Understanding and Improving the Robustness of Terminology Constraints in Neural Machine Translation

Huaao Zhang[1], **Qiang Wang**[1,2], **Bo Qin**[1], **Zelin Shi**[1], **Haibo Wang**[1], **Ming Chen**[1*]
[1] RoyalFlush AI Research Institute, Hangzhou, China
[2] Zhejiang University, Hangzhou, China
{zhanghuaao, wangqiang3, qinbo, shizelin}@myhexin.com, chm@zju.edu.cn

Abstract

In this work, we study the robustness of two typical terminology translation methods: Placeholder (PH) and Code-Switch (CS), concerning (1) the number of constraints and (2) the target constraint length. We identify that existing terminology constraint test sets, such as IATE, Wiktionary, and TICO, are blind to this issue due to oversimplified constraint settings. To solve it, we create a new challenging test set of English-German, increasing the average constraint count per sentence from 1.1∼1.7 to 6.1 and the length per target constraint from 1.1∼1.2 words to 3.4 words. Then we find that PH and CS methods degrade as the number of constraints increases, but they have complementary strengths. Specifically, PH is better at retaining high constraint accuracy but lower translation quality as measured by BLEU and COMET scores. In contrast, CS has the opposite results. Based on these observations, we propose a simple but effective method combining the advantages of PH and CS. This approach involves training a model like PH to predict the term labels, and then during inference replacing those labels with target terminology text like CS, so that the subsequent generation is aware of the target term content. Extensive experimental results show that this approach can achieve high constraint accuracy and translation quality simultaneously, regardless of the number or length of constraints.[1]

1 Introduction

Although Neural Machine Translation (NMT) has achieved expressive performance improvement with the increase of model and data scale, it still struggles when involved in mismatched domains and rare entities (Koehn and Knowles, 2017). Terminology constraints (TC) is a popular solution that requires the model to generate the translation following the pre-provided terminology pairs and has been widely applied in commercial translation systems, such as Google, DeepL, etc.

Perhaps the most popular approach for TC is learning the constraint-aware model through data augmentation (Song et al., 2019; Dinu et al., 2019; Ailem et al., 2021; Bergmanis and Pinnis, 2021).[2] Early data augmentation is based on placeholder (PH). During training, PH methods replace the terminology terms in both source and target sentences with ordered labels (e.g., “$\mathcal{T}_1$”, “$\mathcal{T}_2$”), while the model predicts labels rather than the concrete terms at inference (Crego et al., 2016; Michon et al., 2020). The main drawback of PH methods is that the term labels lose the original semantic information, resulting in incoherent translation. Unlike PH methods, Code-Switch (CS) methods follow the standard model and generate term translations word by word by injecting target constraints in the source sequence (Song et al., 2019; Dinu et al., 2019; Ailem et al., 2021).

In this work, we focus on understanding the robustness of existing terminology constraint methods in challenging constraint settings in practice. Our contributions are four-fold:

- We point out that the widely used terminology constraint test sets (IATE[3], Wiktionary[4], TICO[5]) are too oversimplified to evaluate the robustness. To address this, we have created a new, challenging English-German terminology constraint test set containing 500 sentence pairs with multiple long constraints. This proposed test set significantly increases the average number of constraints from 1.1∼1.7 to 6.1, and the target constraint length from

[*] Corresponding author.

[1] https://github.com/zhajiahe/RTT

[2] We notice that all participating systems in the WMT21 Terminology Translation Task adopt this kind of method (Barrault et al., 2021).

[3] https://github.com/mtresearcher/terminology_dataset/tree/master/iate

[4] https://github.com/mtresearcher/terminology_dataset/tree/master/wiktionary

[5] https://tico-19.github.io/

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 6029–6042
July 9-14, 2023 ©2023 Association for Computational Linguistics

1.1∼1.2 words to 3.4 words. We will release this benchmark to promote the development of robust terminology translation.

- Through the proposed test set, we reveal that the performance of both Placeholder and Code-Switch degrades seriously with the increase of constraint count/length. However, it shows a strong complementarity in terms of constraint accuracy and translation quality; Placeholder is better at preserving accurate constraint, while Code-Switch yields higher translation quality as measured by COMET.

- Inspired by our findings, we propose a simple yet effective method for robust terminology translation (RTT), combining PH and CS's advantages. RTT learns to predict the term label and achieves a high constraint accuracy (like PH). Once a term label is generated, RTT appends the constraint counterpart in the decoding sequence to make the consequence generation aware of the semantic constraints (like CS).

- The experimental results of IATE, Wiktionary, and the proposed test set demonstrate that our approach can attain higher constraint accuracy and translation quality compared to using PH or CS alone, regardless of the number and length of the constraints. In addition, RTT maintains a slightly faster inference speed than the vanilla Transformer.

2 Background

Let $\boldsymbol{x} = \{x_1, \dots, x_M\}$ be the source sentence, $\boldsymbol{y} = \{y_1, \dots, y_N\}$ be the target sentence, and $\boldsymbol{C} = \{\langle \boldsymbol{s}_1, \boldsymbol{t}_1 \rangle, \dots, \langle \boldsymbol{s}_K, \boldsymbol{t}_K \rangle\}$ be the constraint set about $\boldsymbol{x}$ and $\boldsymbol{y}$, where $\boldsymbol{s}_i$ and $\boldsymbol{t}_i$ are the i-th source and target constraint respectively. Each constraint could be multi-word, e.g., $|\boldsymbol{s}_i| >= 1, |\boldsymbol{t}_i| >= 1$. Then TC asks the system must translate $\boldsymbol{s}_i$ into $\boldsymbol{t}_i$. In this section, we briefly introduce two typical TC methods based on data augmentation: Placeholder (PH) (Crego et al., 2016) and Code-Switch (CS) (Song et al., 2019; Dinu et al., 2019). We also describe some variants of them. Figure 1 explains the differences between these methods.

