Adding soft terminology constraints to pre-trained generic MT models by means of continued training

Tommi Nieminen
University of Helsinki, Finland
tommi.nieminen@helsinki.fi

Abstract

This article describes an efficient method of adding terminology support to existing machine translation models. The training of the pre-trained models is continued with parallel data where strings identified as terms in the source language data have been annotated with the lemmas of the corresponding target terms. Evaluation using standard test sets and methods confirms that continued training from generic base models can produce term models that are competitive with models specifically trained as term models.

1 Introduction

One of the major challenges of using machine translation (MT) to enhance the productivity of human translators in professional translation is enforcing the use of correct terminology in MT output. In general, a translator is expected to adhere either to standard domain-specific terminology, or to a client-specific terminology, which can be provided as a dedicated terminology database (usually referred to as a *termbase*) or implicitly in the form of a translation memory. In the professional translation setting, when the output of a MT system diverges from the specified terminology, a translator needs to correct the output manually, significantly reducing the utility of MT. It is therefore important that a translator has the capability of influencing the terminological choices that the MT system makes by providing terminology to the system.

In this article, we introduce a method of adding support for enforcing user-provided terminology into existing MT models. The method is based on continued training of the model using data annotated with terminology information.

2 Related work

2.1 Constraining terminology in neural machine translation

The majority of methods of constraining an neural machine translation (NMT) model to use user-provided terminology in translations belong to four distinct categories.

Pass-through placeholders

Source terms in the source sentence are replaced by placeholders, and the NMT model reproduces the placeholders in the translation (Michon et al., 2020). The reproduced placeholders in the translation are then replaced by the target terms corresponding to the source terms that the placeholder had originally replaced. This approach requires that the model is trained with data that has been augmented with sentence pairs containing aligned placeholders on source and target sides. Using pass-through placeholders usually ensures that the target terms are generated in correct positions, but the information contained in the source term is lost and cannot be utilized by the model when generating the translation, which can lead to translation errors. It is also difficult to generate the correct morphological features for the target terms, especially for morphologically complex languages.

Constrained decoding

In constrained decoding, the search algorithm of the MT system is modified to ensure that target terms are generated for each source term identified in the source sentence. For instance, Hokamp and Liu (2017) introduce a variant of beam search

Proceedings of the First International Workshop on Knowledge-Enhanced Machine Translation, pages 21–33
June 27, 2024 ©2024 European Association for Machine Translation

called Grid Beam Search, which only produces hypotheses that contain the required target terms. The benefit of constrained decoding is that it can be used as an add-on component to any MT model. However, most constrained decoding implementations are much slower than normal beam search, and they may cause translation quality issues, as the required target terms will be used even in inappropriate contexts.

Adding target terms as soft constraints

The pass-through placeholder and constrained decoding methods treat terms as unconditional: they should always be included in the generated translation. In those methods, terms can therefore be referred to as hard constraints on the MT output.

It is also possible to add terms as soft constraints, which the MT model can override. The most common method of implementing terminology as soft constraints is to annotate the source data with terminology information. These annotations can be added in different ways. For instance, in the first published work on soft terminology constraints (Dinu et al., 2019), two methods were tested: the target term was either appended after the corresponding source term or the target term replaced the source term. Factors were used to signal that the target terms were to be processed differently from normal source tokens. Like the pass-through placeholder method, the soft constraint method requires that training data of the model is augmented, in this case with sentence pairs, where the source sentence has been annotated with target term information that also occurs in the target sentence. This causes the model to associate a target term in the source sentence with having the same target term in the target sentence.

The annotation-based soft constraint method seems to currently be the most popular and widely used method of enforcing user-provided terminology, and it has also been most successful and common in recent terminology MT shared tasks (Alam et al., 2021b; Semenov et al., 2023).

Using large language models

Large language models (LLMs) provide another way to apply terminology as soft constraints. With LLMs, the use of user-provided terminology can be enforced in several ways. Moslem et al. (2023a) implement constrained terminology in LLM translation by adding terminology translations to the prompts they use to elicit translations from the `GPT-3.5 text-davinci-003` model. Bogoychev and Chen (2023) use the `gpt-3.5-turbo-0613` model to correct terminology in an unconstrained LLM translation by providing a refined prompt containing the required terminology changes. LLMs can also be used to post-edit the terminology in translations generated by conventional NMT systems (Moslem et al., 2023b).

2.2 Continued training

In continued training (also called fine-tuning), the training of a pre-trained NMT model is continued with a training set that is usually either a distinct subset of the original training data of the pre-trained model or a new data set which was not included in the original training data, at least not in its entirety. The most common use case for continued training is domain adaptation, for instance adapting a pre-trained generic NMT model to speech translation using speech data (Luong and Manning, 2015). Continued training has also been used for adding new language pairs to a multilingual NMT model (Neubig and Hu, 2018), and to alleviate the effects of large amounts of back-translated data on translation quality by continuing training with only genuine parallel data (Bawden et al., 2019).

Continued training is widely used in NMT research and industry, and its effectiveness has been demonstrated with manual evaluation (Dogru and Moorkens, 2024). However, continued training always entails a risk of catastrophic forgetting (Mccloskey and Cohen, 1989), where the model partially or completely loses the ability to translate source text that is not present in the training set used for the continued training.

