation of that feature. Such custom terminology features are described more for marketing purposes rather than from an objective technical viewpoint. Usually, no transparent evaluation results are available. Some product descriptions are nevertheless fair enough to describe the limitations of the feature and best practices.

3 Methodology

We experiment with variants of the *train-by* approach introduced by Dinu et al. (2019), which is a form of inline term annotation. Target terms t_t are inserted into a source sentence by either appending them to the source term t_s (*append*) or by replacing t_s completely (*replace*). An additional signal is provided by a term annotation for each input token, where 1 means part of a source term, 2 means part of a target term and 0 is the default. An example for the input is provided in Table 1.

The term annotations are presented as source factors and have their own embedding vectors, which are combined with the respective (sub-) word embeddings to represent the input of the encoder in an encoder-decoder NMT architecture (Sennrich and Haddow, 2016). The two embedding vectors can be combined by either concatenating (*concat*) or summing (*sum*) them. This makes the dimensionality of the source factor embedding either a variable-sized (*concat*) or a fixed sized (*sum*) vector. While Dinu et al. (2019) only report results for the concatenation strategy with an embedding size of 16, we investigate an embedding size of 8 as well as the vector summarization combination.

We are also interested in the impact of the source factors themselves, and thus investigate whether the additionally provided annotation is actually necessary by using only the inline annotation and no term factor annotation.

The source sentences are annotated as described for all terminology entries (t_s, t_t), when t_s is present in the source and t_t occurs in the reference. To check whether a term occurs in a sentence, we use a matching strategy that also covers morphological variants. This is essential as our terminological database contains base forms only. Note that we insert t_t into the source in its base form, because this will also be the scenario at test time.

During training, the model learns to copy the injected target terms to the output. We expect to see morphological variants of the base terms in the output in accordance with the context of the sentence, as is reported in Dinu et al. (2019).

4 Experimental Setup

We evaluate the application of terminology constraints in the usage scenario of MT at SAP, for two language pairs English–German (en–de) and English-Russian (en–ru). We use target languages that are relatively morphologically rich because we want to investigate whether the approach is able to produce the target terms in an appropriate morphological form.

4.1 Data and Data Preparation

Corpus Our parallel data consists of a large collection of proprietary translation memories from within SAP. It is a multi-domain corpus covering different content types, such as documentation, user interface strings and training material in relation to various SAP products. For all our training/validation/test sets we use 5,000,000/2,000/3,000 parallel segments respectively. We use two test sets, where the first is targeted towards the evaluation of terminology and contains at least one terminology entry pair in each sentence, whereas the other does not have terminology annotated. We will refer to them as *terminology* and *no-terminology* test sets respectively.

Terminology SAPterm is organized into concepts where terms that are translations of each other are linked. A concept can cover different term types, such as a main term entry, its synonyms, acronyms or abbreviations. To generate a high-quality glossary, we only consider source-target term pairs consisting of main term entries and their synonyms. To avoid common words and spurious entries, we filter out high-frequency and low-frequency entries.[3] We therefore only select a subset of all entries in SAPterm, consisting of

SDL: https://www.sdl.com/about/news-media/press/2018/sdls-neural-machine-translation-sets-new-industry-standards-with-state-of-the-art-dictionary-and-image-translation-features.html
Systran: https://blog.systransoft.com/our-neural-network-just-learned-syntax/

[3] We filter out term pairs where the English side occurs more than 5,000 times or less than 100 times in a large corpus (>20 million sentences) of proprietary SAP data.

append	en	This$_0$ indicator$_0$ is$_0$ only$_0$ necessary$_0$ for$_0$ **manual**$_1$ **depreciation**$_1$ **manuelle**$_2$ **Abschreibung**$_2$ and$_0$ write-ups$_0$.$_0$
replace	en	This$_0$ indicator$_0$ is$_0$ only$_0$ necessary$_0$ for$_0$ **manuelle**$_2$ **Abschreibung**$_2$ and$_0$ write-ups$_0$.$_0$
Ref.	de	Das Kennzeichen wird nur für manuelle Abschreibungen und Zuschreibungen benötigt .

