OPUS-CAT Terminology Systems for the WMT23 Terminology Shared Task

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

Abstract

This paper describes the submission of the OPUS-CAT project to the WMT 2023 terminology shared task. We trained systems for all three language pairs included in the task. All systems were trained using the same training pipeline with identical methods. Support for terminology was implemented by using the currently popular method of annotating source language terms in the training data with the corresponding target language terms.

1 Introduction

OPUS-CAT (Nieminen, 2021) is a collection of open source software consisting of a local neural machine translation (NMT) engine and plugins for computer-assisted translation (CAT) tools, such as Trados, memoQ and OmegaT. OPUS-CAT enables the use of NMT models trained in the OPUS-MT project (Tiedemann and Thottingal, 2020) in professional translation. As OPUS-CAT is aimed at professional translators, it is designed to be integrated into normal translation workflows. Multilingual term bases are one part of those workflows, so we have decided to implement a functionality for utilizing term bases in OPUS-CAT. This paper describes the methods used in OPUS-CAT for enforcing the use of terminology in machine translation output and the results of applying these methods to the data provided in the shared task. We trained new models for all three language pairs in the shared task. The shared task results were not available at the time of the submission of this paper.

2 Related work

Most published methods of constraining an NMT model to generate terminologically correct translations fall into three categories.

2.1 Constrained decoding

Hokamp and Liu (2017); Hasler et al. (2018): The beam search algorithm is modified to enforce the generation of target terms for each source term identified in the source sentence. The main advantage of constrained decoding is that it can be used with any model. The main disadvantages are slower decoding speed, and quality degradation due to the unconditional prioritizing of target terms, even in inappropriate contexts (such as generating the target term multiple times in the translation).

2.2 Pass-through term placeholders

Michon et al. (2020): Source terms identified in the source sentence are replaced by placeholders, which the NMT model passes through to the translation. The placeholders generated in the translation are then replaced by corresponding target terms. In order for the model to learn the correct pass-through behaviour, the model has to be trained with data that has been augmented with sentence pairs containing aligned placeholders on source and target sides. The main advantage of this approach is that the target terms are usually generated in correct positions. The disadvantage is that the information in the source term is discarded, which may degrade the quality of the overall translation. Generating morphological features for the target term may also be difficult.

2.3 Injecting target terms as soft constraints

Dinu et al. (2019): Source terms identified in the source sentence are annotated with target term information, and the NMT model uses these target term annotations to generate the term translations. Similar to the pass-through placeholder method, the training data of the model needs to be augmented with sentence pairs, where the source sentence has been annotated with target term information that also occurs in the target sentence. This will induce the model to generate translations that conform to the target term information present in the source text. While the constrained decoding and pass-through placeholder methods unconditionally

Proceedings of the Eighth Conference on Machine Translation (WMT), pages 912–918
December 6–7, 2023. ©2023 Association for Computational Linguistics

ally enforce the use of the specified terminology in the generated translation (they place hard constraints on the output), in this method terms are soft constraints on the output: contextual factors may cause the model to not use the specified term in the translation. This is the desired behaviour, since terms are often polysemous, and the specified term translation is usually only appropriate for one sense of the term. For instance, a terminology might specify a translation for the word *file*, but the translation would only be relevant for the sense of *file* meaning an individual file in a computer file system, instead of e.g. a physical file, a wood file, or the imperative of the verb *to file*.

The terminology support in OPUS-CAT is based on the soft constraint method as it is the simplest to implement and has performed best in previous evaluations (Alam et al., 2021b).

3 Model training

The models were trained using a modified version of Mozilla's *firefox-translations-training*[1], an end-to-end pipeline for building NMT models, based on the Snakemake workflow management system (Mölder et al., 2021). The pipeline loads, preprocesses, cleans and filters the training data, and trains and evaluates the NMT models. For this shared task, a terminology annotation workflow has been added to the pipeline[2].