Method	Source	Target
Raw	$A\ B\ C\ D\ E\ F$	$a\ b\ c\ d\ e\ f$
PH	$A\ B\ \mathcal{T}_1\ E\ F$	$a\ b\ \mathcal{T}_1\ e\ f$
PH+SE	$A\ B\ [s]\ C\ D\ \mathcal{T}_1\ [e]\ E\ F$	$a\ b\ \mathcal{T}_1\ e\ f$
PH+SE+TE	$A\ B\ [s]\ C\ D\ \mathcal{T}_1\ c\ d\ [e]\ E\ F$	$a\ b\ \mathcal{T}_1\ e\ f$
CS	$A\ B\ c\ d\ E\ F$	$a\ b\ c\ d\ e\ f$
CS+SE	$A\ B\ [s]\ C\ D\ [e]\ c\ d\ [s]\ E\ F$	$a\ b\ c\ d\ e\ f$
RTT	$A\ B\ \mathcal{T}_1\ E\ F$	$a\ b\ \mathcal{T}_1\ c\ d\ e\ f$
RTT+SE	$A\ B\ [s]\ C\ D\ \mathcal{T}_1\ [e]\ E\ F$	$a\ b\ \mathcal{T}_1\ c\ d\ e\ f$

Table 1: Examples of different data augmentation methods. The terminology constraint is $CD \rightarrow cd$, $\mathcal{T}_1$ represents the term label, [s] and [e] are the start and end tag for the constraint, respectively. Red denotes the newly added token compared with the original text.

Placeholder. Placeholder is an early method for incorporating terminology constraints into machine translation. During training, the raw bitext is pre-processed by replacing source and target constraints with corresponding ordered labels $\mathcal{T}_i$. At inference time, source constraints are marked as ordered labels, and the model predicts the labels autonomously. The translation result is then obtained by replacing the labels with their corresponding target constraints in a post-processing step.

Code-Switch. Instead of using ordered labels, Code-Switch directly substitutes the source constraints with the corresponding target constraints in the input sentence. This allows the model to learn to copy the pre-specified target constraints from the input, so the decoder only needs to generate the target token step-by-step, like a standard system.

Variants. Considering the source side, vanilla PH and CS lose the source constraints' semantics due to direct replacement by labels or target constraints. A simple yet efficient solution is to retain the source constraints but use a tag to distinguish them from the replacement marks, as proposed by Dinu et al. (2019). We refer to this variant as the source-enhanced model (SE). For PH, we can further additionally tag target constraint information in the input sentence, denoted as the target-enhanced model (TE). Since CS has already injected target constraints into the input sentence, TE is not available for it.

3 On the robustness of terminology constraint

In this section, we explore the robustness of existing TC solutions from two aspects: (1) number of constraints and (2) target constraint length. We first point out the oversimplified problem in existing TC test sets in Section 3.1. We describe our proposed

Test Set	#Sent	#Term	#Avg Term	#Avg Word
IATE	414	452	1.1	1.2
Wiktionary	727	884	1.2	1.1
TICO	15676	26492	1.69	1.23
EFA	-	-	<3	$\sim\mathcal{U}(1,3)^{\dagger}$
Ours	500	3052	6.1	3.4

Table 2: Statistics on terminology constraint test sets. *Avg Word* denotes the average number of words in a target constraint. $\mathcal{U}$ denotes uniform sampling. † indicates the number of BPEs rather than words.

challenging TC test set in Section 3.2. Then we conduct comprehensive experiments to analyze the robustness of prior TC solutions in Section 3.3.

3.1 Oversimplified problem

As summarized in Table 2, oversimplified terminology constraint setups are widely present in published test sets, such as IATE, Wiktionary (Dinu et al., 2019), and TICO (Barrault et al., 2021), as well as extracted from alignment data (called EFA) (Wang et al., 2022; Guanhua et al., 2021). Typically, most open-source test sets have only one constraint per sentence, and the target constraint is also short, usually consisting of a single word. We suspect that this easy test set may lead to a misunderstanding of the practical performance of different methods. Intuitively, PH/PH+SE may suffer from poor translation fluency due to more target constraints, as the contents of these constraints are invisible during the generation of the decoder. However, this is not a severe problem for CS/CS+SE. On the other hand, PH should be insensitive to the constraint length, as it uses a single label as an alternative. In contrast, it is more difficult for CS to generate a long constraint due to more decoding steps required.

3.2 Proposed test set

To shed light on this issue, we made up a challenging TC test set. We notice that previous TC test sets generally are made by matching pre-build term database (e.g. IATE, Wiktionary) on existing bitext data sets. Since the term set is not strongly related to bitext, the number of matched constraints is not controlled. Instead, we first decide on the bitext data and then ask the linguistics expert to pick suitable sentence pairs to label constraints satisfying the requirement.

Specifically, we first collect WMT 13-18 test sets on English-German news translation task as the bitext data (14585 sentence pairs); The linguistic expert artificially hand-picks 500 sentence pairs for the study. These pairs are designed to include a minimum of 6 constraints each, drawn from a carefully curated set of noun phrases (such as the names of organizations, persons, movies and brands) and common expressions. By focusing on these types of constraints, the expert aims to replicate the linguistic conditions found in industrial systems as closely as possible. Table 10 in Appendix shows some samples in the proposed test set.

3.3 Experiments

Setup. We conduct experiments on the WMT16 En-De task (4.5M). We replicate the same data processing as Vaswani et al. (2017) with 32k joined BPE codes. We use the standard *transformer-base* model setting: 6-layer encoder/decoder, 8 attention heads, hidden size of 512, and FFN hidden size of 2048. We train all models with 65536 batch tokens for 120k updates and use checkpoint average of the last 5 checkpoints. To apply constraints on training data, we extract terminologies from two publicly available term databases, Wiktionary and IATE. In order to avoid spurious matches, we filtered out the top 10k frequent words in term databases. According to previous work (Dinu et al., 2019), the augmented data size is about 10% of the original data. We compare five TC models from two families, including PH, PH+SE, PH+SE+TE, CS, and CS+SE. The difference lies in augmented data is shown in Figure 1.