Table 1: Example input for the two term injection methods *append* and *replace*. Source factors are indicated as indices. The terminology entry is (manual depreciation, manuelle Abschreibung).

	en–de	en–ru
train	784,666	582,281
validation	303	238
terminology test	4,868	3,510
no-terminology test	0	0

Table 2: Number of term annotations

116,188 entries for English–Russian and 153,417 entries for English–German.

We apply a fuzzy matching strategy to find and annotate the terms in our data, as motivated in Section 3. Specifically, we lemmatize[4] on the English side, and allow for differences of two characters on the target side. In case of multiple overlapping matches, we keep only the longest match. Inspired by Dinu et al. (2019), we strictly separate training and testing terminology entries and select our parallel data accordingly to demonstrate the zero-shot learning capabilities of the model. For *train-by* methods we annotate 10% of the training and validation segments with terminology using the training terms. The term annotation statistics can be found in Table 2.

Preprocessing We tokenize all data using *NLTK*[5] and perform a joint source and target BPE encoding (Sennrich et al., 2016) using 89.5k merge operations. We furthermore inject the target terms for annotated terms according to the *append* and *replace* methods and generate source factors on BPE-level accordingly (cf. Table 1).

4.2 NMT Models

We make use of the Sockeye toolkit (Hieber et al., 2018) for this investigation. It supports source factors and constrained decoding out-of-the-box.[6]

For all our experiments, we use a transformer network (Vaswani et al., 2017). We configure two encoding and two decoding layers, unless stated otherwise. We also conduct experiments with a six layer setup (*6 layers*), which corresponds to the base configuration of Vaswani et al. (2017). The early stopping criterion is computed on the validation data (32 validation runs without improvement). All evaluations are performed with beam size 5.

For both the *append* and *replace* method, we train and evaluate models in which the embedding of the term annotation is added or concatenated to the corresponding subword embedding. We experiment with embedding sizes of 8 and 16 for concatenation. To investigate the impact of the term annotation in the form of source factors, we also train and evaluate models without source factors (*nofactors*), while still using the term injection of the *append* and *replace* method.

For comparison, we train a baseline without injected terms and source factors. We further compare against Sockeye's implementation of constrained decoding, which is based on Post and Vilar (2018). For this, we use the baseline model and constrain the output to contain the target terms of the terminology entries that are annotated in the *terminology* test set.

5 Automatic Evaluation

In this section we present the results of our experiments using automatic evaluation.

5.1 Metrics

To automatically assess the translation quality, we report BLEU (Papineni et al., 2002) and CHRF (Popović, 2015) on de-BPEed output, using the implementation in *NLTK*[7]. To evaluate how well the models adhere to the terminology constraints, we report *term rates* (TR), computed as the percentage of times the target term is generated in the MT output out of the total number of term annotations. We also employ the previously used fuzzy matching strategy to match the words in the output against the annotated terms in the reference. Note that we are not interested in generating the exact morpho-

[4] http://www.nltk.org/api/nltk.stem.html#module-nltk.stem.wordnet

[5] https://www.nltk.org/api/nltk.tokenize.html

[6] https://awslabs.github.io/sockeye/training.html

[7] https://www.nltk.org/api/nltk.translate.html

logical form of the term that occurs in the reference or in the terminology database, but we want the term in whatever form is required in the sentential context of the MT output. We also report the *variant term rate* (variant TR), in which a target term is also counted as correct if it coincides with one of the other possible translations of the source term according to SAPterm. We are aware that those term rates only approximate the truth, as do all automatic MT evaluation metrics. Hence we quantify some shortcomings in Section 7.2 and add a human evaluation in Section 6.

