Neighbors Are Not Strangers: Improving Non-Autoregressive Translation under Low-Frequency Lexical Constraints

Anonymous ACL submission

Abstract

Lexically constrained neural machine translation (NMT) draws much industrial attention for its practical usage in specific domains. However, current autoregressive approaches suffer from high latency. In this paper, we focus on non-autoregressive translation (NAT) for this problem for its efficiency advantage. We identify that current constrained NAT models, which are based on iterative editing, do not handle low-frequency constraints well. To this end, we propose a plug-in algorithm for this line of work, *i.e.*, Aligned Constrained Training (ACT), which alleviates this problem by familiarizing the model with the source-side context of the constraints. Experiments on the general and domain datasets show that our model improves over the backbone constrained NAT model in constraint preservation and translation quality, especially for rare constraints.[1]

1 Introduction

Despite the success of neural machine translation (NMT) (Bahdanau et al., 2015; Vaswani et al., 2017; Barrault et al., 2020), real applications usually require the precise (if not exact) translation of specific terms. One popular solution is to incorporate dictionaries of pre-defined terminologies as *lexical constraints* to ensure the correct translation of terms, which has been demonstrated to be effective in many areas such as domain adaptation, interactive translation, etc.

Previous methods on lexically constrained translation are mainly built upon Autoregressive Translation (AT) models, imposing constraints at inference-time (Ture et al., 2012; Hokamp and Liu, 2017; Post and Vilar, 2018) or training-time (Luong et al., 2015; Ailem et al., 2021). However, such methods either are time-consuming in real-time applications or do not ensure the appearance of constraints in the output. To develop faster MT models for industrial applications, Non-Autoregressive

[1]Code will be released upon publication.

Source
Travellers [1.8K] screamed [24] and [2.8M] children [30.0K] cried [122] .
Target
Reisende [944] htten [9.9K] geschrien [13] und [2.6M] Kinder [20.1K] geweint [13] .
Terminology Constraints
scream → geschrien
Unconstrained translation
Reisende *schrien* und Kinder rieen. ⇒ *wrong term*
Soft constrained translation
Reisende *rien*. ⇒ *incomplete sentence & wrong term*
Hard constrained translation
Reisende *geschrien*. ⇒ *incomplete sentence*

Table 1: Translation examples of a lexically constrained non-autoregressive translation (NAT) model (Gu et al., 2019) under a low-frequency word as constraint. The underbraced word frequencies (uncased) are calculated from the vast WMT14 English-German translation (En-De) datasets (Vaswani et al., 2017).

Translation (NAT) has been put forth (Gu et al., 2018; Ghazvininejad et al., 2019; Gu et al., 2019; Qian et al., 2021), which aims to generate tokens in parallel, boosting inference efficiency compared with left-to-right autoregressive decoding.

Researches on lexically constrained NAT are relatively under-explored. Recent studies (Susanto et al., 2020; Xu and Carpuat, 2021) impose lexical constraints at inference time upon editing-based iterative NAT models, where constraint tokens are set as the initial sequence for further editing. However, such methods are vulnerable when encountered with low-frequency words as constraints. As illustrated in Table 1, when translated with a rare constraint, the model is unable to generate the correct context of the term “geschrien” as if it does not understand the constraint at all. It is dangerous since terms in specific domains are usually low-frequency words. We argue that the main reasons behind this problem are *1)* the inconsistency between training and constrained inference and *2)* the unawareness of the source-side context of the constraints.

To solve this problem, we build our algorithm based on the idea that the context of a rare constraint tends *not* to be rare as well, *i.e.*, "*a stranger's neighbors are not necessarily strangers*", as demonstrated in Table 1. We believe that, when the constraint is aligned to the source text, the context of its source-side counterpart can be utilized to be translated into the context of the target-side constraint, even if the constraint itself is rare. Also, when enforced to learn to preserve designated constraints at training-time, a model should be better at coping with constraints during inference-time.

Driven by these motivations, we propose a plug-in algorithm to improve constrained NAT, namely **A**ligned **C**onstrained **T**raining (ACT). ACT extends the family of editing-based iterative NAT (Gu et al., 2019; Susanto et al., 2020; Xu and Carpuat, 2021), the current paradigm of constrained NAT. Specifically, ACT is composed of two major components: *Constrained Training* and *Alignment Prompting*. The former extends regular training of iterative NAT with pseudo training-time constraints into the state transition of imitation learning. The latter incorporates source alignment information of constraints into training and inference, indicating the context of the potentially rare terms.

In summary, this work makes the following contributions: *1)* We identify and analyse the problems w.r.t. rare lexical constraints in current constrained NAT methods; *2)* We propose a plug-in algorithm for current constrained NAT models, *i.e.*, aligned constrained training, to improve the translation under rare constraints; *3)* Experiments show that our approach improves the backbone model w.r.t. constraint preservation and translation quality, especially for rare constraints.