Metrics. We use several metrics to study the performance of different methods comprehensively. Specifically, in addition to reporting detokenized BLEU scores with *sacrebleu*[6] (Post, 2018), we also use COMET[7] (Rei et al., 2020) to evaluate the translation quality, inspired by the inconsistent trend in recent study (Helcl et al., 2022). Besides, we use strict sentence-level constraint accuracy (SCA) as the metric for terminology constraint. That is to say, only translations that satisfy all constraints in the sentence are considered correct. In contrast, most previous studies consider term-level constraint accuracy (TCA). Compared to TCA, SCA is more desired in the practical system because the translation may be severely misunderstood even if only one constraint is wrong.

[6] BLEU+case.mixed+numrefs.1+smooth.exp+tok.13a +version.1.5.1

[7] wmt20-comet-da

T_i	PH			PH+SE			PH+SE+TE			CS			CS+SE		
	BLEU	COMET	SCA	BLEU	COMET	SCA	BLEU	COMET	SCA	BLEU	COMET	SCA	BLEU	COMET	SCA
1	35.2	0.4098	97.8	36.8	0.4340	98.6	35.7	0.3995	98.8	36.3	0.4423	89.8	36.9	0.4537	89.6
2	36.1	0.4172	96.8	38.2	0.4242	96.0	36.9	0.4079	97.4	37.0	0.4381	82.6	37.9	0.4547	84.8
3	36.4	0.4156	95.0	38.5	0.4294	93.4	38.0	0.4122	94.2	38.4	0.4538	76.2	39.2	0.4627	75.2
4	36.8	0.3934	92.6	40.2	0.4351	89.4	38.8	0.4094	91.2	38.6	0.4555	69.0	40.1	0.4754	69.9
5	37.2	0.3787	93.0	41.1	0.4315	87.8	39.6	0.3867	87.0	39.3	0.4568	62.3	40.6	0.4719	58.8
6	36.6	0.3327	91.0	41.9	0.4232	84.0	40.4	0.3803	83.2	40.1	0.4579	57.2	41.4	0.4735	52.8
avg	36.4	0.3912	94.4	39.5	0.4296	91.5	38.2	0.3993	92.0	38.3	0.4507	72.9	39.4	0.4651	71.9

Table 3: Results of BLEU, COMET, and SCA against the number of constraint counts. *avg* denotes the average results of $\{T_1, \ldots, T_6\}$. The results of baseline without any TC are 36.0/0.4356/{60.0, 39.2, 28.0, 17.0, 12.8, 8.8} for BLEU, COMET and SCA in $T_1, \ldots, T_6$.

Results on various constraint counts. To simulate the case of various constraint counts, suppose there are N constraints for each sentence pair in the proposed test set, we randomly pick up 1,. . . ,N constraints. As a result, we conduct k TC test sets with constraint count ranges from 1 to k, denoted by $T_1, \ldots, T_k$, where every pair in T_i has exactly i constraints. Table 3 shows the results of three metrics (BLEU, COMET, SCA) along with the number of constraint counts ($k = 6$). We can see that:

(i) The SE variants based on either PH or CS significantly improve translation quality in terms of BLEU and COMET, which indicates that it is necessary to make the model aware of source terminology semantics. The exceptions are the SCA results when increasing T_i. The possible reason is that injecting too much non-source information (e.g., label, target constraints) in the input confuses the model, decreasing the copying success rate.

(ii) The PH family performs better in SCA than the CS family, especially for larger T_i. For example, the gap between PH and CS is 8.8% in T_1, extending to 26.8% in T_6. To our best knowledge, it is the first time to reveal that dramatic SCA degradation in CS models.

(iii) According to COMET, the family of CS has a superior translation quality compared to the PH family. We contend that COMET is a crucial supplement to BLEU for assessing terminology constraints. We observe that PH+SE and CS+SE have similar average BLEU scores, yet there is a substantial performance gap in COMET. This is due to BLEU's insensitivity to syntactic errors, whereas COMET imposes a hefty penalty, which is in line with earlier finding (Helcl et al., 2022).

L	Count	None	PH	PH+SE	PH+SE+TE	CS	CS+SE
1	427	85.5	99.3	97.9	98.1	96.7	94.6
2	618	72.8	98.5	98.2	97.7	92.9	90.5
3	698	65.6	98.4	96.8	96.8	92.6	91.5
4	528	55.7	97.5	97.3	96.6	88.6	90.9
5	343	51.0	98.3	98.0	96.2	87.8	86.0
>6	386	40.9	96.4	94.0	93.5	84.0	81.6
avg	-	63.3	98.1	97.1	96.7	90.9	89.7

Table 4: Results of term-level constraint accuracy (TCA) against the BPE length of target constraints. *None* represents the baseline system without any terminology constraint.

Results on various target constraint lengths. To study the impact of target constraint lengths, we report the TCA on different constraint length in the proposed test set as shown in Table 4. Like the trend of SCA in various constraint counts, we find that the PH family is again significantly superior to the CS family, especially when the length becomes longer. This result also proves that the benefits of label prediction in terms of constraint accuracy exist widely in different situations.

4 Our approach

4.1 Basic idea

The above experiments empirically show the solid complementarity between PH and CS, and here we analyze the reason behind it (see Figure 1). We suppose there are two sequences impacting the decoding process: ***prediction sequence*** and ***context sequence***, where the former is the realistic prediction by the model, and the latter decides the target context exposed to the model. For both PH-like and CS-like methods, the common problem is that they share the two sequences. Specifically, using placeholders in PH simplifies the prediction sequence but leads to the loss of constraint information (Figure 1a). In contrast, CS can observe the completed context but is redundant in the prediction sequence