5.2 Results

Results for en–de and en–ru can be found in Tables 3 and 4 respectively. Our *train-by* systems are labeled according to whether they use the *append* or *replace* method from Dinu et al. (2019) and which kind of source factor embedding strategy they employ. We present results for the test sets *terminology* and *no-terminology* separately. The first allows us to demonstrate how the different approaches fare in terms of translation quality and term accuracy, while the latter serves as a sanity check to make sure that the general translation quality does not suffer for data without terminology.

The first thing to note is that BLEU scores for en–ru on the *terminology* data set are a lot higher than for en–de. This can be explained by the test sets that differ in sentence length and grammatical complexity. With an average of 17.7 words, the en–de data contains a large number of longer sentences with a higher term density. The en–ru data in contrast contains many short simple sentences with an average of 9.04 words per segment with mostly only one term.

Terminology test data It can be easily seen that, for both language pairs, all *train-by* models outperform the baseline in terms of translation quality and term rate by a wide margin. Comparing the term rate with the variant term rate for the individual models reveals that, while the baseline sometimes chooses an alternative translation for a term, this does not hold for the *train-by* models where the two term rates are basically the same. Overall, the results show that the *train-by* approach is effective in improving the translation quality using terminology constraints in the evaluated usage scenario of SAP data annotated with terminology from SAPterm.

Taking all results into account, the *append* method works better than the *replace* method for our experimental setup. Looking only at the *append* method results, concatenation of the two embedding vectors works better than summarization. From the approaches that use source factors, the *append-concat16* setting consistently performs best, both in terms of overall translation quality and term rate. This finding holds for both language pairs.

We rerun the most promising setting as well as the baseline with the six-layer transformer for en–de. As expected, both show an improvement for all metrics over their respective two-layer counterpart. The finding that the *append-concat16* approach outperforms the baseline in terms of translation quality and term rate by a wide margin thus holds for the shallow model as well as for the deeper model.

Somewhat surprisingly, we can observe that the impact of source factors is small for en–de and nonexistent, or even slightly detrimental for en–ru. It seems that the model has learned the code switching that happens in the source sentence and the intended copy behavior of the injected terms to the output, without requiring the additional input signal. We hypothesize that the different scripts of English and Russian, Latin and Cyrillic, are the reason why the model picks up the code switching better than for en-de, which both use the Latin alphabet.

Finally, when comparing the *train-by* methods to constrained decoding, we observe that even though constrained decoding reaches almost perfect term rates (>99%), the overall translation quality that is achieved with the *train-by* models is clearly superior. The decrease in BLEU further confirms observations that have previously been made in the literature (cf. Section 2), namely that constrained decoding can sometimes lead to questionable translation quality. In addition, it is important to note that constrained decoding caused an approximate sixfold increase in translation time in our experiments, while no such impact was observed for the *train-by* models.

Test data without terminology The results of the individual approaches on the *no-terminology* test data show slight differences in translation quality as measured by BLEU and CHRF. We deem those to be within the regular variation that we see amongst different training runs with the same data

	terminology				*no-terminology*	
	BLEU	CHRF	TR	Variant TR	BLEU	CHRF
Baseline	42.74	72.11	71.20	76.73	48.02	71.87
Constrained decoding	41.81	73.91	**99.51**	**99.65**	– ” –	– ” –
Append-concat16	**47.08**	**76.06**	96.40	96.52	48.22	72.01
Append-concat8	46.72	75.81	96.30	96.50	47.67	71.59
Append-sum	46.45	75.74	96.24	96.42	47.83	71.62
Replace-concat16	45.41	75.31	96.30	96.34	47.79	71.67
Replace-sum	45.75	75.46	96.44	96.50	48.21	71.99
Append-nofactors	46.19	75.58	95.06	95.43	47.26	71.56
Replace-nofactors	45.50	75.16	95.37	95.52	48.04	72.13
Baseline (6 layers)	43.50	72.66	71.98	77.31	48.66	72.52
Append-concat16 (6 layers)	47.45	76.60	96.87	97.16	48.98	72.79