3 Model training

We generate models with terminology support (term models) for multiple language pairs by continuing the training of generic base models with data annotated with terminology information.

Pre-trained models published as part of the Tatoeba-Challenge (Tiedemann, 2020)[1] project are used as the base models for the continued training. Tatoeba-Challenge project includes MT models for hundreds of language pairs, and for many language pairs there are multiple models available. According to automatic evaluations performed on

[1] https://github.com/Helsinki-NLP/Tatoeba-Challenge

the Tatoeba-Challenge models, the models with the best performance are based on the *transformer-big* architecture. However, as we intend to deploy these terminology models for low-latency CPU inference on desktop computers, we use the *transformer-base* models, which generate translations much quicker.

	Base model
en-bg	opus+bt-2021-04-13
en-da	opus+bt-2021-04-13
en-de	opus+bt-2021-04-13
en-es	opus+bt-2021-04-10
en-et	opus-2019-12-18
en-fi	opusTCv20210807+bt-2021-09-01
en-fr	opus-2021-02-22
en-it	opus+bt-2021-04-14
en-lt	opus+bt-2021-04-14
en-nl	opus+bt-2021-04-14
en-sv	opus+bt-2021-04-14
fi-en	opusTCv20210807+bt-2021-08-25

Table 1: Models that were used as base models for term fine-tuning (all are different bilingual models).

For the experiments, we selected a subset of language pairs for which base models of reasonable quality (according to the published automatic metrics[2]) were available. The selection includes 12 medium- and high-resource translation directions between different pairs of European languages. For some language pairs, such as English to Estonian, *transformer-base* models are not available among Tatoeba-Challenge models, and models from the OPUS-MT model collection (Tiedemann and Thottingal, 2020) are used instead. All models have been trained on data that has been segmented with `SentencePiece` (Kudo and Richardson, 2018) (see table 1 for the model names).

The continued training is performed with `MarianNMT` (Junczys-Dowmunt et al., 2018) using the default settings (v1.11.13). While adjusting hyperparameters, such as learning rate, might make the continued training more efficient, the initial automatic metric results indicated that the default settings were sufficient for the task, so we decided not to experiment with any hyperparameter adjustments. The duration of continued training was one epoch.

[2]https://opus.nlpl.eu/dashboard/

3.1 Data

The training of each model is continued with a subset of the Tatoeba-Challenge data set *v2023-09-26* for the language pair in question. Tatoeba-Challenge data sets contain most of the data available in the OPUS corpus collection (Tiedemann, 2009). The base models were originally trained with an earlier version of the Tatoeba-Challenge data set, so the original training data and the data for continued training overlap significantly. Since the data sets contain large proportions of crawled data, which often has quality issues (Kreutzer et al., 2022), Bicleaner-AI (Zaragoza-Bernabeu et al., 2022) scores (model version 2.0) are used to extract the best quality parallel sentence pairs to be used as the continued training set. Ten million best-scoring sentence pairs are extracted as fine-tuning data for each language pair.

3.2 Training pipeline

A modified version of Mozilla's *firefox-translations-training*[3] pipeline was used to train the models. This pipeline, which is based on the Snakemake workflow management system (Mölder et al., 2021), can perform all the steps required for building NMT models, such as loading, pre-processing, cleaning and filtering the training data, and training and evaluating the NMT models. For the work described in this article, terminology annotation and evaluation components were added to the pipeline. The code for the modified pipeline is available on GitHub.[4]

3.3 Terminology annotation

As mentioned, training data for soft constraint terminology training needs to be annotated with terminology information. Two different methods are commonly used for generating such annotated training data:

1. **Annotating a corpus using a termbase** (e.g. Dinu et al. (2019)): Given a termbase, such as IATE,[5] and a parallel corpus, search the parallel corpus for sentence pairs where the source sentence contains source terms from the termbase. For those sentence pairs with source terms, check for each source term whether the corresponding target term also

[3]https://github.com/mozilla/firefox-translations-training
[4]https://github.com/Helsinki-NLP/OpusDistillery/tree/eamt_opuscat_terms
[5]https://iate.europa.eu/home

occurs in the target sentence. Then annotate those source terms that have corresponding target terms with terminology information.

2. **Annotating a corpus using aligned pseudo-terms** (e.g. Bergmanis and Pinnis (2021)): Given a parallel corpus, align source and target tokens with an alignment tool such as `FastAlign` (Dyer et al., 2013). Then select aligned subsets of tokens and use them as pseudo-terms.

There are benefits and disadvantages to both of these methods: using a termbase ensures that the annotated terms are reasonable, but it also restricts the annotations to the subject matter of the database making them uniform, and unless the database is very large, there may not be enough term matches found in the parallel data to generate an annotated training set that is large enough. On the other hand, pseudo-terms may not bear much resemblance to actual terminology, unless their generation is restricted in some way. One major benefit of the pseudo-term approach is that it is language-independent, while the database approach is only usable for language pairs for which suitable termbases exist.

We use the aligned pseudo-term approach for reasons of simplicity. The pseudo-term generation is restricted to aligned noun and verb phrase chunks, as real-world terminology generally also consists of noun and verb phrases. The process for generating the annotated training data from parallel data is the following:

1. **Parse data to identify POS and dependencies:** Generate the parts-of-speech (POS) and dependency relations of source and target sentence tokens using `Stanza` (Qi et al., 2020).