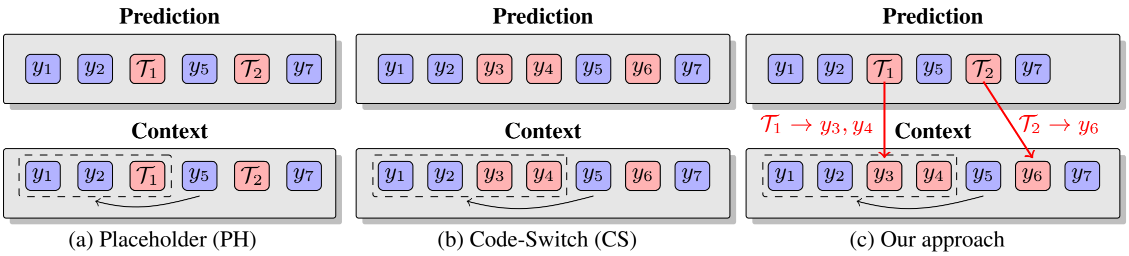
Prediction
y_1 y_2 $\mathcal{T}_1$ y_5 $\mathcal{T}_2$ y_7
Context
y_1 y_2 $\mathcal{T}_1$ y_5 $\mathcal{T}_2$ y_7
(a) Placeholder (PH)
Prediction
y_1 y_2 y_3 y_4 y_5 y_6 y_7
Context
y_1 y_2 y_3 y_4 y_5 y_6 y_7
(b) Code-Switch (CS)
Prediction
y_1 y_2 $\mathcal{T}_1$ y_5 $\mathcal{T}_2$ y_7
$\mathcal{T}_1 \to y_3, y_4$
Context
$\mathcal{T}_2 \to y_6$
y_1 y_2 y_3 y_4 y_5 y_6 y_7
(c) Our approach

Figure 1: Illustration of the difference between Placeholder (a), Code-Switch (b), and our approach (c). For the sake of clarity, we omit the source sentence. The target sentence is $y_1, \ldots, y_7$, which contains two constrains $\mathcal{T}_1 \to y_3, y_4$ and $\mathcal{T}_2 \to y_6$. Blue squares and red squares represent unconstrained and constrained tokens, respectively. The dotted rectangle represents the visible context of y_5. Compared with PH and CS, our approach keeps the prediction sequence simple and makes unconstrained tokens capture the semantics of constraints at the same time.

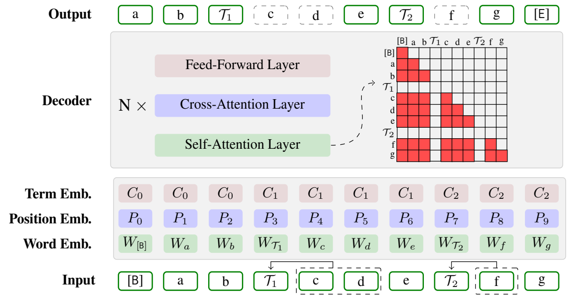
Output
a b $\mathcal{T}_1$ c d e $\mathcal{T}_2$ f g [E]
Decoder
N ×
Feed-Forward Layer
Cross-Attention Layer
Self-Attention Layer
Term Emb.
C_0 C_0 C_0 C_1 C_1 C_1 C_1 C_2 C_2 C_2
Position Emb.
P_0 P_1 P_2 P_3 P_4 P_5 P_6 P_7 P_8 P_9
Word Emb.
$W_{[B]}$ W_a W_b $W_{\mathcal{T}_1}$ W_c W_d W_e $W_{\mathcal{T}_2}$ W_f W_g
Input
[B] a b $\mathcal{T}_1$ c d e $\mathcal{T}_2$ f g

Figure 2: The training framework of proposed Robust Terminology Translation (RTT).

(Figure 1b). Thus, we propose to decouple the two sequences, which is the basis of our approach, referred to as RTT. As illustrated in Figure 1c, we still use placeholders to simplify the prediction sequence, but expose their semantics to future tokens by replacing the placeholder with its text. We explain how to efficiently implement RTT in both training and inference in the following section.

4.2 Training

RTT is agnostic to model architecture, and here we use the vanilla Transformer due to its wide application. Figure 2 illustrates the overall architecture.

Data augmentation. Since RTT behaves the same as PH for the source side, we only describe the data augmentation on the target side. Consider the target sentence $\boldsymbol{y} = \{y_1, \ldots, y_N\}$ and constraints $\boldsymbol{C} = \{\langle \boldsymbol{s}_1, \boldsymbol{t}_1 \rangle, \ldots, \langle \boldsymbol{s}_K, \boldsymbol{t}_K \rangle\}$, then we construct a new target sequence $\boldsymbol{y}^*$ by prepending an ordered term label $\mathcal{T}_i$ before the beginning of constraint $\boldsymbol{t}_i$. For instance, in Figure 2, we augment the original target input "a, b, c, d, e, f, g" by "$a, b, \underline{\mathcal{T}_1}, c, d, e, \underline{\mathcal{T}_2}, f, g$". We do not use any tags to distinguish term labels from normal tokens further to minimize the target sequence length.

Input embedding. In addition to the word embedding and sinusoidal positional embedding utilized in the standard Transformer, we introduce an additional learnable term embedding at the input layer. This term embedding provides information to the model about the number of constraints generated up to position i, thereby reducing the likelihood of generating repetitive constraints. Then the three embeddings are element-wise added to serve as the input to the Transformer layer. We note that the increase in the parameter size, $K \times d$, due to the inclusion of the term embedding is negligible compared to the overall network parameters. Here, K represents the maximum number of constraints in a sentence, and d corresponds to the hidden size. In our work, we set K to be 64.

Control visible context. In the Code-Switch method, term labels $\mathcal{T}$ are not present during the translation generation. To replicate this behavior, we suggest using a mask matrix in the self-attention layer of the RTT's decoder to make $\mathcal{T}$ invisible for subsequent tokens. Let $\mathcal{M}_{N\times N}$ be the mask matrix of the decoder self-attention layer, where $\mathcal{M}_{ij} = 1$ implies that the j-th target token is visible for the i-th target token. In the standard Transformer, $\mathcal{M}$ is a lower triangular matrix, which means that $\mathcal{M}_{ij} = 1$ if $i \leq j$. However, RTT additionally requires that $y_i \neq \mathcal{T}$ and $y_j \neq \mathcal{T}$, thus preventing term labels from being exposed to regular tokens.

