Encouraging Neural Machine Translation to Satisfy Terminology Constraints

Melissa Ailem, Jinghsu Liu and Raheel Qader
Lingua Custodia, France
{melissa.ailem,jingshu.liu,raheel.qader}@linguacustodia.com

Abstract

We present a new approach to encourage neural machine translation to satisfy lexical constraints. Our method acts at the training step and thereby avoiding the introduction of any extra computational overhead at inference step. The proposed method combines three main ingredients. The first one consists in augmenting the training data to specify the constraints. Intuitively, this encourages the model to learn a copy behavior when it encounters constraint terms. Compared to previous work, we use a simplified augmentation strategy without source factors. The second ingredient is constraint token masking, which makes it even easier for the model to learn the copy behavior and generalize better. The third one, is a modification of the standard cross entropy loss to bias the model towards assigning high probabilities to constraint words. Empirical results show that our method improves upon related baselines in terms of both BLEU score and the percentage of generated constraint terms.

1 Introduction

Neural Machine Translation (NMT) systems enjoy high performance and efficient inference (Sutskever et al., 2014; Bahdanau et al., 2014; Luong et al., 2015; Vaswani et al., 2017). However, when it comes to domain specific scenarios, where it is often necessary to take into account terminology constraints, NMT models suffer from the lack of explicit source-target correspondences making it challenging to enforce such constraints. For instance, consider the following sentence from the financial domain : *"**Holders** may submit instructions based on a minimum quantity being accepted by the **offeror.**"*. According to the financial terminology, the words *Holders* and *offeror* should be translated *porteurs* and *initiateur* respectively. Unfortunately, a generic English-French NMT model would translate the above sentence as: *"Les **titulaires** peuvent soumettre des instructions en fonction d'une quantité minimale acceptée par l'**offrant**."*, where the words *Holders* and *offeror* are translated into *titulaires* and *offrant* respectively. To address this limitation various approaches have been proposed. They can be grouped into two categories based on whether they enforce constraints at inference or at training time. The former family of methods changes the decoding step to inject the constraint terms in the output. While effective at satisfying constraints, these techniques tend to suffer from several weaknesses such as high computational cost at the decoding stage, decreased translation quality due to strict enforcement of terminology constraints (Hokamp and Liu, 2017; Post and Vilar, 2018), or ineptness if there are multiple constraints in the input/output (Susanto et al., 2020).

The other category of methods, which we follow in this work, integrates lexical constraints during training (Dinu et al., 2019). More precisely, they augment the training data in such a way as to inform the NMT model of the constrains that need to be satisfied (Crego et al., 2016; Song et al., 2019; Dinu et al., 2019). This type of approaches has the advantage of not changing the NMT model as well as of not introducing any additional computational overheads at inference time. One limitation of these methods is their soft nature, i.e, not all constraints are guaranteed to be present in the output.

In this paper we pursue the latter line of research and improve upon the recent work of Dinu et al. (2019) by (i) only using tags –without source factors – to distinguish between constraints and other words, (ii) performing constraint token masking for robustness/generalization purposes and (iii) modifying the standard cross-entropy loss to bias the model towards generating constraint terms. Empirical results show that our approaches improve both the BLEU sore and the number of satisfied constrains compared to previous work.

Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 1450–1455
August 1–6, 2021. ©2021 Association for Computational Linguistics

2 Related Work

Existing approaches can be cataloged based on whether they integrate constraints at inference/decoding (Chatterjee et al., 2017; Hasler et al., 2018; Hokamp and Liu, 2017) or at training time (Dinu et al., 2019).

Among methods of the first category, we can mention the Grid Beam Search (GBS) algorithm, which consists in reorganizing the vanilla beam search to place constraints correctly in the output as well as infer accurately the constraints-free parts. While successful in placing constrains compared to the original BS algorithm, GBS suffers from a high decoding time, it increases inference complexity exponentially with the number of constraints. To alleviate this issue, several improvement have been proposed, such as Dynamic Beam Allocation (DBA) (Post and Vilar, 2018) and its optimized extension, namely vectorized DBA (Hu et al., 2019). Despite an important gain in computational time, these methods still significantly increase the decoding time. For instance, the method of Post and Vilar (2018) is three times slower than the constraint-free beam search. More recently, Susanto et al. (2020) rely on the levenstein transformer (Gu et al., 2019), which uses an edit-based decoder iteratively refining the output using deletion and insertion operations. To enforce constraints using this model, Susanto et al. (2020) add one step to the decoder that consists in placing constraint terms in the output, and they further disallow the deletion operation on constraint terms. Albeit effective, the main limitation of this approach is in constraint ordering – when there is more than one constraint term in the output. That is, the initial order in which constraints have been placed remains unchanged.