Table 3: Results for English–German on the *terminology* and *no-terminology* test sets

	terminology				*no-terminology*	
	BLEU	CHRF	TR	Variant TR	BLEU	CHRF
Baseline	50.24	72.57	64.10	69.09	41.79	63.21
Constrained decoding	42.10	78.08	**99.12**	**99.23**	– ” –	– ” –
Append-concat16	61.23	81.06	95.72	95.81	41.80	63.02
Append-sum	60.94	80.91	95.30	95.32	41.77	62.99
Replace-concat16	60.30	80.46	94.92	94.92	42.04	63.11
Replace-sum	60.29	80.33	95.10	95.10	41.87	63.15
Append-nofactors	**61.47**	**81.48**	96.07	96.18	41.98	63.14
Replace-nofactors	60.83	80.67	95.33	95.33	41.78	62.99

Table 4: Results for English–Russian on the *terminology* and *no-terminology* test sets

and configuration. We thus conclude that the *train-by* approach in the investigated setting generally does not seem to have a negative impact on data without terminology constraints.

6 Translators’ Assessment

As we apply MT in post-editing scenarios, it is of importance that our translators approve of our proposed solution of enforcing SAP-specific terminology. Taking the shortcomings of automatic metrics for MT into account, we therefore also conducted a human evaluation.

6.1 Setup

For the human evaluation, we chose to compare the baseline and the two best-performing *train-by* models *append-concat16* and *append-nofactors* from the automatic evaluation. The latter scored surprisingly well, requires less involved preprocessing and a simpler network architecture, which is appealing in a commercial setup. We selected 100 segments from the *terminology* test set (cf. Section 4.1). As we were primarily interested in the differences between the three systems, we made sure that none of the three translations are identical to each other or to the reference translation. We made sure that 35 of the test sentences contain more than one term annotation, to also cover this particular case.

For both language pairs, we had three testers who evaluated the same 300 translations in a blind evaluation using our in-house MT evaluation tool. Testers were shown the source with highlighted terminology, the relevant terminology entries and one translation at a time in random order. They were asked to rate the target term accuracy and the overall translation quality, both on a scale from one (poor) to six (excellent). Note that the human target term accuracy does not directly correspond to the automatic term rates (cf. Section 5), as testers were advised to also consider whether target terms appear in the expected syntactic position and fit mophologically into their context.

6.2 Results

To consolidate the results of the human evaluation, the accuracy and quality ratings of all testers were averaged for each evaluated segment. Table 5 shows the respective results. Generally, they confirm the findings of the automatic evaluation in

	Term accuracy		Transl. quality	
	en–de	en–ru	en–de	en–ru
Baseline	4.52	4.99	4.40	4.90
Append-concat16	5.74	5.70	4.54	4.98
Append-nofactors	5.79	5.69	4.50	4.90

Table 5: Results of human evaluation: term accuracy rating and translation quality rating

Rating	baseline		nofactors		concat16	
	ende	enru	ende	enru	ende	enru
excellent	50%	53%	86%	80%	87%	77%
very good	6%	12%	9%	13%	7%	14%
good	5%	15%	2%	2%	0%	4%
medium	13%	8%	0%	0%	1%	2%
poor	14%	8%	1%	3%	2%	3%
very poor	12%	4%	2%	2%	3%	0%

Table 6: Distribution of term accuracy ratings for baseline and *append* systems

Section 5. In addition, Table 6 shows the distribution of the average term accuracy ratings.

The accuracy of the term translations of the baseline model clearly lags behind the *train-by* models for both language pairs. The results however also show that terminology is quite well covered by the baseline model already.

The term accuracies for *append-concat16* and *append-nofactors* approach the maximum score for both language pairs, and are very close to each other. This gives rise to the conclusion that the approach works similarly well for enforcing terminology on both morphologically average (de) as well as rich (ru) target languages.