2 Related Work

Lexically Constrained Translation Existing translation methods impose lexical constraints during either inference or training. At training time, constrained MT models include code-switching data augmentation (Dinu et al., 2019; Song et al., 2019; Chen et al., 2020) and training with auxiliary tasks such as token or span-level mask-prediction (Ailem et al., 2021; Lee et al., 2021). At inference time, autoregressive constrained decoding algorithms include utilizing placeholder tag (Luong et al., 2015; Crego et al., 2016), grid beam search (Hokamp and Liu, 2017; Post and Vilar, 2018) and alignment-enhanced decoding (Alkhouli et al., 2018; Song et al., 2020; Chen et al., 2021). For the purpose of efficiency, recent studies also focus on non-autoregressive constrained translation. Susanto et al. (2020) proposes to modify the inference procedure of Levenshtein Transformer (Gu et al., 2019) where they disallow the deletion of constraint words during iterative editing. Xu and Carpuat (2021) further develops this idea and introduces a reposition operation that can reorder the constraint tokens. Our work absorbs the idea of both lines of work. Based on NAT methods, we brings alignment information by terminologies to help learn the contextual information for lexical constraints, especially the rare ones.

Non-Autoregressive Translation Although enjoy the speed advantage, NAT models suffer from performance degradation due to the multi-modality problem, *i.e.*, generating text when multiple translations are plausible. Gu et al. (2018) applies sequence-level knowledge distillation (KD) (Kim and Rush, 2016) that uses an AT's output as an NAT's new target, which reduces word diversity and reordering complexity in reference, resulting in fewer modes (Zhou et al., 2020; Xu et al., 2021). Various algorithms have also been proposed to alleviate this problem, including incorporating latent variables (Kaiser et al., 2018; Shu et al., 2020), iterative refinement (Ghazvininejad et al., 2019; Stern et al., 2019; Gu et al., 2019; Guo et al., 2020), advanced training objective (Wang et al., 2019; Du et al., 2021) and gradually learning target-side word inter-dependency by curriculum learning (Qian et al., 2021). Our work extends the family of editing-based iterative NAT models for its flexibility to impose lexical constraints (Susanto et al., 2020; Xu and Carpuat, 2021).

3 Background

3.1 Non-Autoregressive Translation

Given a source sentence as $\boldsymbol{x}$ and a target sentence as $\boldsymbol{y} = \{y_1, \cdots, y_n\}$, an AT model generates in a left-to-right order, *i.e.*, generating y_t by conditioning on $\boldsymbol{x}$ and $y_{<t}$. An NAT model (Gu et al., 2018), however, discards the word inter-dependency in output tokens, with the conditional independent probability distribution modeled as:

$$P(\boldsymbol{y}|\boldsymbol{x}) = \prod_{t=1}^{n} P(y_t|\boldsymbol{x}). \quad (1)$$

Such factorization is featured with *high effi-*

Action	Implementation
Insertion	**Placeholder Classifier**: predicts the number of tokens ($0 \sim K_{max}$) to be inserted at every consecutive position pairs and then inserts the corresponding number of `[PLH]`. **Token Classifier**: predicts the actual target token of the `[PLH]`.
Deletion	**Deletion Classifier**: predicts whether each token (except for the boundaries) should be "kept" or "deleted".

Table 2: The implementation of insertion and deletion.

ciency at the cost of *performance drop* in translation tasks due to the *multi-modality* problem, *i.e.*, translating in mixed modes and resulting in token repetition, missing, or incoherence.

3.2 Editing-based Iterative NAT

For NATs, iterative refinement by editing is an NAT paradigm that suits constrained translations due to its flexibility. It alleviates the multi-modality problem by being autoregressive in editing previously generated sequences while maintaining non-autoregressiveness within each iteration. Thus, it achieves better performance than fully NATs while is faster than ATs.

Levenshtein Transformer To better illustrate our idea, we use Levenshtein Transformer (LevT, Gu et al., 2019) as the backbone model in this work, which is a representative model for constrained NAT based on iterative editing.

LevT is based on the Transformer architecture (Vaswani et al., 2017), but more flexible and fast than autoregressive ones. It models the generation of sentences as Markov Decision Process (MDP) defined by a tuple $(\mathcal{Y}, \mathcal{A}, \mathcal{E}, \mathcal{R}, \boldsymbol{y}^0)$. At each decoding iteration, the agent $\mathcal{E}$ receives an input $\boldsymbol{y} \in \mathcal{Y}$, chooses an action $a \in \mathcal{A}$ and gets reward r. $\mathcal{Y}$ is a set of discrete sentences and $\mathcal{R}$ is the reward function. $\boldsymbol{y}^0 \in \mathcal{A}$ is the initial sentence to be edited.

Each iteration consists of two basic operations, *i.e.*, *deletion* and *insertion*, which is described in Table 2. For the k-th iteration of the sentence $\boldsymbol{y}^k = (\texttt{<s>}, y_1, ..., y_n, \texttt{</s>})$, the insertion consists of placeholder and token classifiers, and the deletion is achieved by a deletion classifier. LevT trains the model with imitation learning to insert and delete, which lets the agent imitate the behaviors drawn from the expert policy:

- **Learning to insert**: edit to reference by inserting tokens from a fragmented sentence (*e.g.*, random deletion of reference).

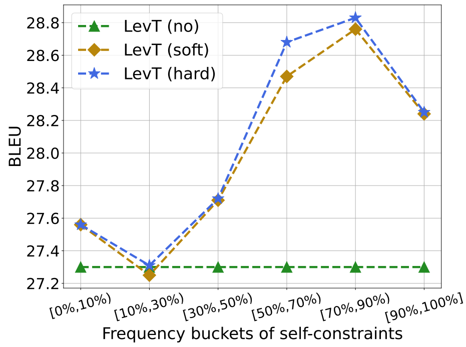


Figure 1: Ablation study of self-constrained translation on WMT14 En→De test set with Wiktionary terminology constraints (Dinu et al., 2019).