Different from the above, the second family of methods integrates lexical constraints at training time. For instance, Crego et al. (2016) replace the terminology terms with placeholders during training and then add them back in a post-processing step. Song et al. (2019) proposed to annotate the training set by adding the target side of the terminology terms in the source sentences. A transformer model (Vaswani et al., 2017) is then trained on this augmented training set. This training data annotation has been also explored to teach the NMT to use translation memories (Gu et al., 2018) or to enforce copy behavior (Pham et al., 2018). Dinu et al. (2019) proposed two different ways to augment the training data, namely the append and the replace approaches. The former is similar to approach proposed in (Song et al., 2019), and the second requires to replace the source term of the constraints in the source sentence by its corresponding target side in the terminology entries. This method further uses source factors in order to distinguish the constraints from the rest of the source sentence. This is the closest approach to ours. The key differences are as follows. Our method uses only tags (without source factors) to specify constraints in the training set, and we further perform constraint-token masking, which improves model robustness/generalization as supported by our experiments. Moreover, we investigate a biased cross-entropy loss to encourage the NMT model to assign higher probabilities to constraint words.

3 Method

Our objective is to encourage neural machine translation to satisfy lexical constraints. To this end we introduce three changes to the standard procedure, namely training data augmentation, token masking, and cross-entropy loss modification.

TrAining Data Augmentation (TADA). Similar to previous work, the key idea is to bias the NMT model to exhibit a copy behavior when it encounters constraints. To this end, given some source sentence along with some constraints, we use tags to specify the constraints in the source sentence where relevant, as depicted in Figure 1. Note that as opposed to previous work, we do not introduce any further information (e.g., source factors), the constraints are specified using tags only.

Token MASKing (MASK). We further consider masking the source part of the constraint – tokens in blue – as illustrated in Figure 1 last row. We postulate that this might be useful from at least two perspectives. For one, this provides a more general pattern for the model to learn to perform the copy operation every time it encounters the tag $< S >$ followed by the MASK token. For another, this makes the model more apt to support conflicting constraints, i.e., constraints sharing the same source part but which have different target parts. This may be useful if some tokens must be translated into different targets for some specific documents and contexts at test time.

Weighted Cross-Entropy (WCE) Loss. Let $\mathbf{x} = (x_1, \ldots, x_{T_x})$ denote a sentence in some input language represented as a sequence of T_x words,

Source	His critics state that this will just increase the **budgetary deficit** .
Constraint	**budgetary deficit** → **Haushaltsdefizit**
TADA	His critics state that this will just increase the <S> **budgetary deficit** <C> **Haushaltsdefizit** </C> .
+MASK	His critics state that this will just increase the <S> **MASK MASK** <C> **Haushaltsdefizit** </C> .

Figure 1: Illustration of TrAining Data Augmentation (TADA) and MASK.

and $\mathbf{y} = (y_1, \ldots, y_{T_y})$ its translation in some target language. From a probabilistic perspective neural machine translation can be cast as estimating the conditional probability $p(\mathbf{y}|\mathbf{x})$ parametrized with neural networks, and which is usually assumed to factorize as follows,

$$p(\mathbf{y}|\mathbf{x}) = p(y_1|\mathbf{x}) \prod_{t=2}^{T_y} p(y_t|\mathbf{x}, y_{1:t-1}) \qquad (1)$$

where $y_{1:t-1}$ denote previously generated tokens. A predominant loss function in this context is the well know cross-entropy given by,

$$\mathcal{L} = -\log p(\mathbf{y}|\mathbf{x}) = -\sum_{t=1}^{T_y} \log p(y_t|\mathbf{x}, y_{1:t-1}) \quad (2)$$