In terms of overall translation quality, the difference between the baseline and the *append* systems is less pronounced than suggested by the automatic scores. For both language pairs, the quality ratings of the *append* models are comparable. Term enforcement does not seem to have noticeable negative side effects on overall translation quality.

Human evaluation also reveals that there is no quality loss when more than one term is injected into a sentence. In the 35% of test segments with multiple terms, term accuracies of the *append* models are even sightly higher than for sentences with one term. This also has an effect on the overall translation quality. For *append-concat16*, for example, we see a positive difference of 0.13 (en–de) and 0.18 (en–ru) points between the average quality ratings of sentences with one and with multiple terms.

7 Examples & Discussion

In this section, we present examples of correct term translations as well as an in-depth human analysis of the terms that were not produced according to the automatic evaluation. Examples for en–de and en–ru are displayed in Table 7.

7.1 Analysis of Term Translations

With the high term rates of all *train-by* models (cf. Tables 3 and 4) it is expected that the models adhere well to the terminology constraints. When taking a closer look into the output of *append-concat16*, we make the following observations (examples taken from Table 7):

- Terminology integrates smoothly into the context of the target language using correct morphological forms (ex. 2). This is especially important for a highly inflecting language like Russian where case information is properly transferred (ex. 5, 6)
- Single terms can build natural compound words in German (ex. 3).
- When enforcing nominal terminology, English verb-noun ambiguities are often resolved towards nouns, which is reflected in the translation (ex. 5 compared to baseline). Another effect is the verbal translation of English imperatives instead of using its nominalization (ex. 7 compared to baseline).
- Enforcing nominal terminology leads to less compounding and prevents overcompounding in German target (ex. 4).
- Abbreviations in the translation are prevented. In our case, they are caused by large amounts of training data from heavily abbreviated content (ex. 4 reference and ex. 8 baseline).
- The baseline translation often uses synonyms of the expected term (ex. 2, 6). This means that the translation does not adhere to the terminology constraint, but that it is not completely wrong either.

7.2 Missed Term Translations

We also analyzed sentences for which term enforcement did not work as expected, i.e. the remaining 3.6% and 4.3% from *append-concat16* in Tables 3 and 4 respectively. For this, 75 segments with missing term translations according to the automatic evaluation were analyzed manually. The results of this investigation are shown in Table 8.