- **Learning to delete**: delete from the insertion result of the current training status to the reference.

The key idea is to learn how to edit from a ground truth after adding noise or the output of an adversary policy to the reference. The ground truth of the editing process is derived from the Levenshtein distance (Levenshtein, 1965).

Lexically Constrained Inference Lexical constraints can be imposed upon a translation model in: *1) soft constraints*: allowing the constraints not to appear in the translation; and *2) hard constraints*: forcing the constraints to appear in the translation. In NAT, the constraints are generally incorporated at inference time. Susanto et al. (2020) injects constraints as the initial sequence for iterative editing in Levenshtein Transformer (LevT, Gu et al., 2019), achieving soft constrained translation. And hard constrained translation can be easily done by disallowing the deletion of the constraints. Xu and Carpuat (2021) alters the deletion action in LevT with the reposition operation, allowing the reordering of multiple constraints.

3.3 Motivating Study: Self-Constrained Translation

According to Table 1, constrained NAT models seem to suffer from the low-frequency of lexical constraints, which is dangerous as most terms in practice are rare. To further explore the impact of constraint frequency upon NATs, we conduct a preliminary analysis on constrained LevT (Susanto et al., 2020). We sort words in each reference text based on frequency, dividing them into *six* buckets by frequency order (as in Figure 1), and sample a word from each bucket as lexical constraints for translation. We denote these constraints as *self-constraints*. In this way, we have six times the data, and the six samples derived from one raw sample only differ in the lexical constraints.

As shown in Figure 1, translation performance generally keeps improving as the self-constraint gets rarer. This is because setting low-frequency words in a sentence as constraints, which are often hard to translate, actually lightens the load of an NAT model. However, there are two noticeable performance drops around relative frequency ranges of 10%-30% and 90%-100%, denoted as *Drop#1* (-0.3 BLEU) and *Drop#2* (-0.6 BLEU). Drop#1 is probably because the constraint words within this range are mostly functional or less important. Such words are not as universal as ones at the left-most that can fit in most contexts and do not have to appear in the target due to multiple modes in translation.

However, we are more interested in the reasons for Drop#2 when constraints are low-frequency words. We assume a *trade-off* in *self-constrained* NAT: the model does not have to translate rare words as they are set as an initial sequence (constraints), but it will have a hard time understanding the context of the rare constraint due to 1) the rareness itself and 2) the lack of the alignment information between target-side constraint tokens and source tokens. Thus, the model does not know how many tokens should be inserted to the left and right of the constraint, which is consistent with the findings in Table 1.

4 Proposed Approach

The findings and assumptions discussed above motivate us to propose a plug-in algorithm for lexically constrained NAT models, *i.e.*, **A**ligned **C**onstrained **T**raining (ACT). ACT is designed based on two major ideas: *1) Constrained Training*: bridging the discrepancy between training and constrained inference; *2) Alignment Prompting*: helping the model understand the context of the constraints.

4.1 Constrained Training

As introduced in §3.2, constraints are typically imposed during inference time in NAT (Susanto et al., 2020; Xu and Carpuat, 2021). Specifically, lexical constraints are imposed by setting the initial sequence $\boldsymbol{y}^0$ as $(\texttt{<S>}, C_1, C_2, ..., C_k, \texttt{</S>})$, where $C_i = (c_1, c_2, ..., c_l)$ is the i-th lexical constrained word, l is the number of tokens in the i-th constraint, and k is the number of constraints.

However, such mandatory preservation of the constraints is not carried out during training. During imitation learning, *random deletion* is applied for ground-truth $\boldsymbol{y}^*$ to get the incomplete sentences $\boldsymbol{y}'$, producing the data samples for expert policies of how to insert from $\boldsymbol{y}'$ to $\boldsymbol{y}^*$. This leads to a situation where the model does not learn to preserve fixed tokens and organize the translation around the tokens. Such discrepancy could harm the applications of soft constrained translation.

To solve this problem, we propose a simple but effective **C**onstrained **T**raining (CT) algorithm. We first build *pseudo terms* from the target by sampling 0-3 words from reference as the pre-defined constraints for training.[2] Afterward, we disallow the deletion of pseudo term tokens during building data samples for imitation learning. This encourages the model to edit incomplete sentences containing lexical constraints into complete ones, bridging the gap between training and inference.

4.2 Alignment Prompting

As stated in §3.3, we assume the rareness of constraints hinders the model to insert proper tokens of its contexts (*i.e.*, *a stranger's neighbors are also strangers*). To make the matter worse, previous research (Ding et al., 2021) has also shown that lexical choice errors on low-frequency words tend to be propagated from the teacher (an AT model) to the student (an NAT model) in knowledge distillation.

However, terminologies, by nature, provide hard alignment information for source and target which the model can conveniently utilize. Thus, on top of constrained training, we propose an enhanced approach named **A**ligned **C**onstrained **T**raining (ACT). We propose to directly align the target-side constraints with the source words and prompt the alignment information to the model during both training and inference.

Building Alignment for Constraints We first align the source words to the target-side constraints, which are either pseudo constraints during training or actual constraints during inference. For each translated sentence constraints $\mathcal{C}_{\text{tgt}} = (C_1, C_2, ..., C_k)$, we use an external alignment tool external aligner, such as GIZA++ (Brown

[2]In the experiments, these pseudo constraints are sampled based on TF-IDF score to mimic the rare but important terminology constraints in practice.

