Training Neural Machine Translation To Apply Terminology Constraints

Georgiana Dinu **Prashant Mathur** **Marcello Federico** **Yaser Al-Onaizan**

Amazon AI

{gddinu,pramathu,marcfede,onaizan}@amazon.com

Abstract

This paper proposes a novel method to inject custom terminology into neural machine translation at run time. Previous works have mainly proposed modifications to the decoding algorithm in order to constrain the output to include run-time-provided target terms. While being effective, these constrained decoding methods add, however, significant computational overhead to the inference step, and, as we show in this paper, can be brittle when tested in realistic conditions. In this paper we approach the problem by training a neural MT system to learn how to use custom terminology when provided with the input. Comparative experiments show that our method is not only more effective than a state-of-the-art implementation of constrained decoding, but is also as fast as constraint-free decoding.

1 Introduction

Despite the high quality reached nowadays by neural machine translation (NMT), its output is often still not adequate for many specific domains handled daily by the translation industry. While NMT has shown to benefit from the availability of in-domain parallel or monolingual data to learn domain specific terms (Farajian et al., 2018), it is not a universally applicable solution as often a domain may be too narrow and lacking in data for such bootstrapping techniques to work. For this reason, most multilingual content providers maintain terminologies for all their domains which are created by language specialists. For example, an entry such as *Jaws* (en) → *Lo Squalo* (it) would exist in order to indicate that the input *Jaws is a scary movie* should be translated as *Lo Squalo è un film pauroso*. While translation memories can be seen as ready-to-use training data for NMT domain adaptation, terminology databases (in short term bases) are more difficult to handle and there has been significant work on proposing methods to integrate domain terminology into NMT at run time.

Constrained decoding is the main approach to this problem. In short, it uses the target side of terminology entries whose source side match the input as decoding-time constraints. Constrained decoding and various improvements were addressed in Chatterjee et al. (2017), Hasler et al. (2018), Hokamp and Liu (2017) among others. Hokamp and Liu (2017) recently introduced the grid beam search (GBS) algorithm which uses a separate beam for each supplied lexical constraint. This solution however increases the run time complexity of the decoding process exponentially in the number of constraints. Post and Vilar (2018) recently suggested using a dynamic beam allocation (DBA) technique that reduces the computational overhead to a constant factor, independent from the number of constraints. In practice, results reported in Post and Vilar (2018) show that constrained decoding with DBA is effective but still causes a 3-fold increase in translation time when used with a beam size of 5.

In this paper we address the problem of constrained decoding as that of learning a *copy behaviour* of terminology at *training time*. By modifying the training procedure of neural MT we are completely eliminating any computational overhead at inference time. Specifically, the NMT model is trained to learn how to use terminology entries when they are provided as additional input to the source sentence. Term translations are inserted as inline annotations and additional input streams (so called source *factors*) are used to signal the switch between running text and target terms. We present experiments on English-to-German translation with terms extracted from two terminology dictionaries. As we do not assume terminology is available at train-time, all our

Append	En	All$_0$ **alternates**$_1$ **Stellvertreter**$_2$ shall$_0$ be$_0$ elected$_0$ for$_0$ one$_0$ term$_0$
Replace	En	All$_0$ **Stellvertreter**$_2$ shall$_0$ be$_0$ elected$_0$ for$_0$ a$_0$ term$_0$
	De	Alle **Stellvertreter** werden für eine Amtszeit gewählt

Table 1: The two alternative ways used to generate source-target training data, including target terms in the source and factors indicating source words (0), source terms (1), and target terms (2).

tests are performed in a **zero-shot** setting, that is with unseen terminology terms. We compare our approach against the efficient implementation of constrained decoding with DBA proposed by Post and Vilar (2018).

While our goal resembles that of Gu et al. (2017) (teaching NMT to use translation memories) and of Pham et al. (2018) (exploring network architectures to enforce copy behaviour), the method we propose works with a standard transformer NMT model (Vaswani et al., 2017) which is fed a hybrid input containing running text and inline annotations. This decouples the terminology functionality from the NMT architecture, which is particularly important as the state-of-the-art architectures are continuously changing.