2. **Create noun and verb chunks:** Identify noun and verb phrase chunks in the source and target sentences based on the POS and dependency information.

3. **Token alignment:** Align parallel corpus on token-level with FastAlign, using the *grow-diag-final-and* heuristic.

4. **Chunk alignment:** Use alignment information from step 3 to identify source noun chunks that are aligned to target noun chunks and source verb chunks that are aligned to target verb chunks.

5. **Appending target chunk lemmas to source chunks:** Append lemma forms of target chunks after the corresponding source chunks in the source sentence.

Our pseudo-term generation method is very similar to that used in (Bergmanis and Pinnis, 2021). The difference is that we align chunks instead of words, and the alignment is performed on the subword units of the sentences instead of the lemma forms of words in the sentence.

The parallel data is annotated with the pseudo-terms by appending the target term after the corresponding source term in the source sentence. The term annotation is indicated by using three indicator tags: one before the source term, one between the source term and the target term, and one after the target term. See table 2 for an example of the annotation scheme.

The annotation scheme is similar to the *append* method used in (Dinu et al., 2019). The main difference is that like Ailem et al. (2021) we use tags and not factors to indicate target terms. Lemma forms of target terms are used in the source sentence in order to make the model associate a lemma form in the source sentence with an inflected form in the target sentence, which is the behaviour that the model should ideally adapt during the training.

Any number of terms can occur in a source sentence, so the training data needs to contain source sentences with varying amounts of annotated terms. Our annotation script keeps a running count of the number of sentences with n terms that have been annotated, and ensures that there is variability in the amounts of terms in the training data sentence pairs. The amount of sentence pairs per term count approximates a geometric series, where the amount of sentence pairs is halved for each term. The ratio is chosen on the assumption that only a few terms will occur in most sentences, although in actual production cases the frequency of terms will probably vary greatly by domain and the scale and level of detail of the terminology database that is used.

For some sentence pairs in the training corpus, no aligned term chunks are found using the above method, so for each language pair there is a varying amount of sentence pairs without term annota-

Source	`British Library releases a million images on Flickr`
Annotated source	`British Library <term_start> releases <term_end> veröffentlichen <trans_end> a million <term_start> images </term_end> Bild <trans_end> on Flickr`

Table 2: Example of the annotation method scheme used in the experiments (note in the actual training data the sentences are split into sub-word units, here they appear unsegmented for clarity)

tions. To see the effect of having a mix of annotated and unannotated sentence pairs in the training corpus, two models are trained for each language pair: one with both unannotated and annotated sentence pairs, and one with only annotated sentence pairs (referred to as the *only-terms* model in the tables). See table 3 for amounts of sentences annotated with terms for each language pair.

	Annotated sentences
en-bg	7,604,181
en-da	7,441,517
en-de	6,092,623
en-es	5,782,967
en-et	7,226,641
en-fi	6,706,819
en-fr	4,599,385
en-it	3,143,592
en-lt	7,495,889
en-nl	7,358,655
en-sv	7,330,407
fi-en	6,510,906

Table 3: The amounts of sentences annotated with terms for each language pair. Annotated sentences contain 1.99 terms on average. *only-terms* models are trained with this data only, while *term* models are trained with the whole 10 million sentence pair training set, including sentence pairs without terms.

3.4 Vocabulary adaptation

The vocabularies of the base Tatoeba-Challenge models contain only symbols that have occurred in the original training corpus, i.e. the Tatoeba-Challenge data set segmented with SentencePiece. There are no spare symbols that can be used as terminology tags, so naturally occurring symbols have to be repurposed to act as the terminology tags. We use an automatic method to choose three uncommon vocabulary units to act as the terminology tags. As the symbols chosen as the terminology tags do not occur in the filtered training data (they are extremely rarely occurring tokens, such as characters from non-Latin scripts), re-purposing them should have no effect on translation quality.

4 Evaluation

There are three important aspects to the evaluation of NMT models with terminology support:

1. **Overall translation quality without terminology**: how well the model translates source sentences with no terms present.
2. **Terminological accuracy**: how many of the source terms have a corresponding target term present in the translation.
3. **Overall translation quality with terminology**: if the source sentence is annotated with terms, how well does the model translate the sentence (regardless of how many terms it gets correct).

Ideally, a terminology model translates terms accurately, while maintaining an overall translation quality level comparable to the base model, both when translating sentences with terms or without them. This kind of model can be used independently, with no supplementary models.

Minimally, a terminology translation model has to have a reasonable level of term accuracy without causing the overall translation quality of sentences with terminology to degrade too much. A term model with this kind of minimal performance can still be useful, as long as it is used together with a generic back-off model that translates sentences without terms.

4.1 Overall translation performance without terminology

The purpose of evaluating translation performance without terminology is to see if catastrophic forgetting occurs, i.e whether the continued training significantly degrades the term model's performance in general translation.

Metrics

Terminology models are compared against the base models using two automatic evaluation metrics. BLEU scores are generated using `sacrebleu` (Post, 2018), and additionally

COMET (Rei et al., 2020) scores are generated with the `wmt22-comet-da` model.

Data for evaluating translation performance without terms

For each language pair, a maximum of four test sets are downloaded using `sacrebleu` and `mtdata` (Gowda et al., 2021) tools. For most language pairs, WMT test sets from different years are used. If no WMT test sets are available for a language pair (such as English to Swedish), the FLORES test set (Goyal et al., 2021) is used instead. The test sets were compared with the fine-tuning sets to verify that there was no overlap that could affect the results.