Dataset (test set)	# Sent.	Avg. Len. of Con.	Avg. Con. Freq.
WMT14-WIKT	454	1.15	25,724.73
WMT17-IATE	414	1.09	3,685.42
WMT17-WIKT	728	1.22	26,252.70
OPUS-EMEA	2,996	1.95	2,187.63
OPUS-JRC	2,984	1.99	3,725.71

Table 3: Statistics of the test sets with target-side lexical constraints. "**Avg. Len. of Con.**" denotes the average number of words in a constraint. "**Avg. Con. Freq.**" is the average frequency of lexical constraints calculated with the training vocabularies of corresponding language.

et al., 1993; Och and Ney, 2003), to find the corresponding source words, denoted as $\mathcal{C}_{\text{src}} = (C'_1, C'_2, ..., C'_k)$.

Prompting Alignment into LevT The encoder in LevT, besides token embedding and position embedding, is further added with a learnable alignment embedding that comes from $\mathcal{C}_{\text{src}}$ and $\mathcal{C}_{\text{tgt}}$. We set the alignment value for each token in C'_i to i and the others to 0, which are further encoded into embeddings. The prompting of alignment is not limited to training, as we also add such alignment embeddings to source tokens aligned to target-side constraints during inference.

5 Experiments

5.1 Data and Evaluation

Parallel Data and Knowledge Distillation We consider the English→German (En→De) translation task and train all of the MT models on WMT14 En-De (3,961K sentence pairs), a benchmark translation dataset. All sentences are pre-processed via byte-pair encoding (BPE) (Sennrich et al., 2016) into sub-word units. Following the common practice of training an NAT model, we use the sentence-level knowledge distillation data generated by a Transformer, (Vaswani et al., 2017) provided by Kasai et al. (2020).

Datasets with Lexical Constraints Given models trained on the above-mentioned training sets, we evaluate them on the *test sets* of several lexically constrained translation datasets. These test sets are categorized into two types of standard lexically constrained translation datasets: *1)* Type#1: tasks from WMT14 (Vaswani et al., 2017) and WMT17 (Bojar et al., 2017), which are of the same general domain (news) as training sets; *2)* Type#2: tasks from OPUS (Tiedemann, 2012) that are of specific domains (medical and law). Particularly, the real application scenarios of lexically constrained MT models are usually domain-specific, and the constrained words in these domain datasets are relatively less frequent and more important.

Following previous work (Dinu et al., 2019; Susanto et al., 2020; Xu and Carpuat, 2021), the lexical constraints in Type#1 tasks are extracted from existing terminology databases such as Interactive Terminology for Europe (IATE)[3] and Wiktionary (WIKT)[4] accordingly. The OPUS-EMEA (medical domain) and OPUS-JRC (legal domain) in Type#2 tasks are datasets from OPUS. The constraints are extracted by randomly sampling 1 to 3 words from the reference (Post and Vilar, 2018). These constraints are then tokenized with BPE, yielding a larger number of tokens as constraints. The statistical report is shown in Table 3, indicating the frequencies of Type#2 datasets are generally much lower than Type#1 ones.

Evaluation Metrics We use BLEU (Papineni et al., 2002) for estimating the general quality of translation. We also use *Term Usage Rate* (Term%, Dinu et al., 2019; Susanto et al., 2020; Lee et al., 2021) to evaluate lexically constrained translation, which is the ratio of term constraints appearing in the translated text.

5.2 Models

We use Levenshtein Transformer (LevT, Gu et al., 2019) as the backbone model to ACT algorithm for constrained NAT. We compare our approach with a series of previous MT models on applying lexical constraints:

- *Transformer* (Vaswani et al., 2017), set as the AT baseline;
- *Dynamic Beam Allocation* (DBA) (Post and Vilar, 2018) for constrained decoding with dynamic beam allocation over Transformer;
- *Train-by-sep* (Dinu et al., 2019), trained on augmented code-switched data by replacing the source terms with target constraints or append on source terms during training;
- Constrained LevT (Susanto et al., 2020), which develops LevT (Gu et al., 2019) by setting constraints as initial editing sequence;

[3] https://iate.europa.eu
[4] https://www.wiktionary.org

Models	WMT17-IATE Term%	WMT17-IATE BLEU	WMT17-WIKT Term%	WMT17-WIKT BLEU	WMT14-WIKT Term%	WMT14-WIKT BLEU	Latency (ms)
Reported results in previous work							
Transformer (Vaswani et al., 2017)†	79.65	29.58	79.75	30.80	76.77	31.75	244.5
DBA (Post and Vilar, 2018)	82.00	25.30	99.50	25.80	-	-	434.4
Train-by-rep (Dinu et al., 2019)	94.50	26.00	93.40	26.30	-	-	-
LevT (Gu et al., 2019)†	80.31	28.97	81.11	30.24	80.23	29.86	92.0
w/ *soft constraint* (Susanto et al., 2020)	93.81	29.73	93.44	30.82	94.43	29.93	-
w/ *hard constraint* (Susanto et al., 2020)	100.00	30.13	100.00	31.20	100.00	30.49	-
EDITOR (Xu and Carpuat, 2021)†	83.00	27.90	83.50	28.80	-	-	121.7
w/ *soft constraint*	97.10	28.80	96.80	29.30	-	-	-
w/ *hard constraint*	100.00	28.90	99.80	29.30	-	-	134.1
Our implementation							
LevT†	78.32	**29.80**	80.20	30.75	**79.53**	29.95	**71.9**
+ constrained training (CT)†	78.76	29.46	**80.77**	**30.82**	79.13	30.24	78.6
+ aligned constrained training (ACT)†	**79.43**	29.57	80.20	30.63	77.17	**30.35**	77.0
LevT w/ *soft constraint*	94.25	30.11	93.78	30.92	94.88	30.38	79.5
+ constrained training (CT)	96.24	30.19	96.61	30.96	97.44	31.01	**75.4**
+ aligned constrained training (ACT)	**96.90**	**30.56**	**97.62**	**31.06**	**98.82**	**31.08**	76.3
LevT w/ *hard constraint*	100.00	30.31	100.00	30.65	100.00	30.49	82.7
+ constrained training (CT)	100.00	30.31	100.00	30.99	100.00	31.01	78.1
+ aligned constrained training (ACT)	100.00	**30.68**	100.00	**31.18**	100.00	**31.11**	**77.0**