Loss masking. In the context of RTT, we aim to encourage the model to focus more on learning to predict the term label $\mathcal{T}$ rather than the corresponding constraint tokens $\boldsymbol{t}$. This is because once $\mathcal{T}$ is predicted, the corresponding constraint tokens $\boldsymbol{t}$ will be automatically appended. To achieve this goal, we propose "Loss Masking" to guide the model's attention. Specifically, for each token y_i in the target sequence, we introduce a weight w_i to modify the original log-likelihood $log(P(y_i))$ by $w_i \times log(P(y_i))$. Then, we assign $w_i = 1$ to normal tokens or term labels in the target sequence. However, we set the weight w_i to 0 for tokens that correspond to the target constraint. This is also equivalent to treating the target constraint tokens as padding symbols. It is important to note that even though the target constraint tokens are masked, they can still be learned from the raw training data.

4.3 Inference

RTT follows the autoregressive translation paradigm. At decoding step i, if the prediction $\hat{y}_i$ is a normal token, it is appended to the decoding sequence and the next step is taken. However, if $\hat{y}_i$ is a term label, the sequence will also contain its corresponding target constraint retrieved from the input term base. The use of beam search in RTT can complicate this process, as other translation candidates must add several PADs (padding symbols) to compensate for the increased sequence length when a term label is generated. This can lead to a larger footprint and higher computational costs at inference, especially when the number of constraints or beam size is larger. To address this issue, we propose a dynamic padding strategy that reduces the number of redundant PADs. As shown in Figure 3, we append PADs at the beginning of the sequence rather than the end of a term label. This allows us to truncate the longest portion of common PADs once all candidates have some PADs at the beginning of the sequence, resulting in a shorter sequence. The effectiveness of this implementation trick is shown in Figure 5.

5 Experimental results

We first validate the effectiveness of proposed approach on the same setup as Section 3.3. Then, for fair comparisons to existing work, we also conduct experiments with WMT18 En-De training data (Europarl, News Commentary) and common test sets (IATE, Wiktionary).

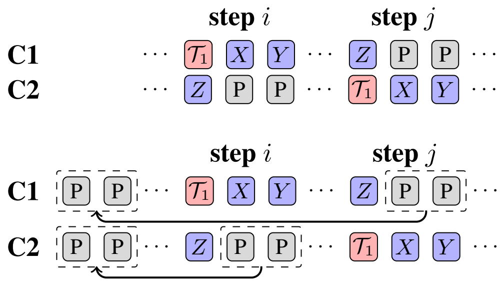
step i
step j
C1 ··· T_1 X Y ··· Z P P ···
C2 ··· Z P P ··· T_1 X Y ···
step i
step j
C1 P P ··· T_1 X Y ··· Z P P ···
C2 P P ··· Z P P ··· T_1 X Y ···

Figure 3: Illustration of naive padding (a) and proposed dynamic padding (b) with a beam size of 2. Red blocks represent term labels, respectively. The red, gray, and blue blocks represent the term label, PAD, and normal token, respectively. For naive padding, the sequence length is 6 with two redundant PAD symbols. In contrast, dynamic padding reduces the sequence length to 4 by moving PADs at the beginning and then truncating.

5.1 Results on proposed test set

We compared the performance of our proposed RTT model with two types of baseline methods: Placeholder approaches (PH) and Code-Switch approaches (CS). We also included the Transformer model as a baseline for comparison. Table 5 shows the average results of BLEU, COMET and SCA on our proposed test sets ($T_1, \ldots, T_6$). Unlike the PH family or CS family, which are either proficient in BLEU/COMET or SCA, our proposed RTT model achieves high translation quality and constraint accuracy at the same time. Specifically, the proposed RTT model with source enhancement (RTT+SE) achieved the highest BLEU score, with an average score of 40.2. It also achieved the highest COMET score, with an average of 0.4866. In terms of SCA, although RTT+SE slightly falls behind the best system (PH), it outperforms CS+SE in a significant performance gap (about 20%). Similar to PH, we note that additionally applying TE to RTT+SE is not consistently optimal. Therefore, unless otherwise stated, we take RTT+SE as our primary model in the following experiments. We note that the use of source enhancement is critical. Otherwise, the pure RTT model degrades severely due to asymmetric constraint information between the source and target side. That is, the constraints on the source side are term labels, while those on the target side are constraint text. To make the improvement of RTT clear, we also draw performance curves along with the change in the number of constraints, as

Method	BLEU	COMET	SCA
Transformer	36.0	0.4356	27.6
PH	36.4	0.3912	**94.4**
PH + SE	39.5	0.4296	91.5
PH + SE + TE	38.2	0.3993	92.0
CS	38.3	0.4507	72.9
CS + SE	39.4	0.4651	71.9
RTT	36.1	0.3943	91.0
RTT + SE	**40.2**	**0.4866**	91.9
RTT + SE + TE	40.1	0.4604	93.3

Table 5: Average BLEU, COMET and SCA scores on proposed test sets ($T_1, \ldots, T_6$).

Method	IATE		Wiktionary	
	TCA%	BLEU	TCA%	BLEU
Previous works				
Transformer	76.3	25.8	76.9	26.0
Const. Dec.	82.0	25.3	**99.5**	25.8
Source. Fact.	94.5	26.0	93.4	26.3
TADA.	98.0	27.1	96.8	26.7
Our work				
RTT + SE	**99.6**	**27.2**	98.3	**27.8**

Table 6: BLEU and **T**erm-level **C**onstraint **A**ccuracy (TCA) on IATE and Wiktionary test sets.

illustrated in Figure 4.

5.2 Comparisons to existing methods

To compare RTT fairly with existing methods, we perform additional experiments on WMT18 En-De task and replicate Dinu et al. (2019)'s setup. We use Europarl and News Commentary data as training data (2.2M), and report BLEU (sacrebleu) and TERM accuracy (TCA) on two easy TC test sets (IATE, Wiktionary). We consider several systems as our baselines, such as *Transformer* (Vaswani et al., 2017), *Const. Dec.* (Post and Vilar, 2018), *Source. Fact.* (Dinu et al., 2019) and *TADA* (Ailem et al., 2021). The results of our experiments are shown in Table 6. Our proposed RTT model with source enhancement (RTT + SE) achieved the highest BLEU score on both test sets, with 27.2 on IATE and 27.8 on Wiktionary. It also achieved the highest TCA on the IATE test set, with a score of 99.6%. On the Wiktionary test set, the RTT model achieved a TCA score of 98.3%, which was slightly lower than the constraint decoding method but still significantly higher than the other methods. Overall, the results indicate that our proposed RTT model is not only capable of handling difficult constraints, but also works well on such easy test sets.