2 Model

We propose an integrated approach in which the MT model learns, at training time, how to use terminology when target terms are provided in input. In particular, the model should learn to *bias* the translation to contain the provided terms, even if they were not observed in the training data. We augment the traditional MT input to contain a source sentence as well as a list of terminology entries that are triggered for that sentences, specifically those whose source sides match the sentence. While many different ways have been explored to augment MT input with additional information, we opt here for integrating terminology information as inline annotations in the source sentence, by either appending the target term to its source version, or by directly replacing the original term with the target one. We add an additional parallel stream to signal this "code-switching" in the source sentence. When the translation is appended this stream has three possible values: 0 for source words (default), 1 for source terms, and 2 for target terms. The two tested variants, one in which the source side of the terminology is retained and one in which it is discarded, are illustrated with an example in Table 1.

2.1 Training data creation

As we do not modify the original sequence-to-sequence NMT architecture, the network can learn the use of terminology from the augmentation of the training data. We hypothesize that the model will learn to use the provided terminology at training time if it holds true that when a terminology entry (t_s, t_t) is annotated in the source, the target side t_t is present in the reference. For this reason we annotate only terminology pairs that fit this criterion. The term bases used in the experiments are quite large and annotating all matches leads to most of the sentences containing term annotations. Since we want to model to perform equally well in a baseline, constraint-free condition, we limit the number of annotations by randomly ignoring some of the matches.

A sentence $\mathbf{s}$ may contain multiple matches from a term base, but we keep the longest match in the case of overlapping source terms. Moreover, when checking for matches of a term inside a sentence, we apply *approximate matching* to allow for some morphological variations in the term. In our current implementation, we use a simple character sequence match, allowing for example for base word forms to be considered matches even if they are inflected or as part of compounds.

3 Experiments

3.1 Evaluation setting

Parallel data and NMT architecture We test our approach on the WMT 2018 English-German news translation tasks[1], by training models on Europarl and news commentary data, for a total 2.2 million sentences. The baselines use this train data as is. For the other conditions sentences containing term annotations are added amounting to approximately 10% of the original data. We limit the amount of data added (by randomly ignoring some of the matched terms) as we want the model to work equally well when there are no terms provided as input. Note that these sentences are from the original data pool and therefore no actual new data is introduced.

We tokenize the corpora using Moses (Koehn et al., 2007) and perform joint source and target

[1] http://www.statmt.org/wmt18/translation-task.html

BPE encoding (Sennrich et al., 2016) to a vocabulary of 32K tokens. We use the source factor streams described in the previous section which are broadcast from word streams to BPE streams in a trivial way. We embed the three values of this additional stream into vectors of size 16 and concatenate them to the corresponding sub-word embeddings. We train all models using a transformer network (Vaswani et al., 2017) with two encoding layers and two decoding layers, shared source and target embeddings, and use the Sockeye toolkit (Hieber et al., 2018) (see full training configuration in the Appendix). The WMT newstest 2013 development set is used to compute the stopping criterion and all models are trained for a minimum of 50 and a maximum of 100 epochs. We compare the two methods we propose, *train-by-appending* and *train-by-replace* with the constrained decoding algorithm of Post and Vilar (2018) available in Sockeye in identical conditions and using a beam size of 5.

Terminology databases We extracted the English-German portions of two publicly available term bases, Wiktionary and IATE.[2] In order to avoid spurious matches, we filtered out entries occurring in the top 500 most frequent English words as well as single character entries. We split the term bases into train and test lists by making sure there is no overlap on the source side.

3.2 Results

We perform our evaluation on WMT newstest 2013/2017 as development (dev) and test sets respectively and use the test portions of Wiktionary and IATE to annotate the test set.[3] We select the sentences in which the term is used in the reference and therefore the copy behaviour is justified. The test set extracted with the Wiktionary term base contains 727 sentences and 884 terms, while the IATE one contains 414 sentences and 452 terms.

Table 2 shows the results. We report decoding speed, BLEU scores, as well as term use rates, computed as the percentage of times the term translation was generated in the output out of the total number of term annotations.

Term use rates and decoding speed The first observation we make is that the baseline model already uses the terminology translation at a high rate of 76%. Train-by-appending settings reach a term usage rate of around 90% while train-by-replace reaches even higher usage rates (93%-94%) indicating that completely eliminating the source term enforces the copy behaviour even more strongly. All these compare favourably to constrained decoding which reaches 99% on Wiktionary but only 82% on IATE.[4]

Second, the decoding speed of both our settings is comparable with that of the baseline, thus three times faster than the translation speed of constrained decoding (CD). This is an important difference because a three-fold increase of decoding time can hinder the use of terminology in latency-critical applications. Notice that decoding times were measured by running experiments with batch size 1 on a single GPU P3 AWS instance.[5]