Table 4: Translation results with lexical constraints. **Term%** is the constraint term usage rate. Method† translates *without* lexical constraints in input.

- EDITOR (Xu and Carpuat, 2021), a variant of LevT, replacing the delete action with a reposition action.

Implementation Details We use and extend the `FairSeq` framework (Ott et al., 2019) for training our models. We keep mostly the default parameters of `FairSeq`, such as setting d_{model} = 512, d_{hidden} = 2,048, n_{heads} = 8, n_{layers} = 6 and $p_{dropout}$ = 0.3. The learning rate is set as 0.0005, the warmup step is set as 4,000 steps. All models are trained with a batch size of 16,000 tokens for maximum of 300,000 steps with Adam optimizer (Kingma and Ba, 2014) on 2 NVIDIA GeForce RTX 3090 GPUs with gradient accumulation of 4 batches. Checkpoints for testing are selected from the average weights of the last 5 checkpoints. For Transformer (Vaswani et al., 2017), we use the checkpoint released by Ott et al. (2018).

5.3 Main Results

Table 4 reports the performance of LevT with ACT (as well as the CT ablation) on the type 1 tasks (WIKT and IATE as terminologies), compared with baselines. In general, the results indicate the proposed CT/ACT algorithms achieve a consistent gain in performance, term coverage, and speed over the backbone model mainly in the setting of constrained translation.

When translating with *soft* constraints, *i.e.*, the constraints need not appear in the output, adding ACT to LevT helps preserve the terminology constraints (+∼5 Term%) and improves translation performance (+0.31-0.88 on BLEU). If we enforce *hard* constraints, the term usage rate doubtlessly reaches 100%, with reasonable improvements on BLEU. When translating *without* constraints, however, adding ACT does not bring consistent improvements as hard and soft constraints do, which could be attributed to the discrepancy between training and inference.

As for the ablation for CT and ACT, we have two observations: 1) term usage rate increases mainly because of CT, and can be further improved by ACT; 2) translation quality (BLEU) increases due to the additional hard alignment of ACT over CT. The former could be attributed to the behavior of *not deleting the constraints* in CT. The latter is because of the introduction of source-side information of constraints that familiarize the model with the constraint context.

Table 3 also shows the efficiency advantage of non-autoregressive methods compared with autoregressive ones, which is widely reported in the NAT research literature. The proposed methods do not cause drops in translation speed against the backbone LevT. When translating with lexical constraints, LevT with CT or ACT is even faster than LevT. In contrast, constrained decoding methods for autoregressive models (*i.e.*, DBA) nearly double the translation latency. Since the main purpose of non-autoregressive research is developing effi-

Model	OPUS-EMEA		OPUS-JRC	
	Term%	BLEU	Term%	BLEU
LevT†	52.40	27.90	**55.39**	30.24
+ ACT†	**53.41**	**28.30**	55.35	**31.01**
LevT w/ *soft*	83.37	30.35	84.32	32.53
+ ACT	**92.09**	**32.02**	**91.94**	**33.70**
LevT w/ *hard*	100.00	30.77	100.00	30.08
+ ACT	100.00	**32.30**	100.00	**34.09**

Table 5: Experiments on test sets from OPUS, which is outside the training set (WMT14 En→De). Results shows ACT brings larger performance for lower-frequency lexical constraints within these datasets.

cient algorithms, such findings could facilitate the industrial usage for constrained translation.

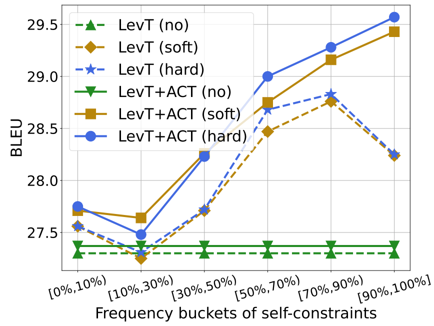
Translation Results on Domain Datasets For a generalized evaluation of our methods, we apply the models trained on the general domain dataset (WMT14 En-De) to medical (OPUS-EMEA) and legal domains (OPUS-JRC). As seen in Table 5, even greater performance boosts are witnessed. When trained with ACT, both term usage (+∼8-10 Term%) and translation performance (up to 4 BLEU points) largely increase, which is more significant than the general domain.