6 Analysis

6.1 Inference speed

As illustrated in Figure 5, we compared the decoding step size and inference speed between our model and the vanilla Transformer. We also study the effect when our model decodes with naive padding (NP) and dynamic padding (DP). It is clear that the decoding step of NP is linearly increasing along with the number of constraints. Instead, the DP strategy successfully reduces an average of 52% decoding step and is very close to the baseline. As a bonus, the shorter decoding step in DP leads to a faster inference speed than NP. We note that RTT with DP can also run faster than the Transformer baseline when the constraint count is large because the corresponding target constraints in RTT are directly substituted to avoid costly model generation.

6.2 RTT without training

Without training, RTT can also be regarded as a modified Placeholder method. That is, the replacement of term labels transformers from the end of generation (as post-process) to the generation period. We are interested in whether the performance of Placeholder methods can be improved by plug-and-play the inference part of RTT. To this end, we tested it on two pre-trained models: PH and PH+SE, and Table 7 listed the results. We can see that the impact of RTT inference is different: PH+SE benefits in COMET (+0.0061) and SCA (+1.4%), while all metrics degrade in the vanilla PH model. We attribute it to asymmetry constraint information between the source and target like RTT and RTT+SE. Specifically, RTT inference makes the model aware of the semantics of constraint, while the source side of PH loses information. Even so, the improvement in PH+SE indicates that RTT inference can be used directly on the existing PH+SE model without further training.

6.3 Ablation study

In Table 8, we demonstrate the effects of two training components: term embedding (TermE) and loss masking (LM). As expected, using TermE and LM yields the best performance, as indicated by the

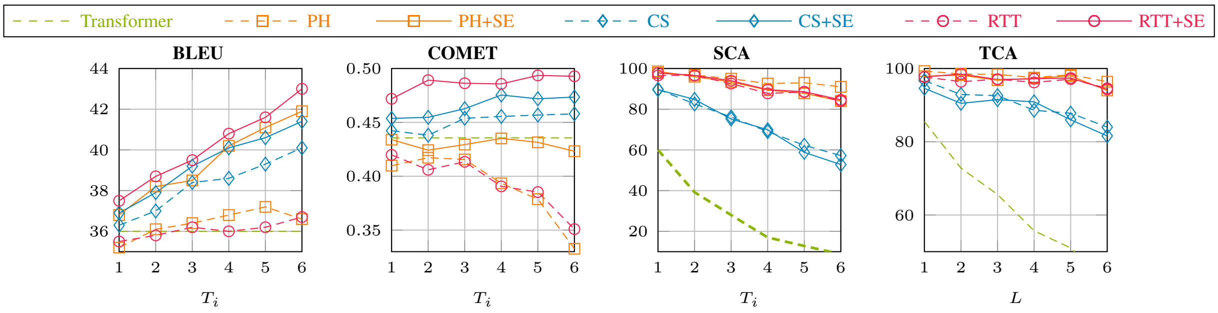

Figure 4: Performance curves against the number of constraint (T_i) and the length of target constraints (L).

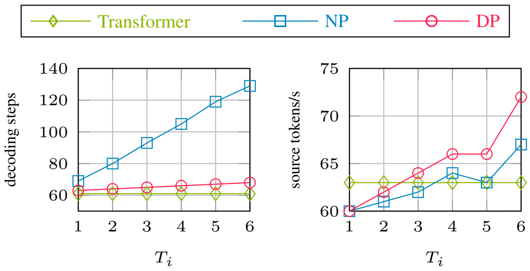

Figure 5: Compare the decoding step size (left) and inference speed (right) against varying constraint counts.

Model	RTT Inference	BLEU	COMET	SCA
PH	w/o	36.5	0.3912	94.4
	w/	33.8↓	0.3358↓	79.1↓
PH + SE	w/o	39.5	0.4296	91.5
	w/	39.4↓	0.4357↑	92.9↑

Table 7: Average results of applying RTT's inference to Placeholder methods on proposed test sets $T_1, \ldots, T_6$.

highest scores on all three evaluation metrics. Not utilizing either component leads to a decrease in performance. Notably, LM has a greater effect than TermE, suggesting that allowing the model to focus on learning the desired targets is essential. The model appears less sensitive to TermE, likely because the word embedding of the introduced term label implicitly informs the model of the state of constraints.

7 Related work

There have been several approaches to addressing the issue of translating specialized terminology in the field of machine translation. One branch of approaches focuses on the decoding process, such as extending the search space (Hokamp and Liu, 2017; Post and Vilar, 2018; Hu et al., 2019) or using a finite-state acceptor (Hasler et al., 2018), to enforce terminology translation strictly. However, these methods can incur high calculation costs and often result in poor translation quality (Guanhua et al., 2021). Another branch of approaches aims to modify the network architecture to better integrate with external terminologies through the use of alignment information (Song et al., 2020; Guanhua et al., 2021), vectorized terminology representation (Wang et al., 2022), or non-autoregressive translation (Susanto et al., 2020). These methods can potentially improve the integration of terminologies, but the big changes in network architecture greatly reduce their usability.

TermE	LM	BLEU	COMET	SCA
✓	✓	**40.2**	**0.4866**	**91.9**
✓	✗	39.4 (-0.8)	0.4550 (-0.0316)	88.5 (-3.4)
✗	✓	40.1 (-0.1)	0.4671 (-0.0195)	91.0 (-0.9)
✗	✗	39.5 (-0.7)	0.4438 (-0.0428)	89.0 (-2.9)

Table 8: Ablation study on term embedding (TermE) and loss masking (LM). The values in the parentheses differ from the first row, which serves as a reference point.