3.1 Data

The models were trained using the data provided for the constrained track of WMT23. Since sufficient parallel data was available for each language pair, we did not include any back-translated monolingual data in the training corpus. This simplifies and speeds up training, and from the point of view of terminological correctness there does not appear to be any obvious benefit to using back-translated data, even though it would almost certainly increase general output quality.

3.2 Data cleaning

The data was cleaned and filtered using the standard *firefox-translations-training* workflow, which consists of monolingual cleaning of source and target corpora, followed by the filtering of parallel sentences with Bicleaner or Bicleaner-AI. Data for **en-cs** and **de-en** were filtered with Bicleaner-AI, while no parallel cleaning was performed for **zh-en**, as no Bicleaner-AI model for **zh-en** was available to the pipeline.

3.3 Terminology annotation

A part of the cleaned and filtered data was annotated with artificial term information (the annotation script is available from https://github.com/TommiNieminen/soft-term-constraints). First, artificial term data is generated from the parallel data:

1. **POS tagging and dependency parsing:** Stanza (Qi et al., 2020) was used to identify the parts-of-speech (POS) and dependency relations of the tokens in the source and target sentences.

2. **Chunking:** The POS and dependency data from step 1 was used to identify noun and verb phrase chunks in the source and target sentences.

3. **Word alignment:** The filtered parallel corpus was aligned on word-level using FastAlign (Dyer et al., 2013).

4. **Chunk alignment:** Source chunks that were aligned with target chunks were identified based on the word alignment from step 3.

The above method is identical to the one in Bergmanis and Pinnis (2021) except for the addition of chunking.

As analyzing sentences with Stanza is quite slow, only a small portion of the parallel data was analyzed (approximately one in ten sentences). The noun and verb phrase chunks identified on the basis of the analysis were saved and used to annotate the data using two different annotation methods (see table 2 for examples):

1. **Append**: The target language chunk was appended to the aligned source language chunk, with the source and target chunks separated with a special separator tag. A start tag was also added before the start of the source chunk, and an end tag was added after the end of the target chunk.

2. **Replace**: The source language chunk was replaced with the aligned target language chunk. The target chunk in the source sentence was tagged with start and end tags.

[1] https://github.com/mozilla/firefox-translations-training

[2] https://github.com/GreenNLP/firefox-translations-training/tree/develop

Language pair	Raw	Cleaned	Annotated
Chinese to English	35,452,884	28,840,867	2,884,058
German to English	294,331,299	182,977,635	18,297,581
English to Czech	56,288,239	35,046,151	2,704,588

Table 1: Amount of parallel sentences available for each language pair. Base model is trained with cleaned data, and the terminology models are fine-tuned with a combination of clean and annotated data or just annotated data (**-omit** models).

Source	`This product is no longer available`
Append	`This <term_start> product <term_end> produkt <trans_end> is no longer available.`
Replace	`This <term_start> produkt <term_end> is no longer available.`

Table 2: Examples of **append** and **replace** annotation methods

These methods are identical to the ones in Dinu et al. (2019) except for the use of tags instead of factors to identify terms (similar to Ailem et al. (2021).

Since a source sentence can potentially have any number of source terms, the training data needs to contain source sentences with different amounts of annotated terms. The annotation algorithm keeps track of how many sentences with *n* terms have been annotated so far, and tries to ensure that the sentence counts approximate a geometric series, where the amount of sentences gets halved for every extra term. For instance, the annotated corpus for en-cs contains 1,353,810 sentences with one term, 676,895 sentences with two terms, 338,414 sentences with three terms and so forth. The justification for the ratio is that most sentences will contain only few terms, so the lower counts should be emphasized in training.

4 Observations on the shared task

This year's terminology task differs in from real-world use of terminology in machine translation in two important aspects:

1. Source terms have been unambiguously identified.

2. Target terms are specified in an already inflected form. This inflected form has been extracted from a reference translation, and therefore has a high probability of being a correct form to use in a translation.

In actual use cases, the NMT system would have to identify the source terms based on a lemma form provided in a term base, and only the lemma form of the target term would be available. The probability of the lemma term occurring as such in a correct translation is much lower than for the inflected term from a reference translation. The shared task is therefore much easier than the real-world task of translating with a term base.