The reason behind this observation is that the backbone LevT would have a hard time recognizing them as constraints since the lexical constraints in these datasets are much rarer. Therefore, forcing LevT to translate with these rare constraints would generate worse text, *e.g.*, BLEU drops for 2.45 points on OPUS-JRC than with soft constraints. And when translating with soft constraints, LevT over-deletes these rare constraints. In contrast, the context information around constraints is effectively pin-pointed by ACT, so ACT would know the context ("neighbors") of the rare constraint ("strangers") and insert the translated context around the lexical constraints. In this way, more terms are preserved by ACT, and the translation achieves better results.

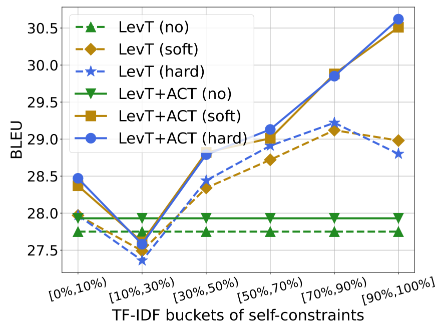
6 Analysis

6.1 Self-Constrained Translation *Revisited*

As a direct response to our motivation in this paper, we revisit the ablation study of self-constrained NAT in §3.3 with the proposed ACT algorithm. Same as before, we build self-constraints from each target sentence and sort them by frequency. As shown in Figure 2(a), different from constrained



(a) Sorting self-constraints by frequency.



(b) Sorting self-constraints by TF-IDF.

Figure 2: Extended self-constrained translation results on WMT14-WIKT. Each and every word of a reference is used as a lexical constraint (*i.e.*, self-constraint) for translation, sorted by frequency or TF-IDF.

LevT that suffers from Drop#2 (§3.3), ACT managed to handle this scenario pretty well. Following the motivations given in §3.3, when constraints become rarer, ACT successfully breaks the *trade-off* with better understanding of the provided contextual information.

What if the self-constraints are sorted based on TF-IDF? We also study the importance of different words in a sentence via TF-IDF by forcing them as constraints. As results in Figure 2(b) show, we have very similar observations from frequency-based self-constraints at Figure 2(a), and the gap between LevT and LevT + ACT is even higher as TF-IDF score reaches the highest.

6.2 How does ACT perform under different kinds of lexical constraints?

The experiments in §6.1 create pseudo lexical constraints by traversing the target-side reference for understanding the proposed ACT. In the following analyses, we study different properties of lexical constraints, *e.g.*, frequency and numbers, and how they affect constrained translation.

Model	WMT14-WIKT				WMT17-IATE				WMT17-WIKT			
	ALL	HIGH	MED.	LOW	ALL	HIGH	MED.	LOW	ALL	HIGH	MED.	LOW
LevT†	29.95	30.46	**28.03**	31.49	**29.80**	**30.08**	**29.72**	**29.45**	**30.75**	**30.96**	29.09	32.16
+ ACT†	**30.35**	**30.68**	28.00	**32.54**	29.57	29.63	29.57	29.20	30.63	30.35	**29.11**	**32.46**
LevT w/ *soft*	30.38	30.37	28.50	32.19	30.11	29.25	30.67	30.04	30.92	30.70	**29.58**	32.23
+ ACT	**31.08**	**30.48**	**29.18**	**33.85**	**30.56**	**29.93**	**31.05**	**30.36**	**31.06**	**30.72**	29.53	**32.73**
LevT w/ *hard*	30.49	**30.50**	28.67	31.99	30.31	29.46	30.66	30.37	30.65	30.28	29.44	32.00
+ ACT	**31.11**	30.23	**29.32**	**33.85**	**30.68**	**29.97**	**31.18**	**30.67**	**31.18**	**30.58**	**29.71**	**32.90**

Table 6: Ablation results of terminology-constrained En→De translation tasks w.r.t. word frequency of terms.

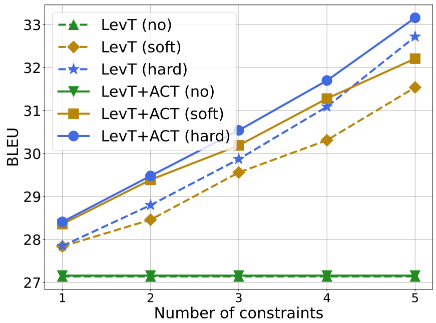


Figure 3: Ablation results of constrained translation with one-to-multiple constraints.

Are improvements by ACT robust against constraints of different frequencies? Given terminology constraints in the samples, we sort them by (averaged) frequency and evenly divide the corresponding data samples into *high*, *medium* and *low* categories.The results on translation quality of each category for the En→De translation tasks are presented in Table 6. We find that LevT benefits mostly from ACT in the scenarios of lower-frequency terms for three datasets. Although, in some settings such as HIGH in WMT14-WIKT and MED in WMT17-WIKT, the introduction of ACT for constrained LevT seems to bring performance drops for those higher-frequency terms. Since terms from IATE are rarer than WIKT as in Table 3, the improvements brought by ACT are steady.

Are improvements by ACT robust against constraints of different numbers? In more practical settings, the number of constraints is usually more than one. To simulate this, we randomly sample 1-5 words from each reference as lexical constraints, and results are presented in Figure 3. We find that, as the number of constraints grows, the translation quality ostensibly becomes better for LevT with or without ACT. And ACT consistently brings extra improvements, indicating the help by ACT for constrained decoding in constrained NAT.