The results of evaluation without terminology are listed in table 4.

4.2 Terminological accuracy

Term models are assessed on how well they reproduce the specified terminology in their outputs. The evaluation is primarily performed with the methods outlined in (Alam et al., 2021a), using the `terminology_evaluation` [6] script provided by the authors. As the script assumes tokenized and truecased input, we use a modified script that tokenizes and truecases the SentencePiece output from the models using Stanza. Due to this and other changes, the modified script is made separately available. [7]

The main evaluation metric included in the script is *Exact-Match Accuracy*, which scores a translation based on how many of the required target terms it contains. Despite the name, the metric also accepts inflected forms of the target terms in addition to exact matches.

The principal difficulty in judging the terminological correctness of a translation is that while it is simple to check if a translation contains the lemma or inflected forms of required target terms, it is not easy to check whether the target term has the correct form or that it is placed grammatically in the translation. If terminological correctness is evaluated solely by counting the occurrence of target terms in any inflection form, the evaluation becomes very easy to cheat in (purposefully or by accident): the model simply needs to add the terms in any position in the translation. This cheating problem particularly affects hard terminology constraint methods, i.e. constrained decoding and pass-through placeholders, since they will always produce the target terms, but soft constraint models are not immune to it either.

(Alam et al., 2021a) proposes multiple solutions to the cheating problem:

1. Window overlap: When a target term occurs in a translation, extract *n* content words surrounding the target term and check how many of those content words also occur in the *n* content words surrounding the same target term in a reference translation. This will reward terms that are placed similarly to the corresponding term in a reference translation.

2. Terminology-biased TER (TERm): A modified TER metric, where the edit cost is doubled for any reference word belonging to a target term.

It should be noted that both of these metrics rely on reference translations, so they are are affected by the same problem as all reference-based metrics: the single reference translation available represents only one of many possible valid translations, and many valid translations are therefore scored incorrectly. However, combined with Exact-Match Accuracy, these metrics can provide some extra information about the term accuracy of MT models.

Data for term accuracy evaluation

Evaluating term accuracy requires minimally a terminology and a collection of source language sentences which contain terms present in the terminology. This type of data is easy to obtain in theory, since monolingual data is plentiful, and there are many freely available and extensive terminology databases, such as IATE. However, test data created in this manner is artificial and may not reflect actual use cases of terminology, unless the data is carefully prepared and reviewed. Because of this, we use publicly available terminology test sets for evaluation. We found three potentially suitable test sets:

1. Annotated Tico-19 test set published for the WMT21 term task (Alam et al., 2021b).[8]

2. Test set for a case study on terminology translation for the Canadian Parliament (Knowles et al., 2023). [9]

[6] https://github.com/mahfuzibnalam/terminology_evaluation
[7] https://github.com/TommiNieminen/soft-term-constraints
[8] https://www.statmt.org/wmt21/terminology-task.html
[9] https://github.com/nrc-cnrc/PFT-ef-EAMT23

	Test sets	Base model	Term model	Only-terms model	Change: base to term
en-bg	FLORES	41.64 / 0.866	42.91 / 0.875	43.33 / 0.877	1.27 / 0.009
en-da	FLORES	45.85 / 0.865	46.71 / 0.866	47.10 / 0.865	0.86 / 0.001
en-de	WMT17,18,19+FLORES	40.55 / 0.787	40.65 / 0.787	41.60 / 0.788	0.1 / 0
en-es	WMT11,12,13+FLORES	38.20 / 0.816	37.70 / 0.814	38.06 / 0.817	-0.5 / -0.002
en-et	WMT18+FLORES	23.71 / 0.824	25.55 / 0.848	25.56 / 0.849	1.84 / 0.024
en-fi	WMT17,18,19+FLORES	26.63 / 0.862	25.91 / 0.866	25.94 / 0.866	-0.72 / 0.004
en-fr	WMT11,12,13+FLORES	35.98 / 0.798	33.68 / 0.795	34.94 / 0.805	-2.3 / -0.003
en-it	WMT09+FLORES	33.11 / 0.816	32.84 / 0.817	34.38 / 0.825	-0.27 / 0.001
en-nl	FLORES	26.49 / 0.824	27.76 / 0.826	27.11 / 0.825	1.27 / 0.002
en-lt	WMT19+FLORES	20.63 / 0.782	22.84 / 0.814	22.68 / 0.813	2.21 / 0.032
en-sv	FLORES	44.29 / 0.868	45.43 / 0.867	45.56 / 0.865	1.14 / -0.001
fi-en	WMT17,18,19+FLORES	31.70 / 0.849	30.82 / 0.845	30.96 / 0.846	-0.88 / -0.004

Table 4: General translation performance measured as BLEU/COMET. Note that the input to the term models was not annotated with terms when translating these test sets, they translated the same unannotated input as the base model. Therefore it would be expected that the term models would perform worse in this evaluation due to being further trained for another task.