Data augmentation perhaps be the most widely used approach for terminology translation in machine translation. The placeholder method is an early solution for terminology translation by introducing special term labels (Crego et al., 2016). Michon et al. (2020) add linguistic information in the label to compensate for the semantic loss. Although effective, Placeholder techniques have difficulties producing smooth translations. Recently, Code-Switch methods have become popular as it overcomes this problem by allowing the model to generate word-by-word constraint translation, like standard neural machine translation. Song et al. (2019) directly replaces the source constraint with its translation in the input sequence; Dinu et al. (2019) uses some tags to distinguish between source constraints and target constraints; Ailem et al. (2021) further improves performance by masking the source constraints; Bergmanis and Pinnis (2021) uses target lemma to make the model learn morphology knowledge. As observed in our experiments, Code-Switch methods are fluent in translation but degrade in constraint accuracy. In contrast, our approach attempts to combine the strengths of Placeholder and Code-Switch, achieving high translation quality and constraint accuracy simultaneously.

8 Conclusion

Our study has highlighted the importance of taking robustness into account when comparing different methods of terminology constraint translation. We have found that the Placeholder and Code-Switch families are superior in different metrics, and the gap between them increases when dealing with more and longer terms. Additionally, we have observed that current TC test sets are inadequate for testing the robustness of different methods. To address this problem, we have created a new, more difficult terminology constraint test set. Moreover, we have proposed the RTT model, which merges the best features of the Placeholder and Code-Switch approaches and is capable of delivering both high translation quality and constraint accuracy regardless of the number of constraints and their length.

Limitations

While our proposed method demonstrates high translation quality and constraint accuracy, it is important to acknowledge that the hard copy mechanism may not be suitable for certain morphologically complex languages, such as Arabic. In Arabic, phrases or terminologies often involve conjunctions or prepositions and exhibit varying morphological forms. Unfortunately, our proposed method is not capable of effectively handling such cases, and addressing this challenge remains an open area for future research.

Acknowledgements

We would like to thank the anonymous reviewers for their helpful comments. We also thank Shuqin Pan for the writing suggestions.

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.

Loic Barrault, Ondrej Bojar, Fethi Bougares, Rajen Chatterjee, Marta R. Costa-jussa, Christian Federmann, Mark Fishel, Alexander Fraser, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Paco Guzman, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Tom Kocmi, Andre Martins, Makoto Morishita, and Christof Monz, editors. 2021. Proceedings of the Sixth Conference on Machine Translation. Association for Computational Linguistics, Online.

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.

Josep Maria Crego, Jungi Kim, Guillaume Klein, Anabel Rebollo, Kathy Yang, Jean Senellart, Egor Akhanov, Patrice Brunelle, Aurélien 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.

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.

Chen Guanhua, Chen Yun, and Li Victor O.K. 2021. Lexically constrained neural machine translation with explicit alignment guidance. In Proceedings of AAAI, volume 35, pages 12630–12638.

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

Jindřich Helcl, Barry Haddow, and Alexandra Birch. 2022. Non-autoregressive machine translation: It's not as fast as it seems. In Proceedings of the 2022

Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 1780–1790, Seattle, United States. Association for Computational Linguistics.

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 (Volume 1: Long Papers), pages 1535–1546, Vancouver, Canada. Association for Computational Linguistics.

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, Minneapolis, Minnesota. Association for Computational Linguistics.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In Proceedings of the First Workshop on Neural Machine Translation, pages 28–39.

Elise Michon, Josep Crego, and Jean Senellart. 2020. Integrating domain terminology into neural machine translation. In Proceedings of the 28th International Conference on Computational Linguistics, pages 3925–3937, Barcelona, Spain (Online). International Committee on 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, Brussels, Belgium. Association for Computational Linguistics.

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, Volume 1 (Long Papers), pages 1314–1324, New Orleans, Louisiana. Association for Computational Linguistics.

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

Kai Song, Kun Wang, Heng Yu, Yue Zhang, Zhongqiang Huang, Weihua Luo, Xiangyu Duan, and Min Zhang. 2020. Alignment-enhanced transformer for constraining nmt with pre-specified translations. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, pages 8886–8893.

Kai Song, Yue Zhang, Heng Yu, Weihua Luo, Kun Wang, and Min Zhang. 2019. Code-switching for enhancing NMT with pre-specified translation. 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 449–459, Minneapolis, Minnesota. Association for Computational Linguistics.

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

Shuo Wang, Zhixing Tan, and Yang Liu. 2022. Integrating vectorized lexical constraints for neural machine translation. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 7063–7073, Dublin, Ireland. Association for Computational Linguistics.

A Appendix

A.1 Samples of different terminology constraint test sets

We pick samples from IATE, Wiktionary, and our proposed test set randomly and show them in Table 10.

A.2 Detailed settings

We take two different settings for proposed test set and previous public test set, the detailed settings are listed in Table 9.

A.3 Samples of translation results

Table 11 shows the translation result of different systerms.

Settings	***Transformer Base***	***Transformer small***
Encoder layers	6	3
Decoder layers	6	3
Hidden size	512	512
FFN hidden size	2048	2048
Dropout	0.1	0.1
Label smoothing	0.1	0.1
Adam(α, β)	(0.9,0.98)	(0.9,0.98)
Learning rate	5e-4	5e-4
Total parameters	58.1M	35.1M
GPU time (h)	12.5	11.3
Beam size	4	5

Table 9: Detailed settings. The transformer base model is for proposed test set, the transformer small model is for public test set in order to be par with Ailem et al. (2021).