As our objective is to encourage the NMT model towards generating the desired constraints, we propose to modify the above loss to provide a stronger learning signal to the model when it assigns a low probability to a constrain token y_t, as follows.

$$\mathcal{L} = -\sum_{t=1}^{T} w_{y_t} \log p(y_t|\mathbf{x}, y_{1:t-1}) \qquad (3)$$

where, $w_{y_t} = \alpha \geq 1$ if y_t is a constraint word, and $w_{y_t} = 1$ otherwise. As long as α is strictly greater than 1, the model would be biased towards assigning higher probabilities to constraint tokens. In practice one can set α to either a fixed value (e.g., selected based on some validation set) or using some annealing heuristic, i.e., start with $\alpha = 1$ and then gradually increase its value as learning progresses.

4 Experiments

4.1 Parallel Data

Following previous work (Dinu et al., 2019; Susanto et al., 2020), we assess our approach using the WMT 2018 English-German news translation tasks[1]. Our training dataset consists of nearly 2.2 million English-German parallel sentences from Europarl and news commentary. To compare our approach against existing works, we use two parallel English-German test sets extracted from WMT newstest 2017, and made available by Dinu et al. (2019) (see section 4.2 for details). Following the same authors, we use WMT newstest 2013 for validation containing 3000 parallel sentences.

4.2 Terminologies

In order to take into account lexical constraints, training, test and validation sets were annotated using two English-German bilingual terminologies extracted from IATE[2] and Wiktionary[3]. The two test sets released by (Dinu et al., 2019) have been extracted from WMT 2017 using IATE and Wiktionary respectively. The lexical constraints are added in the source sentences when source and target terms in the dictionaries entries are present in source and target sentences in the parallel dataset respectively. The test set extracted using IATE (wiktionary) contains 414 (727) sentences and 452 (884) term annotations. The training and validation sets have been annotated using both dictionaries making sure there is no overlap with the term annotations used in the test sets. For the training dataset, only 10% of the original data have been annotated with lexical constraints in order to preserve as far as possible the same performance when the model is not terminology-grounded (Dinu et al., 2019).

4.3 Settings

We use Moses tokenizer (Koehn et al., 2007) to tokenize our corpus and we learn a joint source and target BPE encoding (Sennrich et al., 2015) with 40k merge operations to segment it into sub-word units, resulting in a vocabulary size of 40388 words. Our models are trained using the transformer architecture (Vaswani et al., 2017) with three stacked encoders and decoders. The same hyperparameters as in (Dinu et al., 2019) were used where source and target embeddings are tied with the softmax layer. The models are trained for a minimum of 50 epochs and a maximum of 100 epochs with a batch size of 3000 tokens per iteration. Our validation set WMT 2013 is used to compute the stopping criterion. We use a beam size of 5 during inference for

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

[2] https://iate.europa.eu

[3] https://www.wiktionary.org/

Without MASK	
Source	For a while, one major problem has been finding homes subsequently for refugees that have been given <S> **certified** <C> **anerkannt** </C> status.
TADA	Seit geraumer Zeit besteht ein großes Problem darin, Häuser für Flüchtlinge zu finden, die **zertifiziert** wurden.
With MASK	
Source	For a while, one major problem has been finding homes subsequently for refugees that have been given <S> **MASK** <C> **anerkannt** </C> status.
TADA+MASK	Seit einiger Zeit besteht ein großes Problem darin, später Heime für Flüchtlinge zu finden , die **anerkannt** wurden.
+ WCE Loss	Seit einiger Zeit besteht ein großes Problem darin, später Heime für Flüchtlinge zu finden, die **anerkannt** worden sind.
Target	Ein schwerwiegendes Problem ist es seit einiger Zeit , Wohnungen für die Anschlussflüchtlinge zu finden, die **anerkannt** worden sind.

Figure 2: IATE : Example of en-de translation generated with TADA only and with TADA+MASK. With TADA only we observe that a variant of the target side of the constraint has been used (zertifiziert). In contrast, with MASK we observe that the target side of the constraint has been copied directly. Furthermore, using WCE loss leads to a translation which is even closer to the ground truth.