6.3 Limitations

Although the proposed ACT algorithm is effective to improve NAT models on constrained translation, we also find it does not bring much performance gain on translation quality (*i.e.*, BLEU) over the backbone LevT for unconstrained translation. The results on the full set of WMT14 En→De test set further corroborate this finding, which is shown in Appendix A.

Another limitation of our work is that we do not propose a new paradigm for constrained NAT. The purpose of this work is to enhance existing methods for constrained NAT, *i.e.*, editing-based iterative NAT methods, under rare lexical constraints. It would be interesting for future research to explore new ways to impose lexical constraints on NAT models, perhaps on non-iterative NAT.

Note that, machine translation in real scenario still falls behind human performance. Moreover, since we primary focus on improving constrained NAT, real applications calls for refinement in various aspects that we do not consider in this work.

7 Conclusion

In this work, we propose a plug-in algorithm (ACT) to improve lexically constrained non-autoregressive translation, especially under low-frequency constraints. ACT bridges the gap between training and constrained inference and prompts the context information of the constraints to the constrained NAT model. Experiments show that ACT improves translation quality and term preservation over the backbone NAT model Levenshtein Transformer. Further analyses show that the findings are consistent over constraints varied from frequency, TF-IDF, and lengths. In the future, we will explore the application of this approach to more languages. We also encourage future research to explore a new paradigm of constrained NAT methods beyond editing-based iterative NAT.

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.

Tamer Alkhouli, Gabriel Bretschner, and Hermann Ney. 2018. On the alignment problem in multi-head attention-based neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 177–185, Brussels, Belgium. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. 3rd International Conference on Learning Representations, ICLR 2015 ; Conference date: 07-05-2015 Through 09-05-2015.

Loïc Barrault, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joanis, Tom Kocmi, Philipp Koehn, Chi-kiu Lo, Nikola Ljubešić, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, and Marcos Zampieri. 2020. Findings of the 2020 conference on machine translation (WMT20). In *Proceedings of the Fifth Conference on Machine Translation*, pages 1–55, Online. Association for Computational Linguistics.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, and Marco Turchi. 2017. Findings of the 2017 conference on machine translation (WMT17). In *Proceedings of the Second Conference on Machine Translation*, pages 169–214, Copenhagen, Denmark. Association for Computational Linguistics.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

Guanhua Chen, Yun Chen, and Victor O.K. Li. 2021. Lexically constrained neural machine translation with explicit alignment guidance. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(14):12630–12638.

Guanhua Chen, Yun Chen, Yong Wang, and Victor O.K. Li. 2020. Lexical-constraint-aware neural machine translation via data augmentation. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 3587–3593. International Joint Conferences on Artificial Intelligence Organization. Main track.

Josep 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.

Liang Ding, Longyue Wang, Xuebo Liu, Derek F. Wong, Dacheng Tao, and Zhaopeng Tu. 2021. Understanding and improving lexical choice in non-autoregressive translation. In *International Conference on Learning Representations*.

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.

Cunxiao Du, Zhaopeng Tu, and Jing Jiang. 2021. Order-agnostic cross entropy for non-autoregressive machine translation. In *Proc. of ICML*.

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. 2019. Mask-predict: Parallel decoding of conditional masked language models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6112–6121, Hong Kong, China. Association for Computational Linguistics.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor O.K. Li, and Richard Socher. 2018. Non-autoregressive neural machine translation. In *International Conference on Learning Representations*.

Jiatao Gu, Changhan Wang, and Junbo Zhao. 2019. Levenshtein transformer. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Junliang Guo, Zhirui Zhang, Linli Xu, Hao-Ran Wei, Boxing Chen, and Enhong Chen. 2020. Incorporating bert into parallel sequence decoding with adapters. In *Advances in Neural Information Processing Systems*, volume 33, pages 10843–10854. Curran Associates, Inc.

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.

Lukasz Kaiser, Samy Bengio, Aurko Roy, Ashish Vaswani, Niki Parmar, Jakob Uszkoreit, and Noam Shazeer. 2018. Fast decoding in sequence models using discrete latent variables. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2390–2399. PMLR.

Jungo Kasai, James Cross, Marjan Ghazvininejad, and Jiatao Gu. 2020. Non-autoregressive machine translation with disentangled context transformer. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5144–5155. PMLR.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, Austin, Texas. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. Cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015.

Gyubok Lee, Seongjun Yang, and Edward Choi. 2021. Improving lexically constrained neural machine translation with source-conditioned masked span prediction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 743–753, Online. Association for Computational Linguistics.

Vladimir I. Levenshtein. 1965. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet physics. Doklady*, 10:707–710.

Thang Luong, Ilya Sutskever, Quoc Le, Oriol Vinyals, and Wojciech Zaremba. 2015. Addressing the rare word problem in neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 11–19, Beijing, China. Association for Computational Linguistics.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. 2018. Scaling neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 1–9, Brussels, Belgium. Association for Computational Linguistics.

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, Philadelphia, Pennsylvania, USA. 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.

Lihua Qian, Hao Zhou, Yu Bao, Mingxuan Wang, Lin Qiu, Weinan Zhang, Yong Yu, and Lei Li. 2021. Glancing transformer for non-autoregressive neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1993–2003, Online. Association for Computational Linguistics.

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, Berlin, Germany. Association for Computational Linguistics.

Raphael Shu, Jason Lee, Hideki Nakayama, and Kyunghyun Cho. 2020. Latent-variable non-autoregressive neural machine translation with deterministic inference using a delta posterior. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8846–8853.