Dataset	Source	Target
IATE	Donald Trump wouldn't really mind if he lost the US presidential election in November: "Either it'll work out, or I'll go on a long, long holiday," the Republican candidate said in an telephone interview$_1$ with US television channel CNBC.	Verliert Donald Trump die US-Präsidentschaftswahlen im November, wäre ihm das relativ egal: "Letztlich wird es entweder klappen, oder ich habe einen sehr, sehr schönen, langen Urlaub", sagte der Kandidat der Republikaner in einem Telefon- Interview$_1$ mit dem US-Fernsehsenders CNBC.
Wiktion.	In 2014, police$_1$ raided the property and found more than 70g in cannabis$_2$ as well as scales, paraphernalia and £1,700 in cash.	2014 führte die Polizei$_1$ eine Razzia in dem Haus durch und fand über 70g Cannabis$_2$ sowie Waagen, Paraphernalien und Bargeld in Höhe von £1.700.
Ours (1)	She pointed out some exceptional successes$_1$, including commissioning and opening the new spring$_2$ (1994), purchasing land near the new spring$_3$ (1998), compensating farmers$_4$, renovating the elevated tanks$_5$ (1999), creating a new computer management system$_6$ (2004), and renewing some of the water pipes.	Als herausragende Ereignisse nannte$_1$ sie die Inbetriebnahme und Einweihung der neuen Brunnen$_2$ (1994), Grundstückskäufe im Bereich der neuen Brunnen$_3$ (1998) und Entschädigung der Landwirte$_4$, die Sanierung der Hochbehälter$_5$ (1999), die Neuanschaffung einer zentralen Computer-Steuerung$_6$ (2004) und die Teilerneuerungen von Wasserleitungen.
(2)	The vice chairman$_1$ of the Standing Committee's$_2$ Legislative Affairs Commission, Zhang Rongshun, and the Deputy Director$_3$ of the State Council's Hong Kong$_4$ and Macau Affairs Office$_5$, Feng Wei, were also due to speak in a series of briefings throughout the day$_6$.	Der stellvertretende Vorsitzende$_1$ der Rechtskommission des ständigen Ausschusses$_2$, Zhang Rongshun sowie der Vizedirektor$_3$ des Büros für Angelegenheiten Hongkongs$_4$ und Macaus des Staatsrates$_5$, Feng Wei, sollten im Verlauf des Tages$_6$ befragt werden.

Table 10: Samples of different terminology constraint test sets. The red text indicates the term, and the blue subscript denotes the corresponding order.

Terminology Constraints: $\mathcal{T}_1$**(EU transport commissioner→EU-Kommissarin für),** $\mathcal{T}_2$**(Hungarian justice minister→Justizminister in Ungarn),** $\mathcal{T}_3$**(financial statements→Finanzberichten)**	
Source	The EU transport commissioner-designate, Romanian socialist politician Rovana Plumb, and the proposed commissioner for EU enlargement, former Hungarian justice minister Laszlo Trocsanyi, were told their confirmation hearings could not go ahead because of inconsistencies in their financial statements, Reuters reported.
Transformer	Der designierte EU-Verkehrskommissar, der rumänische sozialistische Politiker Rovana Plumb, und der vorgeschlagene Kommissar für die EU-Erweiterung, der ehemalige ungarische Justizminister Laszlo Trocsanyi, wurden mitgeteilt, dass ihre Bestätigungsanhörungen aufgrund von Ungereimtheiten in ihren Jahresabschlüssen nicht stattfinden könnten, so Reuters.
PH	Dem designierten rumänischen sozialistischen Politiker Rovana Plumb und dem vorgeschlagenen Kommissar für die EU-Erweiterung, dem ehemaligen Justizminister in Ungarn Laszlo Trocsanyi, wurde mitgeteilt, dass ihre Anhörungen aufgrund von Widersprüchen in ihrem Finanzberichten nicht durchgeführt werden können. Reuters hat berichtet.
PH + SE	Dem designierten EU-Kommissarin für Verkehr, dem rumänischen sozialistischen Politiker Rovana Plumb, und dem vorgeschlagenen Kommissar für die EU-Erweiterung, dem ehemaligen Justizminister in Ungarn, Laszlo Trocsanyi, wurde mitgeteilt, dass ihre Bestätigungsanhörungen aufgrund von Unstimmigkeiten in ihren Finanzberichten nicht durchgeführt werden können, berichtet Reuters.
CS	Der designierten EU-Kommissarin für Verkehr, der rumänischen sozialistischen Politikerin Rovana Plumb, und dem vorgeschlagenen Kommissar für die EU-Erweiterung, dem ehemaligen Justizminister in Ungarn Laszlo Trocsanyi, wurde mitgeteilt, dass ihre Bestätigungsanhörungen aufgrund von Ungereimtheiten in ihren Finanzberichten nicht durchgeführt werden können, berichtete Reuters.
CS + SE	Der designierte EU-Kommissarin für Verkehr, die rumänische sozialistische Politikerin Rovana Plumb und der vorgeschlagene Kommissar für die EU-Erweiterung, der ehemalige Justizminister in Ungarn Laszlo Trocsanyi, wurden aufgrund von Ungereimtheiten in ihren Finanzberichten mitgeteilt, dass ihre Bestätigungsanhörungen nicht fortgesetzt werden könnten.
RTT + SE	Der designierte EU-Kommissarin für Verkehr, die rumänische sozialistische Politikerin Rovana Plumb, und der vorgeschlagene Kommissar für die EU-Erweiterung, der ehemalige Justizminister in Ungarn Laszlo Trocsanyi, wurden aufgefordert, ihre Bestätigungsanhörungen wegen Ungereimtheiten in ihren Finanzberichten nicht durchzuführen, berichtete Reuters.
Reference	Rovana Plum, angehende EU-Kommissarin für Verkehr und rumänische Sozialdemokratin, sowie László Trócsányi, nominiert als EU-Kommissar für die Erweiterung und Europäische Nachbarschaftspolitik und ehemaliger Justizminister in Ungarn, wurden nach einem Hearing darüber informiert, dass ihre Nominierungen aufgrund von Unstimmigkeiten in ihren Finanzberichten aufgehoben wurden, berichtete die Nachrichtenagentur Reuters.

Table 11: Samples of different system's results

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section Limitations

☒ A2. Did you discuss any potential risks of your work?
Our work attend to a new method for terminology constraint and hope to benefit terminology translation in the future.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Section Abstract and Introduction

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Section 3

☑ B1. Did you cite the creators of artifacts you used?
Section 3

□ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Not applicable. Left blank.

□ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Not applicable. Left blank.

□ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

□ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Not applicable. Left blank.

□ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Not applicable. Left blank.

C ☑ Did you run computational experiments?

Section 3,5,6

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section Appendix

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 3

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 3,5,6

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Section 3

D ☑ Did you use human annotators (e.g., crowdworkers) or research with human participants?
section 3.2

□ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Not applicable. Left blank.

□ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Not applicable. Left blank.

□ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Not applicable. Left blank.

□ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. Left blank.

□ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Not applicable. Left blank.