Without MASK	
Source	If perpetrators have to leave the country quicker , that will boost security and increase the <S> **general public** <C> **Bevölkerung** </C> 's <S> **approval** <C> **Zustimmung** </C> of refugee politics .
TADA	Wenn die Täter das Land schneller verlassen müssen , wird dies die Sicherheit erhöhen und die **Zustimmung** der Öffentlichkeit zur Flüchtlingspolitik erhöhen .
With MASK	
Source	If perpetrators have to leave the country quicker , that will boost security and increase the <S> **MASK MASK** <C> **Bevölkerung** </C> 's <S> **MASK** <C> **Zustimmung** </C> of refugee politics .
TADA+MASK	Wenn die Täter das Land schneller verlassen müssen , wird dies die Sicherheit erhöhen und die **Zustimmung** der **Bevölkerung** zur Flüchtlingspolitik erhöhen .
+ WCE Loss	Wenn die Täter das Land schneller verlassen müssen , wird dies die Sicherheit erhöhen und die **Zustimmung** der **Bevölkerung** zur Flüchtlingspolitik erhöhen .
Target	Wenn Straftäter schneller das Land verlassen müssten , erhöhe das aber die Sicherheit und stärke auch die **Zustimmung** der **Bevölkerung** für die Flüchtlingspolitik .

Figure 3: IATE : Example with multiple constraints. With TADA we observe that only one constraint is satisfied. Adding MASK makes it possible to satisfy both constraints.

all models. Regarding the proposed WCE Loss, we start training with $\alpha = 1$ for the first ninety epochs, then we continue learning for ten more epochs with $\alpha = 2$. In a pilot experiment, we explored different strategies to set the value of α, such as using $\alpha > 1$ from the beginning of training, increase the value of α every 5/10 iterations by $+0.1$, or train with $\alpha = 1$ for most iterations and then set α to a higher value (e.g., $\alpha = 2$) for the last few iterations. We retained the latter approach as it worked best among the ones we investigated.

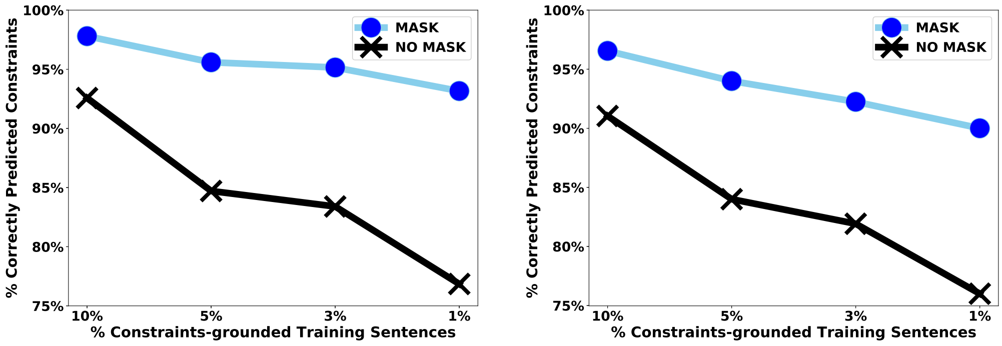
	IATE		Wiktionary	
	Term%	BLEU	Term%	BLEU
Previous works				
Transformer[†]	76.30	25.80	76.90	26.00
Const. Dec.[‡]	82.00	25.30	99.50	25.80
Source. Fact.[§]	94.50	26.00	93.40	26.30
Our work				
TADA+MASK	97.80	26.89	96.55	26.69
+WCE Loss	**98.02**	**27.11**	**96.84**	**26.73**

[†]:(Vaswani et al., 2017), [‡]: (Post and Vilar, 2018), [§]: (Dinu et al., 2019)

Table 1: Comparison with baselines in terms of BLEU score and Term usage percentage.

4.4 Results

We compare our approach to related NMT models integrating terminology constraints in terms of BLEU score (Papineni et al., 2002) and term usage rate (Term%), which is defined as the ratio between



(a) IATE test set

(b) Wiktionary test set

Figure 4: Percentage of correctly generated constraints with 10%, 5%, 3% and 1% of constraint-grounded training source sentences. With MASK the NMT model is less sensitive to the diminution of the percentage constraint-grounded sentences.

the number of constraints generated by the model and the total number of constraints. The results are presented in Table 1, and the main findings are as follows.