	Wikt		
Model	Term%	BLEU (Δ)	Time(s)
Baseline	76.9	26.0	0.19
Constr. dec.	99.5	25.8 (-0.2)	0.68
Train-by-app.	90.7	26.9 (+0.9)↑	0.19
Train-by-rep.	93.4	26.3 (+0.3)	0.19
	IATE		
Model	Term%	BLEU (Δ)	Time(s)
Baseline	76.3	25.8	0.19
Constr. dec.	82.0	25.3 (-0.5)↓	0.68
Train-by-app.	92.9	26.0 (+0.2)	0.19
Train-by-rep.	94.5	26.0 (+0.2)	0.20

Table 2: Term usage percentage and BLEU scores of systems supplied with correct term entries, exactly matching the source and the target. We also provide the P99 latency numbers (i.e. 99% of the times the translations were completed within the given number of seconds). ↑ and ↓ represent significantly better and worse systems than the baseline system at a p-value < 0.05.

Translation quality Surprisingly, we observe significant variance w.r.t BLEU scores. Note that the terminologies affect only a small part of a sentence and most of the times the baseline already contains the desired term, therefore high BLEU variations are impossible on this test set. Constrained decoding does not lead to any changes in BLEU, other than a decrease on IATE with a small beam size of 5. However, all train-by-models show BLEU increases (+0.2 to +0.9), in

[2] Available at https://iate.europa.eu and https://www.wiktionary.org/

[3] https://github.com/mtresearcher/terminology_dataset

[4] We ran an additional experiment with a larger beam size of 20 and confirmed that constrained decoding can reach 99% term use on IATE, however at a drastic latency cost.

[5] The reason for using batch size 1 is that the CD implementation does not yet offer an optimized batched version.

src	Plain clothes officers from Dusseldorf's police force managed to **arrest** two women and two men, aged between 50 and 61, on Thursday.
constr dec	Plain Kleidungsbeamte der Polizei Dusseldorf konnten am Donnerstag zwei Frauen und zwei Männer im Alter von 50 bis 61 **Festnahme festzunehmen**.
train-by-app	Plain Kleidungsbeamte der Polizei von Dusseldorf konnten am Donnerstag zwei Frauen und zwei Männer **festzunehmen** , die zwischen 50 und 61 Jahre alt waren.
ref	Zivilfahndern der Dsseldorfer Polizei gelang am Donnerstag die **Festnahme** von zwei Frauen und zwei Mnnern im Alter von 50 bis 61 Jahren.
src	The letter extends an offer to cooperate with German authorities "when the difficulties of this **humanitarian** situation have been resolved" .
constr dec	Das Schreiben erweitert ein Angebot zur Zusammenarbeit mit den deutschen Behörden, "wenn die Schwierigkeiten dieser **humanitär** gelöst sind".
train-by-app	Das Schreiben erweitert ein Angebot zur Zusammenarbeit mit den deutschen Behörden, "wenn die Schwierigkeiten dieser **humanitären** Situation gelöst sind."
ref	"In seinem Brief macht Snowden den deutschen Behörden ein Angebot der Zusammenarbeit, wenn die Schwierigkeiten rund um die **humanitäre** Situation gelöst wurden .

Table 3: Examples in which constrained decoding leads to lower translation quality due to strict enforcement of constraints. The terms are *arrest* → *Festnahme* and *humanitarian* → *humanitär* (IATE terminology)

	Wiktionary	IATE
Model	BLEU (Δ)	BLEU (Δ)
Baseline	25.0	25.0
Constr. dec.	24.1 (-0.9)↓	23.7 (-1.3)↓
Train-by-app.	25.0 (0.0)	25.4 (+0.4)
Train-by-rep.	24.8 (-0.2)	25.3 (+0.3)

Table 4: Machine translation results of systems supplied with term entries showing exact source matches and approximate reference matches. ↓ represent significantly worse system than baseline with a p-value < 0.05.

particular the train-by-appending ones which have a lower terminology use rate. When examining the errors of the methods we observe cases in which constrained decoding alters the translation to accommodate a term even if a variation of that term is already in the translation as in the *festzunehmen/Festnahme* example of Table 3 (and sometimes even if the identical term is already used). A closer look at previous constrained decoding literature shows that most of the evaluations are performed differently than in this paper: The data sets contain only sentences for which the reference contains the term *and also the baseline fails to produce it.* This is an ideal setting which we believe to mimic few, if any, real world applications.