3. Automotive Test Suite, an automotive corpus annotated with terms (Bergmanis and Pinnis, 2021). [10]

Out of these three, only the Tico-19 set includes term annotations on the target side, which are required by the `terminology_evaluation` script (the Tico-19 test set uses the exact formatting that the script expects, as they were both used in the WMT21 terminology shared task). The terminology in the Canadian Parliament test set appears to be fairly generic and sparse in terminology, so we decided not the use it (especially since the English to French language pair is already covered by Tico-19). For the Automotive Test Suite, we only evaluated term accuracy, using the same script as in (Bergmanis and Pinnis, 2021) in order to produce comparable results.

We do not include results for the *only-terms* model for these test sets, as all other results point to there being very little difference in performance between the *term* and *only-terms* models.

The test sets are primarily used to compare base model and term model performance to see if any improvement in term translation occurs. Although we include the results from the articles connected to these sets in our result tables (tables 5 and 6) for reference, they are not directly comparable to the results obtained with our models. First of all, the base models we use have been trained on a larger parallel corpus, which affects the COMET and BLEU metrics and may also affect the term accuracy score. Secondly, even though we use the same scripts as in the referred articles for evaluation, there may be subtle differences due to post-publication changes to the scripts.

[10]https://github.com/tilde-nlp/terminology_translation

Artificial test sets

The available test sets are relatively small and cover only a few of the language pairs for which we have trained models for, so we additionally test the models on artificial test sets which have been generated with the same method as the annotated training set. These test sets are created by concatenating the normal test sets for a language pair, annotating the concatenated file with pseudo-terms, and then generating source and target files in the .sgm format required by the `terminology_evaluation` script. One limitation of the artificial test sets is that the pseudo-terms tend to be common words and phrases, which often have only one suitable translation in the context. This means they probably overestimate the term accuracy of the base models. The results of the artificial test set evaluation are listed in table 7.

Discussion of automatic evaluation results

Automatic evaluation with both the previously published test sets and the artificial test sets clearly indicate that the continued training with terminology annotations increases terminology accuracy significantly, without degradation in overall translation quality, whether or not the source sentence contains terms. Term models consistently have

	Exact Match Accuracy	Window Overlap 1	Window 2 Overlap 2	TERm	BLEU	COMET
base	0.838	0.253	0.264	0.609	46.80	0.802
term	0.931	0.245	0.257	0.582	42.54	0.806
best in WMT21	0.974	0.359	0.352	0.625	47.69	

Table 5: Evaluation results for the Tico-19 test set from WMT21 shared terminology task (EN-FR only). Note that the best WMT21 model scores are not directly comparable due to possible differences in evaluation setup (WMT21 COMET score is omitted completely, as it is based on a different COMET model).

	Base model	Term model	TLA
en-de	29.5 / 47.6	33.2 / 95.1	33.5 / 94.0
en-et	19.8 / 40.2	22.6 / 82.5	21.0 / 87.2
en-lt	17.9 / 38.8	20.3 / 59.9	30.1 / 90.3

Table 6: BLEU scores and terminology accuracy scores for the Automotive Test Suite. TLA (Target Lemma Annotations) refers to results from Bergmanis and Pinnis (2021).

better term accuracy than base models, and term accuracy is usually very high (over 0.95 for all term models with the artificial test sets). The improvement of the term model Window Overlap scores compared to the base model scores also indicates that the placement of the terms in the output is reasonable.

One exception to the high term accuracy is the EN-LT term model, where term accuracy is fairly low with the ATS test set. This may be due to the low quality of the base model for EN-LT, which is reflected in the large disparity between the BLEU score (17.9 vs 30.1) of the base model and the model used by Bergmanis and Pinnis (2021).

In general, the term models perform better with the artificial test sets than with the Tico-19 and ATS test sets. This is probably due to the large amount of generic terms in the artificial test sets, which are easy for the model to get right. However, the term model performance still remains at a reasonably high level, and is considerably better than base model performance.

The evaluation of translation performance without terms indicates that no catastrophic forgetting takes place during the continued training. With most language pairs, the continued training even increases the BLEU score, although the COMET scores remain similar. This may be partly due to the fact that the training set for the continued training has been filtered with Bicleaner-AI, and should be of higher quality than the rest of the Tatoeba-Challenge data.

4.3 Manual evaluation

Since automatic evaluation cannot conclusively judge whether the term models improve terminology translation without degrading general translation quality, we conducted a short manual evaluation to determine the effect more reliably. The manual evaluation is conducted with the English to Finnish language direction. Finnish is a morphologically complex language, so problems in the grammaticality of the terms should be more apparent than with morphologically simpler target languages. The evaluator is an experienced professional English-to-Finnish translator, who is a native Finnish speaker.

51 sentence pairs were selected for manual evaluation from the artificial term test set. As mentioned, the artificial term test set contains a large amount of cases where the terms are obvious, i.e. there are only few realistic term translations, and therefore any decent model will likely translate the term according to the terminology. To extract interesting test cases, the evaluation set was picked from those sentences where the base model translation did not contain the required terms. These are more likely to be sentences for which the base model would struggle to produce correct terminology. From this set, 51 sentences for which the term model had produced a terminologically correct translation were randomly selected as the final manual evaluation set.