(1)	product substitution location substitution	– –	Produktsubstitution Lokationsfindung
Source	**Product Substitution** e.g. no **location substitution** for oversea customer		
Baseline	Produktersetzung, z.B. keine Lokationsersetzung für ÜberseeKunde		
Append-concat16	**Produktsubstitution** z.B. keine **Lokationsfindung** für Überseekunden		
Reference	**Produktsubstitution**; Beispiel: keine **Lokationsfindung** für Überseekunden		
(2)	budget hierarchy budget	– –	Haushaltsstruktur Haushalt
Source	Defining a **budget hierarchy** is the first step in setting up an overall **budget**.		
Baseline	Die Definition einer Budgethierarchie ist der erste Schritt bei der Einrichtung eines Gesamtbudgets.		
Append-concat16	Die Definition einer **Haushaltsstruktur** ist der erste Arbeitsschritt im Aufbau eines **Haushalts**.		
Reference	Der Aufbau einer **Haushaltsstruktur** ist der erste Schritt beim Einrichten eines **Haushalts**.		
(3)	inconsistency program error	– –	Inkonsistenz Programmfehler
Source	The table **inconsistency** is probably due to a program error.		
Baseline	Wahrscheinlich liegt ein **Programmfehler** vor.		
Append-concat16	Die Tabellen**inkonsistenz** wird wahrscheinlich durch einen **Programmfehler** verursacht.		
Reference	Die Tabellen**inkonsistenz** ist vermutlich durch einen **Programmfehler** entstanden.		
(4)	processing time	–	Bearbeitungszeit
Source	Field: Goods receipt **processing time**		
Baseline	Feld: Wareneingangs**bearbeitungszeit**		
Append-concat16	Feld: **Bearbeitungszeit** für den Wareneingang		
Reference	Field: WE **Bearbeitungszeit**		
(5)	release order package number	– –	отзыв номер пакета
Source	Purchase order: **release order package number**		
Baseline	Заказ на поставку: деблокировать **номер пакета** заказов		
Append-concat16	Заказ на поставку: **номер пакета отзыва**		
Reference	Заказ на поставку: **номер пакета отзыва**		
(6)	logical port proxy class	– –	логический порт прокси-класс
Source	No **logical port** found for the **proxy class** &1		
Baseline	**Логический порт** для класса прокси &1 не найден		
Append-concat16	**Логический порт** для **прокси-класса** &1 не найден		
Reference	**Логический порт** для **прокси-класса** &1 не найден		
(7)	markup replacement category	– –	надбавка тип замены
Source	To define a relative **markup**, choose the Relative **Markup** value in the **Replacement Category** field.		
Baseline	Для определения относительной наценки выберите значение Относительная наценка в поле **Тип замены**.		
Append-concat16	Чтобы определить относительную **надбавку**, выберите значение Относительная **надбавка** в поле **Тип замены**.		
Reference	Чтобы определить относительную **надбавку**, выберите значение Относительная **надбавка** в поле **Тип замены**.		
(8)	table key	–	ключ таблицы
Source	**Table Key** > 70 Char		
Baseline	Табл. ключ > 70 символов		
Append-concat16	**Ключ таблицы** > 70 символов		
Reference	**Ключ таблицы** > 70 символов		

Table 7: Examples for en–de and en–ru. Terminology constraints are provided above each example. Underlining is used to highlight linguistic aspects described in Section 7.1.

Type of term match	en–de	en–ru
True negative (unmatched)	56%	55%
False negative (matched)	44%	45%

Table 8: Results of analysis of negative term rate samples

It was found that among the analyzed examples there are many false negatives, i.e. the expected term translations were indeed produced. The reason is that our fuzzy term matching strategy on which the term rates are based does not cover them. In the investigated examples, for both languages, around 45% of the terms were not recognized by the term rate for the following reasons:

- The term occurs in an inflected form that escapes the fuzzy match of the term rate (ex. 7).
- The term is part of a compound word that escapes the fuzzy match of the term rate (ex. 3).

When analyzing truly problematic terms, i.e. the true negatives that were not generated in the translation at all, patterns that hint at a reason are harder to detect. Generally, there are three types of behavior: most of the time, the term in question is translated by a synonym, sometimes it is mistranslated, and in rare cases it is dropped. For en–ru, there are a few terms in our test set that were not produced by the NMT model, for example *transaction control* - управление транзакциями. The problem also occurs for en–de but to a lesser extent. All those missed terms are properly annotated in the source text and, as the other terms in the test set, all segments containing these terms were removed from the training data. Without looking at the decoder in detail, we cannot draw any conclusions for now. It is possible that some translations are not enforced since another translation is too “strong”, or the target word does not exist in the training data and is therefore difficult to assemble and produce. We also noticed some problems in compounding, for example an incorrect connecting element on non-head words.

From our analysis we conclude that term enforcement using the *train-by* method does not always work perfectly - but we also know that MT in general does not always work perfectly either. Nevertheless, we have shown that the term rate is higher than what we have reported in Tables 3 and 4. This is due to the large number of false negatives of the term rate caused by the automatic evaluation strategy.

7.3 Considerations for a Production Setting

With the high term rates paired with an improved translation quality and no negative impact on translation speed, the *train-by* method, specifically the *append* variant, offers a good trade-off for terminology enforcement in a production setting, particularly compared to current alternatives in the class of constrained decoding. Whether term rates are high enough for a productive scenario obviously depends on the specific requirements on the MT system and cannot be answered universally.