Due to the use of inflected terms, the shared task also favours soft constraint models where the model is trained on surface forms of terms instead of lemma forms. Because of this, the models we have submitted for the shared task all use surface forms of the terms. However, this will induce the models to learn a simple copy behaviour (Dinu et al., 2019), instead of the more desirable copy-and-inflect behaviour (Bergmanis and Pinnis, 2021). In our OPUS-CAT production models, we intend to use lemma-based constraints, since we expect them to perform better in real-world scenarios, especially with morphologically complex target languages.

5 Models

Five different models were trained for each language pair. All of the models were trained with Marian (Junczys-Dowmunt et al., 2018) using the **transformer-big** model architecture (Vaswani et al., 2017). For each language pair, a combined SentencePiece (Kudo and Richardson, 2018) vocabulary (32,000 symbols, out of which ten symbols were reserved as potential term tags by using the user-defined symbol functionality of SentencePiece) was trained and used for both source and target languages. As transformer-big models are costly to train, a single base model was trained for each language pair using just the filtered corpus, and the base model was then fine-tuned with data that had been augmented with the terminological annotations. Another motivation for using fine-tuning is the reuse of models: OPUS-CAT uses the OPUS-MT model collection that contains thousands of pre-trained models, and fine-tuning those models to support the use of terminology instead of training terminology models from scratch saves time and resources.

Yet another advantage of fine-tuning is that it makes it possible to quickly test the performance of different term annotation schemes. As mentioned, we experimented with the **append** and **replace** methods. For both methods, two models were trained, one where the annotated sentences were combined with the unannotated sentences when fine-tuning (**add**), and one where the unannotated sentences were omitted (**omit**). The expectation is that the **omit** model will specialize better to term translation, while the **add** model will retain better generic translation capabilities. In production use it may be best to use a specialized term model when terms are detected in the source sentence, and revert back to a generic model when no terms are detected.

The **zh-en** base model was trained until convergence (chrF validation metric did not improve for 20 consecutive validation steps). For the **en-cs** and **de-en** base models the training did not have time to converge before the deadline for shared task submission, but both models were trained sufficiently long to obtain competitive evaluation scores (on par with scores published for existing OPUS-MT models). The terminology models were trained by fine-tuning the base model with annotated data for one epoch.

When translating with a terminology model and a term base, a script is used to identify terms in the source text and to annotate the terms in the source sentence before translation, using the same annotation scheme as in the training data. Since the target side of the training data was not modified, the translation does not need to be post-processed.

5.1 Model n-best combination and reranking

For the submission to the shared task, we combine the outputs of the different types of models using a simple n-best reranking method (this is referred to as the **mixture** model in the tables):

1. An n-best list of size 8 is generated for each source sentence by each model.
2. Term occurrences are counted for each translation in the n-best lists.
3. The translation containing the most terms in all n-best lists is chosen as the final translation.
4. If translations from different models have the same amount of terms, the final translation is picked based on the following model hierarchy: **base**, **append**, **replace**, **append-omit**, **replace-omit** (the assumption is that the quality is best for the base model and worst for the omit models).
5. If there are multiple translations with the same amount of terms in a model's n-best list, translations higher in the n-best list are preferred.

The motivation for using this reranking method is that since the models use different approaches to generate translations, their combined n-best lists will be diverse, which increases the probability of finding a translation with correct terms. Also, in general it makes sense to rerank n-best lists in terminology translation, since the criteria for reranking is so clear (the highest amount of term occurrences).

6 Evaluation

6.1 Evaluation methods

General model performance was evaluated with BLEU and chrF metrics using sacreBLEU (Post, 2018).

Terminological correctness was evaluated by simply counting what percentage of the specified terms actually occur in the translation in the surface form in which they are defined. This naive method ignores two important issues: the correct placement of the term within the translation, and the matching of all other inflected forms of the term. Alam et al. (2021a) introduces more sophisticated term accuracy metrics to alleviate these issues, but we decided against applying them. Since we use evaluation mainly for sanity checking soft constraint models, which generally place terms correctly (and do not place terms at all if no plausible position is found for them), evaluating the correct placement is not crucial. Likewise, matching all inflected forms is not crucial in the context of this shared task, since the terminology is provided in an already inflected form, and our models have been trained with surface term annotations, and will likely have learned to copy the single inflected form provided to them.