In the first phase of the manual evaluation, the evaluator was presented with the source sentences one by one, along with the base model and term model translations for each sentence in random order. The reviewer was instructed to select from three options for each pair of translations A and B: 1. translation A is better, 2. translation B is better or 3. translations A and B are equally good. The purpose of this phase was to determine whether the term model translations are noticeably inferior to the base model translations. Note that in this

	Model	Exact Match Accuracy	Window Overlap 1	Window 2 Overlap 2	TERm	BLEU	COMET
en-et	base	0.739	0.272	0.292	0.402	23.71	0.824
	only-terms	0.962	0.334	0.362	0.460	27.49	0.854
	term	0.964	0.337	0.364	0.457	27.71	0.855
en-nl	base	0.715	0.366	0.369	0.412	26.49	0.824
	only-terms	0.966	0.437	0.445	0.448	29.58	0.829
	term	0.970	0.439	0.448	0.452	29.83	0.831
en-fi	base	0.731	0.296	0.309	0.407	26.63	0.862
	only-terms	0.964	0.354	0.374	0.454	29.45	0.873
	term	0.967	0.356	0.376	0.454	29.59	0.874
en-sv	base	0.750	0.464	0.478	0.604	44.29	0.868
	only-terms	0.983	0.539	0.559	0.650	48.68	0.873
	term	0.980	0.537	0.556	0.655	48.78	0.874
en-bg	base	0.772	0.374	0.407	0.571	41.64	0.866
	only-terms	0.959	0.443	0.482	0.607	45.41	0.881
	term	0.965	0.442	0.481	0.609	45.39	0.879
en-es	base	0.750	0.364	0.388	0.512	38.20	0.816
	only-terms	0.975	0.421	0.451	0.553	40.85	0.825
	term	0.979	0.419	0.450	0.553	40.79	0.824
en-da	base	0.775	0.428	0.459	0.620	45.85	0.865
	only-terms	0.986	0.499	0.532	0.658	49.72	0.872
	term	0.987	0.495	0.531	0.656	49.57	0.871
fi-en	base	0.697	0.311	0.342	0.476	31.70	0.849
	only-terms	0.982	0.387	0.424	0.527	34.96	0.856
	term	0.982	0.386	0.424	0.528	34.85	0.855
en-fr	base	0.735	0.323	0.352	0.481	35.98	0.798
	only-terms	0.974	0.376	0.412	0.525	37.80	0.816
	term	0.978	0.375	0.410	0.524	37.57	0.815
en-it	base	0.763	0.350	0.367	0.463	33.11	0.816
	only-terms	0.960	0.410	0.440	0.520	37.38	0.834
	term	0.967	0.415	0.442	0.523	37.42	0.836
en-lt	base	0.708	0.212	0.236	0.333	20.63	0.782
	only-terms	0.961	0.277	0.308	0.386	25.06	0.821
	term	0.967	0.280	0.307	0.386	24.96	0.821
en-de	base	0.733	0.367	0.399	0.540	40.55	0.787
	only-terms	0.985	0.442	0.481	0.603	45.46	0.802
	term	0.986	0.440	0.479	0.601	45.24	0.802

Table 7: Term translation performance measured with the `terminology_evaluation` script using artificial term test sets. Pseudo-terms have been annotated in the term model input, but not in the base model input. Note that since the annotated terms occur in the reference translation, BLEU and COMET scores favour the term models. Test sets are the same as in Table 2.

phase the translator was not given details of the terms used in generating the translation, and they only ranked the sentences based on overall quality according to the normal translation industry standards. In this phase, the reviewer was also not yet informed that the evaluation concerned terminology.

Since the term model had access to terms that had been used in at least one acceptable translation (the reference translation based on which the pseudo-terms were generated), it would be expected to perform better than the base model in the first phase. Again, the purpose of this phase was not to compare the base and term model translations on even ground, but to determine whether noticeable quality degradation takes place with the term model.

In the second phase of the manual evaluation,

Source	The students gathered on the pier.
Terms	the student = uimakoululainen, pier = laituri
Target	Uimakoululaiset kokoontuivat laiturille.

Table 8: Example of the term model inflecting lemma forms of terms. The term model clearly utilizes the term information, as the Finnish translation of *the student* here means a student of a swimming school, and would be a very unlikely translation without the term information.

the evaluator was instructed to judge whether the term translations in the output of the term model were syntactically and/or semantically correct. The purpose of this phase was to determine whether the term placement in the term model output is reasonable, i.e. that the model is not cheating the automatic evaluation metric by placing the term in an incorrect place and/or in an incorrect morphological form. For each source sentence in the evaluation set, the reviewer was presented with the term model output and a list of terms that were expected to be in the output, in addition to the source sentence. For each translation, the reviewer recorded the number of terms which had been correctly used in the translation.

4.4 Results of manual evaluation

The results of the manual evaluation clearly indicate that the term model performs well, even if the target language is morphologically complex. In the first phase, the term model was ranked as performing better than base model in 20 cases, while the base model was judged to be better than the term model in 11 cases. In 20 cases, the model outputs were judged to be of equal quality. The results of the second phase also indicate that the term model performs well, with the reviewer judging 171 out of 178 terms as being correctly used. Since the morphological forms of the terms present in the output are very varied, it is clear that the model is capable of inflecting the lemma forms of the terms. Table 8 shows one example of the term model correctly inflecting several terms.