We observed an additional surprisingly positive behavior with our approach which constrained decoding does not handle: in some cases, our models generate morphological variants of terminology translations provided by the term base. Following up on this we set up an additional experiment by extending the previous set to also include approximate matches on the target side (identical to the approximate match in training explained in Section 2.1).

Table 4 shows these results. We observe that this test case is already more difficult for constrained decoding as well as for train-by-replace, most likely due to the removal of the original source side content. On the other hand, train-by-append still performs better than the baseline, while constrained decoding shows significant BLEU score reductions of 0.9-1.3 BLEU points. The *humanitarian* → *humanitär* example in Table 3 is a representative of the errors introduced by constrained decoding in case of source matching terms whose target side needs to be inflected.

4 Conclusion

While most of previous work on neural MT addressed the integration of terminology with constrained decoding, we proposed a *black-box* approach in which a generic neural MT architecture is directly trained to learn how to use an external terminology that is provided at run-time. We performed experiments in a *zero-shot* setting, showing that the copy behaviour is triggered at test time with terms that were never seen in training. In contrast to constrained decoding, we have also observed that the method exhibits flexible use of terminology as in some cases the terms are used in their provided form while other times inflection is performed. [6]

To our knowledge there is no existing work that

[6] Luong et al. (2015) and SYSTRANs Pure NMT system (Crego et al., 2016) are an exception to the constrained decoding approach as they replace entities with special tags that remain unchanged during translation and are replaced in a post-processing step. However this method also lacks flexibility, as the model will always replace the placeholder with the same phrase irrespective of grammatical context. We leave comparison to their approach to future work.

has a better speed vs performance trade-off than our method in the space of constrained decoding algorithms for neural MT, which we believe makes it particularly suitable for production environments.

5 Aknowledgments

The authors would like to thank Wael Hamza, Faisal Ladhak, Mona Diab and the anonymous reviewers for their advice and comments.

References

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

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

M. Amin Farajian, 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, Alicante, Spain. European Association for Machine Translation.

Jiatao Gu, Yong Wang, Kyunghyun Cho, and Victor O. K. Li. 2017. Search engine guided non-parametric neural machine translation. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 5133–5140, New Orleans, Louisiana, USA. Association for the Advancement of Artificial Intelligence.

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

Felix Hieber, 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 (Volume 1: Research Papers)*, pages 200–207. Association for Machine Translation in the Americas.

Chris Hokamp and Qun Liu. 2017. Lexically constrained decoding for sequence generation using grid beam search. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1535–1546.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180. Association for Computational Linguistics.

Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. *CoRR*, abs/1508.04025.

Ngoc-Quan Pham, Jan Niehues, and Alexander H. Waibel. 2018. Towards one-shot learning for rare-word translation with external experts. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation, NMT@ACL 2018, Melbourne, Australia, July 20, 2018*, pages 100–109.

Matt Post and David Vilar. 2018. Fast lexically constrained decoding with dynamic beam allocation for neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 1314–1324.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 6000–6010.

NMT Sockeye train parameters

```
encoder-config:
  act_type: relu
  attention_heads: 8
  conv_config: null
  dropout_act: 0.1
  dropout_attention: 0.1
  dropout_prepost: 0.1
  dtype: float32
  feed_forward_num_hidden: 2048
  lhuc: false
  max_seq_len_source: 101
  max_seq_len_target: 101
  model_size: 512
  num_layers: 2
  positional_embedding_type:
     fixed
  postprocess_sequence: dr
  preprocess_sequence: n
  use_lhuc: false

decoder config:
  act_type: relu
  attention_heads: 8
  conv_config: null
  dropout_act: 0.1
  dropout_attention: 0.1
  dropout_prepost: 0.1
  dtype: float32
  feed_forward_num_hidden: 2048
  max_seq_len_source: 101
  max_seq_len_target: 101
  model_size: 512
  num_layers: 2
  positional_embedding_type:
     fixed
  postprocess_sequence: dr
  preprocess_sequence: n

config_loss: !LossConfig
  label_smoothing: 0.1
  name: cross-entropy
  normalization_type: valid
  vocab_size: 32302

config_embed_source: !
   EmbeddingConfig
  dropout: 0.0
  dtype: float32
  factor_configs: null
  num_embed: 512
  num_factors: 1
  vocab_size: 32302

config_embed_target: !
   EmbeddingConfig
  dropout: 0.0
  dtype: float32
  factor_configs: null
  num_embed: 512
  num_factors: 1
  vocab_size: 32302
```