DE-EN	flores-dev	wmt13	wmt16	wmt18	wmt20
base	**37.6/64.7**	**32.4/59.1**	**34.7/61.0**	32.5/58.6	23.3/51.6
append	**37.6**/64.6	**32.4**/58.9	34.5/60.9	32.5/58.5	23.0/51.6
append-omit	37.1/64.5	32.3/59.0	34.5/60.9	**32.6/58.7**	21.8/49.9
replace	**37.6**/64.6	32.1/58.7	34.3/60.7	32.3/58.4	**23.5/51.8**
replace-omit	37.2/64.5	32.2/58.9	34.2/60.7	32.5/58.6	22.0/50.4

ZH-EN	flores	wmt20	wmt21	wmt22
base	25.9/55.8	25.7/55.7	20.4/50.2	18.6/48.6
append	**27.2/56.7**	**27.8/57.3**	**22.2**/51.7	**19.9/49.9**
append-omit	26.8/56.2	27.0/56.5	21.6/51.0	19.3/49.2
replace	27.0/56.6	27.7/57.1	**22.2/51.8**	19.5/49.6
replace-omit	26.9/56.3	27.0/56.6	21.6/51.1	19.2/49.1

EN-CS	flores	wmt13	wmt16	wmt18	wmt20
base	**34.1/60.6**	**27.0/53.5**	**29.3/56.6**	**24.2/52.3**	20.5/**50.4**
append	33.4/60.1	26.8/53.3	29.2/56.5	23.8/51.9	**20.6/50.4**
append-omit	33.6/60.3	**27.0**/53.3	29.0/56.3	24.0/52.0	19.7/49.4
replace	33.5/60.3	26.8/53.4	29.2/56.5	24.1/52.1	20.4/50.2
replace-omit	33.6/60.2	26.8/53.2	29.0/56.2	23.9/51.9	20.2/49.7

Table 3: General translation performance measured as BLEU/chrF. 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 is to be expected that the term models perform worse in this evaluation.

		Exact term accuracy
DE-EN (100)	base	0.618
	append	0.911
	append-omit	0.854
	replace	0.886
	replace-omit	0.902
ZH-EN (100)	base	0.367
	append	0.933
	append-omit	0.933
	replace	0.900
	replace-omit	0.967
EN-CS (100)	base	0.496
	append	0.837
	append-omit	0.756
	replace	0.829
	replace-omit	0.772

Table 4: Term translation accuracy with the shared task dev set (sentence count is in parentheses under the language pair). In this scenario, the terms have been annotated to the input of the term models, and the term models perform better than the base model, as is to be expected.

6.2 Evaluation data

Models were evaluated against a selection of test sets allowed for the constrained track of WMT23 (see table 3 for results). Terminological correctness was evaluated using the development sets provided in the shared task (see table 4 for results). As the shared task development sets were quite small, we also created artificial terminology test sets for each language pair from the constrained track test sets, using the same annotation script that was used to annotate the training data (we did not use pre-existing terminologically annotated corpora due to the constrained track restrictions). Aligned noun and verb phrase chunks were identified in the test set sentences, and converted into sentence-level dictionaries similar to those in the shared task development sets (see table 5 for results).