Note that we did not perform a human analysis of segments without terminology and only interpret the automatic scores. It remains to be seen whether the inline annotation, particularly if used without source factors, is reliable enough to not apply the learned copy mechanism in unsuitable occasions.

Clearly, the results of this approach depend to a high extent on the quality of the term dictionary. Grammatical and lexical ambiguity of terms as well as the quality of translation correspondences are to be considered. Performance and precision of the term recognition mechanism are additional key factors for making this approach work.

8 Conclusion

We have investigated a new approach for terminology integration into NMT, originally proposed by Dinu et al. (2019), in an real-world setup. Our experimental setting was IT-related corporate data from SAP with terminology from SAP’s terminology database, for two language pairs with rather morphologically rich target languages. Our study yields positive results, namely term rates >95% and improvements in translation quality compared to a baseline model as well as constrained decoding, with neither impacting the translation speed nor the translation quality on data without terminology. The improvements in term accuracy were furthermore confirmed in a human evaluation for both language pairs. In an additional manual investigation, we inspected the problematic cases and found that almost half of them are false negatives, meaning that term rates are in fact even higher. We have furthermore confirmed that with this approach the term translations are used flexibly in the surface form required by the sentential context. Overall, it seems to be a promising approach for applying terminology constraints.

References

Burlot, Franck. 2019. Lingua custodia at WMT'19: Attempts to control terminology. In *Proceedings of the Fourth Conference on Machine Translation*, pages 147–154, Florence, Italy. Association for Computational Linguistics.

Chatterjee, Rajen, 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. Association for Computational Linguistics.

Crego, Josep, 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.

Dinu, Georgiana, 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.

Farajian, M. Amin, Nicola Bertoldi, Matteo Negri, Marco Turchi, and Marcello Federico. 2018. Evaluation of terminology translation in instance-based neural MT adaptation. In *Proceedings of the 21st Annual Conference of the European Association for Machine Translation*, pages 149–158, Alacant, Spain.

Hasler, Eva, 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*, pages 506–512, New Orleans, Louisiana. Association for Computational Linguistics.

Hieber, Felix, Tobias Domhan, Michael Denkowski, David Vilar, Artem Sokolov, Ann Clifton, and Matt Post. 2018. The Sockeye Neural Machine Translation toolkit at AMTA 2018. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas*, pages 200–207, Boston, MA. Association for Machine Translation in the Americas.

Hokamp, Chris 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*, pages 1535–1546, Vancouver, Canada. Association for Computational Linguistics.

Hu, J. Edward, Huda Khayrallah, Ryan Culkin, Patrick Xia, Tongfei Chen, Matt Post, and Benjamin Van Durme. 2019. Improved lexically constrained decoding for translation and monolingual rewriting. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 839–850, Minneapolis, Minnesota. Association for Computational Linguistics.

Huck, Matthias, Viktor Hangya, and Alexander Fraser. 2019. Better OOV translation with bilingual terminology mining. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5809–5815, Florence, Italy. Association for Computational Linguistics.

Kobus, Catherine, Josep Crego, and Jean Senellart. 2017. Domain control for neural machine translation. In *Proceedings of the International Conference Recent Advances in Natural Language Processing*, pages 372–378, Varna, Bulgaria. INCOMA Ltd.

Papineni, Kishore, 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. Association for Computational Linguistics.

Popović, Maja. 2015. chrF: character n-gram f-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Post, Matt 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*, pages 1314–1324, New Orleans, Louisiana. Association for Computational Linguistics.

Sennrich, Rico and Barry Haddow. 2016. Linguistic input features improve neural machine translation. In *Proceedings of the First Conference on Machine Translation*, pages 83–91, Berlin, Germany. Association for Computational Linguistics.

Sennrich, Rico, 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*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In Guyon, I., 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.