5 Energy use considerations

Training of NMT models consumes considerable amounts of energy. Strubell et al. (2019) estimate that training a *transformer-base* model of the type used in our experiments consumes 27 kWh of energy. Since we do not train from scratch but use continued training, the energy consumption of actual model training is considerably lower than the 27 kWh baseline. Unfortunately, we could not track the exact energy consumption of the experiments due to the nature of the computing infrastructure that was used (shared dual GPU in a supercomputer, where energy measurement data of the GPU includes the data for other jobs running on the same dual GPU). Based on the partial energy consumption data that we have recorded and the running times on jobs, we estimate that the continued training consumed approximately 0.35 kWh per model.

While the energy consumption of the continued training is low, using Stanza to annotate the training corpus with terminology information consumes significant amounts of energy. We estimate that the terminology annotation consumes around 5 kWh per language pair. The energy use could be minimized by switching to a less resource-intensive parser (such as `spaCy`[11]).

While the energy consumption of Bicleaner-AI is also significant, it is not included here, since we used publicly available pre-existing Bicleaner-AI scores from the Tatoeba-Challenge project.[12]

Based on a survey by Donnellan et al. (2023), the estimates above have been multiplied with a Power Usage Effectiveness (PUE) value of 1.58.

6 Conclusions

The experiments described in this article demonstrate that continued training can be used to add soft terminology constraints to pre-trained generic MT models. Automatic and manual evaluation of the model outputs clearly indicate that high levels of terminology accuracy can be achieved at a fraction of the energy cost of training a new model from scratch.

7 Acknowledgments

This work is part of the GreenNLP project, funded by the Finnish Research Council (funding agreement 353166).

References

Ailem, Melissa, Jingshu Liu, and Raheel Qader. 2021. Encouraging neural machine translation to satisfy

[11] https://spacy.io/

[12] https://github.com/Helsinki-NLP/Tatoeba-Challenge/blob/master/BicleanerScores.md

terminology constraints. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1450–1455, Online, August. Association for Computational Linguistics.

Alam, Md Mahfuz Ibn, Antonios Anastasopoulos, Laurent Besacier, James Cross, Matthias Gallé, Philipp Koehn, and Vassilina Nikoulina. 2021a. On the evaluation of machine translation for terminology consistency. *CoRR*, abs/2106.11891.

Alam, Md Mahfuz Ibn, Ivana Kvapilíková, Antonios Anastasopoulos, Laurent Besacier, Georgiana Dinu, Marcello Federico, Matthias Gallé, Kweonwoo Jung, Philipp Koehn, and Vassilina Nikoulina. 2021b. Findings of the wmt shared task on machine translation using terminologies. In *Proceedings of the Sixth Conference on Machine Translation*, pages 652–663, Online, November. Association for Computational Linguistics.

Bawden, Rachel, Nikolay Bogoychev, Ulrich Germann, Roman Grundkiewicz, Faheem Kirefu, Antonio Valerio Miceli Barone, and Alexandra Birch. 2019. The University of Edinburgh's submissions to the WMT19 news translation task. In Bojar, Ondřej, Rajen Chatterjee, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, André Martins, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana Neves, Matt Post, Marco Turchi, and Karin Verspoor, editors, *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 103–115, Florence, Italy, August. Association for Computational Linguistics.

Bergmanis, Toms 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, April. Association for Computational Linguistics.

Bogoychev, Nikolay and Pinzhen Chen. 2023. Terminology-aware translation with constrained decoding and large language model prompting. In Koehn, Philipp, Barry Haddow, Tom Kocmi, and Christof Monz, editors, *Proceedings of the Eighth Conference on Machine Translation*, pages 890–896, Singapore, December. Association for Computational Linguistics.

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, July. Association for Computational Linguistics.

Dogru, Gokhan and Joss Moorkens. 2024. Data augmentation with translation memories for desktop machine translation fine-tuning in 3 language pairs. *The Journal of Specialised Translation*, (41):149–178, Jan.

Donnellan, Douglas, Daniel Bizo, Jacqueline Davis, Andy Lawrence, Owen Rogers, Lenny Simon, and Max Smolaks. 2023. Uptime Institute's Global Data Center Survey Results 2023. Technical report, Uptime Institute.

Dyer, Chris, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In Vanderwende, Lucy, Hal Daumé III, and Katrin Kirchhoff, editors, *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*, pages 644–648. The Association for Computational Linguistics.

Gowda, Thamme, Zhao Zhang, Chris Mattmann, and Jonathan May. 2021. Many-to-English machine translation tools, data, and pretrained models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 306–316, Online, August. Association for Computational Linguistics.

Goyal, Naman, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2021. The flores-101 evaluation benchmark for low-resource and multilingual machine translation.

Hokamp, Chris and Qun Liu. 2017. Lexically constrained decoding for sequence generation using grid beam search. In *Annual Meeting of the Association for Computational Linguistics*.

Junczys-Dowmunt, Marcin, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia, July. Association for Computational Linguistics.

Knowles, Rebecca, Samuel Larkin, Marc Tessier, and Michel Simard. 2023. Terminology in neural machine translation: A case study of the Canadian Hansard. In *Proceedings of the 24th Annual Conference of the European Association for Machine Translation*, pages 481–488, Tampere, Finland, June. European Association for Machine Translation.