Most NMT models trained on parallel data will exhibit some degree of copy behaviour, since source texts often contain target language words (this is especially common when the target language is English, due to its dominant position as a world language). Therefore it is plausible that the base models are already capable of copying target terms injected into the source sentence to the

DE-EN (6550)	Exact term accuracy	BLEU/ chrF
base	0.732	42.5/65.9
append	0.973	46.8/**69.2**
append-omit	0.942	**46.9/69.2**
replace	**0.977**	46.7/**69.2**
replace-omit	0.958	46.8/**69.2**
mixture	*0.997*	46.4/69.1
base-term	0.945	44.3/67.9

ZH-EN (5687)	Exact term accuracy	BLEU/ chrF
base	0.656	22.9/53.1
append	**0.949**	**26.1**/55.9
append-omit	0.899	24.9/54.8
replace	0.940	25.9/55.9
replace-omit	0.892	25.0/55.0
mixture	*0.985*	**26.1/56.2**
base-term	0.884	22.7/53.6

EN-CS (8204)	Exact term accuracy	BLEU/ chrF
base	0.651	28.3/55.5
append	0.902	31.4/58.4
append-omit	0.803	30.2/57.3
replace	**0.909**	31.2/58.3
replace-omit	0.861	30.2/57.6
mixture	*0.959*	**32.0/59.0**
base-term	0.827	29.0/56.8

Table 5: Term translation accuracy with the artificial term test set (test set sentence count is in parentheses under the language pair). Note that **mixture** will always have the best term accuracy, since it combines the output of other models based on term accuracy. Target terms have been added to the input for all models expect **base**. **base-term** is a **base** model translating input where source terms have been replaced with target terms.

translation. To determine the extent of this innate copying ability of the base model and the actual improvement brought by fine-tuning, a separate **base-term** test set was created from the artificial term test set by replacing the source terms in the source sentences with corresponding target terms.

6.3 Interpretation of the evaluation results

Results of the evaluation mostly conform to expectations. All soft constraint models outperform the base model in term translation, with the **append** and **replace** models performing best. This is somewhat surprising, since the **append-omit** and **replace-omit** models were expected to specialize better to term translation.

It is also surprising that the general translation quality of the soft constraint models is comparable to that of the base models. Strangely, the **zh-en** soft constraint models clearly outperform the base model even in general translation. This may be due to the **zh-en** base model converging early, after only 6 epochs of training. Still, it is counter-intuitive that fine-tuning with the small **omit** data sets consisting only of annotated sentences should noticeably improve general translation quality.

The results also confirm that the base models are quite capable of copying exact terms from the input sentence into the translation, especially the **de-en** model. However, injecting terms directly into the base model input seems to noticeably lower the overall translation quality.

7 Conclusion

Our submission for the shared task confirms that soft terminology constraint methods work with a variety of language pairs. We also demonstrate that soft constraint models can be created by fine-tuning base transformer models, which speeds up training and the investigation of different soft constraint methods and parameters. The results also indicate that fine-tuned soft constraint models have acceptable general translation quality, and do not require a back-off base model in production use.

Limitations

The soft constraint methods discussed assume terms are inflected, which is not usually the case when actually working with term bases. This limits the usability of the methods, especially with morphologically complex target languages. However, the annotation script also supports the use of lemma forms of terms. The reranking method used to produce the best term accuracy is computationally heavy, as it requires decoding with five separate **transformer-big** models.

Acknowledgements

The work reported in this paper has been funded and supported by the Swedish Culture Foundation in Finland and by the Finnish Academy. Computational resources were provided by CSC – IT Center for Science, Finland.

References

Melissa Ailem, Jingshu Liu, and Raheel Qader. 2021. Encouraging neural machine translation to satisfy terminology constraints. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1450–1455, Online. Association for Computational Linguistics.

Md Mahfuz Ibn Alam, 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.

Md Mahfuz Ibn Alam, 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. Association for Computational Linguistics.

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

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

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *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.

Eva Hasler, A. Gispert, Gonzalo Iglesias, and Bill Byrne. 2018. Neural machine translation decoding with terminology constraints. In *North American Chapter of the Association for Computational Linguistics*.

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

Marcin Junczys-Dowmunt, 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. Association for Computational Linguistics.

Taku Kudo 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. Association for Computational Linguistics.

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

Felix Mölder, 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.

Tommi Nieminen. 2021. OPUS-CAT: Desktop NMT with CAT integration and local fine-tuning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 288–294, Online. Association for Computational Linguistics.

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

Peng Qi, 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*.

Jörg Tiedemann 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.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.