Comparison with baselines. Our methods significantly outperform the baselines in terms of both the BLEU score and the percentage of correctly generated constraint terms. TADA+MASK increases the BLEU score with +0.89% and +0.39% for IATE and Wiktionary test sets respectively. Regarding constraints (Terms%) we observe an improvement of +3.3% for IATE. Using the WCE loss further improves performances. For Wiktionary, Constrained Decoder reaches the highest terminology-use rate. However, the latter method suffers from a high decoding time and decreases translation quality.

Importance of MASK. To assess the impact of token masking, we report in Figure 4 the performance of TADA and TADA+MASK when the percentage of constraint annotations used in the training varies from 10% to 1%. Using MASK makes the model more robust to the diminution of the percentage of constraint-grounded sentences. The qualitative examples of Figures 2 and 3 further illustrate the benefit of token masking. In the former example, masking the source part of the constraint "certified" seems to have prevented the model from generating "zertifiziert" – see Figure 2's caption for details. Figure 3 shows a translation example containing multiple constraints to be satisfied. It seems that the use of MASK makes the model more apt to effectively handle and satisfy all the constraints. This is not necessarily the case of the model without MASK, which satisfies one constraint only. The results of Figures 2, 3 and 4 provide empirical support for the benefits of the proposed token masking in model generalization and robustness.

Impact of the WCE loss. To assess the impact of the WCE loss, we revisit the results of Table 1 and the examples of Figures 2 and 3. In all cases, we observe that using the proposed weighted cross-entropy loss further improves the quality of translation and the percentage of generated constraints, which demonstrate the benefits of biasing the model towards generating constraints tokens.

5 Conclusion

To encourage neural machine translation to satisfy terminology constraints, we propose an approach combining training data augmentation and token masking with a weighted cross-entropy loss. Our method is architecture independent and in principle it can be applied to any NMT model. Experiments on real-world datasets show that the proposed approach improves upon recent related baselines in terms of both BLEU score and the percentage of generated constraint terms.

In face of the multiplicity of methods to integrate terminology constraints, an interesting future work is to consider combining our method with other techniques within an ensemble approach.

6 Acknowledgments

This work was partially funded by the French Ministry of Defense.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

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.

Josep Crego, Jungi Kim, Guillaume Klein, Anabel Rebollo, Kathy Yang, Jean Senellart, Egor Akhanov, Patrice Brunelle, Aurelien Coquard, Yongchao Deng, et al. 2016. Systran's pure neural machine translation systems. *arXiv preprint arXiv:1610.05540*.

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

Jiatao Gu, Changhan Wang, and Junbo Zhao. 2019. Levenshtein transformer. In *Advances in Neural Information Processing Systems*, pages 11181–11191.

Jiatao Gu, Yong Wang, Kyunghyun Cho, and Victor OK Li. 2018. Search engine guided neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Eva Hasler, Adrià De Gispert, Gonzalo Iglesias, and Bill Byrne. 2018. Neural machine translation decoding with terminology constraints. *Proceedings of NAACL-HLT*, page 506–512.

Chris Hokamp and Qun Liu. 2017. Lexically constrained decoding for sequence generation using grid beam search. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, page 1535–1546.

J Edward Hu, Huda Khayrallah, Ryan Culkin, Patrick Xia, Tongfei Chen, Matt Post, and Benjamin Van Durme. 2019. Improved lexically constrained decoding for translation and monolingual rewriting. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 839–850.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration 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. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, page 1412–1421.

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

Ngoc-Quan Pham, Jan Niehues, and Alex Waibel. 2018. Towards one-shot learning for rare-word translation with external experts. *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, page 100–109.

Matt Post and David Vilar. 2018. Fast lexically constrained decoding with dynamic beam allocation for neural machine translation. *Proceedings of NAACL-HLT 2018*, page 1314–1324.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, page 1715–1725.

Kai Song, Yue Zhang, Heng Yu, Weihua Luo, Kun Wang, and Min Zhang. 2019. Code-switching for enhancing nmt with pre-specified translation. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, page 449–459.

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

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *arXiv preprint arXiv:1409.3215*.

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 5998–6008.