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. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):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.

Mitchell Stern, William Chan, Jamie Kiros, and Jakob Uszkoreit. 2019. Insertion transformer: Flexible sequence generation via insertion operations. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5976–5985. PMLR.

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.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, Istanbul, Turkey. European Language Resources Association (ELRA).

Ferhan Ture, Douglas W. Oard, and Philip Resnik. 2012. Encouraging consistent translation choices. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 417–426, Montréal, Canada. Association for Computational Linguistics.

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

Yiren Wang, Fei Tian, Di He, Tao Qin, ChengXiang Zhai, and Tie-Yan Liu. 2019. Non-autoregressive machine translation with auxiliary regularization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):5377–5384.

Weijia Xu and Marine Carpuat. 2021. EDITOR: An Edit-Based Transformer with Repositioning for Neural Machine Translation with Soft Lexical Constraints. *Transactions of the Association for Computational Linguistics*, 9:311–328.

Weijia Xu, Shuming Ma, Dongdong Zhang, and Marine Carpuat. 2021. How does distilled data complexity impact the quality and confidence of non-autoregressive machine translation? In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4392–4400, Online. Association for Computational Linguistics.

Chunting Zhou, Jiatao Gu, and Graham Neubig. 2020. Understanding knowledge distillation in non-autoregressive machine translation. In *International Conference on Learning Representations*.

A Results on Full Test Set of WMT14 (En→De)

We extend the experiment on WMT14 En→De task to the full test set (3,003 samples) in Table 7. Following Susanto et al., we report results on both the filtered test set for sentence pairs that contain at least one target constraint ("Con.", 454 sentences) and the full test set ("Full", 3,003 sentences), which contains samples that do not have lexical constraints. When trained on the full test set, term usage rate raises from 94.88% to 98.82% when trained with ACT under soft constrained decoding, but the BLEU score has marginal improvements. The conclusion is consistent with the experiments in the main body of the paper that LevT with ACT is not significantly better than LevT on unconstrained translation, though our main claim rests on the scenario of constrained NAT.

Model	**Term%**	**BLEU**	
		Full (3,003)	Con. (454)
LevT†	**79.53**	**26.95**	29.95
+ ACT†	77.17	26.93	**30.35**
LevT w/ *soft*	94.88	27.04	30.38
+ ACT	**98.82**	**27.06**	**31.08**
LevT w/ *hard*	100.00	27.06	30.49
+ ACT	100.00	**27.07**	**31.11**

Table 7: Experiments on the test set of WMT14 En→De task, which shares the same domain of training set. Following Susanto et al. (2020), "Con." is the subset of WMT14-Full as shown in Table 3, where every sample has at least one lexical term as constraint.

B Case Study

The case study of LevT and LevT with ACT is presented in Table 8. In the case of unconstrained or soft constrained translation, LevT incorrectly translates low frequency constraint words (*e.g.*, *Hühnerfeiern* in case 1). In the case of hard constrained translation, LevT tends to have more interfering words around the constraint words (*e.g.*, *sind* in case 1). After incorporating ACT, we witness consistent improvements in the translation of the constraints for LevT, especially for soft constrained translation where it successfully translates given constraints. However, when the translation is not constrained on lexical terms (*i.e.*, unconstrained translation), LevT with ACT still struggles at translating the term correctly (both case 1 and 2).

Case 1
Source However, carriages are also popular for hen parties, he commented.
Target Kutschen sind aber auch für Jungesellinnenabschiede beliebt, meint er.
Terminology Constraints hen parties → Jungesellinnenabschiede
LevT
Unconstrained translation Kutschen sind aber auch für *Hühnerfeiern* beliebt, kommentierte er. ⇒ *wrong term*
Soft constrained translation Allerdings sind auch für *Hinnenabschiebeliebt*, kommentierte er. ⇒ *wrong term*
Hard constrained translation Aber Auch für *Jungesellinnenabschiede* sind beliebt, sagte er. ⇒ *incomplete sentence*
LevT + ACT
Unconstrained translation Wagen sind aber auch für *Hühnerpartys* beliebt, kommentierte er. ⇒ *wrong term*
Soft constrained translation Kutschen sind aber auch für *Jungesellinnenabschiede* beliebt, sagte er.
Hard constrained translation Kutschen sind aber auch für *Jungesellinnenabschiede* beliebt, sagte er.
Case 2
Source The media also reported that several people injured.
Target Medien berichteten außerdem von mehreren Verletzten.
Terminology Constraints injured → Verletzten
LevT
Unconstrained translation Die Medien berichteten auch, dass mehrere Menschen *verletzt* wurden. ⇒ *wrong term*
Soft constrained translation Die Medien berichteten auch, dass mehrere *Verletzte* wurden. ⇒ *wrong term*
Hard constrained translation Die Medien berichteten auch, dass mehrere *Verletzte* wurden. ⇒ *wrong term*
LevT + ACT
Unconstrained translation Die Medien berichteten auch, dass mehrere Menschen *verletzt* wurden. ⇒ *wrong term*
Soft constrained translation Die Medien berichteten auch, dass mehrere *Verletzten*.
Hard constrained translation Die Medien berichteten auch, dass mehrere *Verletzten*.

Table 8: Case study of LevT and LevT with ACT. Text in brown denotes the constraint word, text in red denotes the translation error of constraints, and ⇒ denotes analysis of the translation errors.