Kreutzer, Julia, Isaac Caswell, Lisa Wang, Ahsan Wahab, Daan van Esch, Nasanbayar Ulzii-Orshikh, Allahsera Tapo, Nishant Subramani, Artem Sokolov, Claytone Sikasote, Monang Setyawan, Supheakmungkol Sarin, Sokhar Samb, Benoît Sagot, Clara Rivera, Annette Rios, Isabel Papadimitriou, Salomey Osei, Pedro Ortiz Suarez, Iroro Orife, Kelechi

Ogueji, Andre Niyongabo Rubungo, Toan Q. Nguyen, Mathias Müller, André Müller, Shamsuddeen Hassan Muhammad, Nanda Muhammad, Ayanda Mnyakeni, Jamshidbek Mirzakhalov, Tapiwanashe Matangira, Colin Leong, Nze Lawson, Sneha Kudugunta, Yacine Jernite, Mathias Jenny, Orhan Firat, Bonaventure F. P. Dossou, Sakhile Dlamini, Nisansa de Silva, Sakine Çabuk Ballı, Stella Biderman, Alessia Battisti, Ahmed Baruwa, Ankur Bapna, Pallavi Baljekar, Israel Abebe Azime, Ayodele Awokoya, Duygu Ataman, Orevaoghene Ahia, Oghenefego Ahia, Sweta Agrawal, and Mofetoluwa Adeyemi. 2022. Quality at a glance: An audit of web-crawled multilingual datasets. *Transactions of the Association for Computational Linguistics*, 10:50–72.

Kudo, Taku and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium, November. Association for Computational Linguistics.

Luong, Minh-Thang and Christopher Manning. 2015. Stanford neural machine translation systems for spoken language domains. In Federico, Marcello, Sebastian Stüker, and Jan Niehues, editors, *Proceedings of the 12th International Workshop on Spoken Language Translation: Evaluation Campaign*, pages 76–79, Da Nang, Vietnam, December 3-4.

Mccloskey, Michael and Neil J. Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. *The Psychology of Learning and Motivation*, 24:104–169.

Michon, Elise, Josep Maria Crego, and Jean Senellart. 2020. Integrating domain terminology into neural machine translation. In *International Conference on Computational Linguistics*.

Mölder, Felix, Kim Philipp Jablonski, Brice Letcher, Michael B. Hall, Christopher H. Tomkins-Tinch, Vanessa V. Sochat, Jan Forster, Soohyun Lee, Sven O. Twardziok, Alexander Kanitz, Andreas Wilm, Manuel Holtgrewe, Sven Rahmann, Sven Nahnsen, and Johannes Köster. 2021. Sustainable data analysis with snakemake. *F1000Research*, 10:33.

Moslem, Yasmin, Rejwanul Haque, John D. Kelleher, and Andy Way. 2023a. Adaptive machine translation with large language models. In Nurminen, Mary, Judith Brenner, Maarit Koponen, Sirkku Latomaa, Mikhail Mikhailov, Frederike Schierl, Tharindu Ranasinghe, Eva Vanmassenhove, Sergi Alvarez Vidal, Nora Aranberri, Mara Nunziatini, Carla Parra Escartín, Mikel Forcada, Maja Popovic, Carolina Scarton, and Helena Moniz, editors, *Proceedings of the 24th Annual Conference of the European Association for Machine Translation*, pages 227–237, Tampere, Finland, June. European Association for Machine Translation.

Moslem, Yasmin, Gianfranco Romani, Mahdi Molaei, John D. Kelleher, Rejwanul Haque, and Andy Way. 2023b. Domain terminology integration into machine translation: Leveraging large language models. In *Proceedings of the Eighth Conference on Machine Translation*, pages 902–911, Singapore, December. Association for Computational Linguistics.

Neubig, Graham and Junjie Hu. 2018. Rapid adaptation of neural machine translation to new languages. In Riloff, Ellen, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 875–880, Brussels, Belgium, October-November. Association for Computational Linguistics.

Post, Matt. 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, October. Association for Computational Linguistics.

Qi, Peng, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

Rei, Ricardo, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In Webber, Bonnie, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online, November. Association for Computational Linguistics.

Semenov, Kirill, Vilém Zouhar, Tom Kocmi, Dongdong Zhang, Wangchunshu Zhou, and Yuchen Eleanor Jiang. 2023. Findings of the WMT 2023 shared task on machine translation with terminologies. In Koehn, Philipp, Barry Haddow, Tom Kocmi, and Christof Monz, editors, *Proceedings of the Eighth Conference on Machine Translation*, pages 663–671, Singapore, December. Association for Computational Linguistics.

Strubell, Emma, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in NLP. In Korhonen, Anna, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650, Florence, Italy, July. Association for Computational Linguistics.

Tiedemann, Jörg and Santhosh Thottingal. 2020. OPUS-MT — Building open translation services for the World. In *Proceedings of the 22nd Annual Conferenec of the European Association for Machine Translation (EAMT)*, Lisbon, Portugal.

Tiedemann, Jörg, 2009. *News from OPUS - A Collection of Multilingual Parallel Corpora with Tools and Interfaces*, volume V, pages 237–248.

Tiedemann, Jörg. 2020. The Tatoeba Translation Challenge – Realistic data sets for low resource and multilingual MT. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1174–1182, Online, November. Association for Computational Linguistics.

Zaragoza-Bernabeu, Jaume, Gema Ramírez-Sánchez, Marta Bañón, and Sergio Ortiz Rojas. "2022". "bicleaner AI: Bicleaner goes neural". In *"Proceedings of the Thirteenth Language Resources and Evaluation Conference"*, pages "824–831", "Marseille, France", June. "European Language Resources Association".