Towards Accurate Translation via Semantically Appropriate Application of Lexical Constraints

Yujin Baek[*◇], **Koanho Lee**[*◇], **Dayeon Ki**[♣]
Cheonbok Park[♡], **Hyoung-Gyu Lee**[♡] **and Jaegul Choo**[◇]
[◇]KAIST ,[♣]Korea University, [♡]Papago, NAVER Corp.
{yujinbaek, le5544, jchoo}@kaist.ac.kr
dayeonki@korea.ac.kr, {cbok.park,hg.lee}@navercorp.com

Abstract

Lexically-constrained NMT (LNMT) aims to incorporate user-provided terminology into translations. Despite its practical advantages, existing work has not evaluated LNMT models under challenging real-world conditions. In this paper, we focus on two important but understudied issues that lie in the current evaluation process of LNMT studies. The model needs to cope with challenging lexical constraints that are "homographs" or "unseen" during training. To this end, we first design a homograph disambiguation module to differentiate the meanings of homographs. Moreover, we propose PLUMCOT, which integrates contextually rich information about unseen lexical constraints from pre-trained language models and strengthens a copy mechanism of the pointer network via direct supervision of a copying score. We also release HOLLY, an evaluation benchmark for assessing the ability of a model to cope with "homographic" and "unseen" lexical constraints. Experiments on HOLLY and the previous test setup show the effectiveness of our method. The effects of PLUMCOT are shown to be remarkable in "unseen" constraints. Our dataset is available at https://github.com/papago-lab/HOLLY-benchmark.

1 Introduction

Lexically-constrained neural machine translation (LNMT) is a task that aims to incorporate pre-specified words or phrases into translations (Hokamp and Liu, 2017; Dinu et al., 2019; Song et al., 2019; Susanto et al., 2020; Xu and Carpuat, 2021a; Chen et al., 2021a,b; Wang et al., 2022b, inter alia). It plays a crucial role in a variety of real-world applications where it is required to translate pre-defined source terms into accurate target terms, such as domain adaptation leveraging domain-specific or user-provided terminology. For example, as shown in Case A of Table 1, an LNMT model successfully translates the source term ("코로나") into its corresponding target term ("Covid-19") by adhering to a given lexical constraint ("코로나" → "Covid-19").

Despite its practicality, previous studies on LNMT have not evaluated their performances under challenging real-world conditions. In this paper, we focus on two important but understudied issues that lie in the current evaluation process of the previous LNMT studies.

***Semantics* of lexical constraints must be considered.** In previous work, at test time, lexical constraints are automatically identified from the source sentences by going through an *automatic* string-matching process (Dinu et al., 2019; Ailem et al., 2021; Chen et al., 2021b). For example, in Case B of Table 1, a source term ("코로나") in the bilingual terminology is present as a substring in the source sentence. Accordingly, its corresponding target term ("Covid-19") is automatically bound together as a lexical constraint ("코로나" → "Covid-19") without considering the semantics of the matched source term,[1] which can lead to a serious mistranslation. This automatic string-matching cannot differentiate textually identical yet semantically different source terms. Thus, the more accurately the LNMT reflects the lexical constraint, the more pronounced the severity of the homographic issue is. To address this homograph issue, LNMT systems must be equipped to understand the semantics of identified lexical constraints and determine whether or not these constraints should be imposed.

***Unseen* lexical constraints need to be examined.** One desideratum of LNMT systems is their robustness to handle "unseen" lexical constraints, thereby responding to random, potentially neologistic, or technical terms that users might bring up. However,

[*] Equal Contribution

[1] Here, 코로나 in Case B indicates Corona, a brand of beer produced by a Mexican brewery.

Findings of the Association for Computational Linguistics: ACL 2023, pages 4839–4855

Bilingual Terminology	
Source Term	**Target Term**
선별진료소	Testing Center
코로나	Covid-19
⋮	⋮

(Case A) Semantically Relevant Lexical Constraint	
Source	코로나 이전 수준으로 경기가 완전히 회복하는 날이 올까요?
Lexical Constraint	코로나 → Covid-19 ✔ Automatically Retrieved from Bilingual Terminology
Translation	Will the economy ever fully recover to before Covid-19 levels? ✔

(Case B) Semantically Irrelevant Lexical Constraint	
Source	코로나 엑스트라는 1998년 이후 미국에서 가장 많이 팔린 수입 음료이다.
Lexical constraint	코로나 → Covid-19 ✗ Automatically Retrieved from Bilingual Terminology
Translation	~~Covid-19~~ Extra has been the top-selling imported drink in the U.S. since 1998. ✗ Corona

Table 1: Automatically retrieved lexical constraint.

in previous studies, a significant portion of the lexical constraints is exposed during training. Wang et al. (2022b) demonstrated the overlapped ratio of lexical constraints between the training and evaluation data (35.6% on average). Meanwhile, Zeng et al. (2022) also raises the issue of the high frequency of lexical constraints for test sets appearing in the training data.

When lexical constraints are included in the training examples, we find that a well-optimized vanilla Transformer (Vaswani et al., 2017) already satisfies lexical constraints by merely learning the alignment between the source and target terms co-occurring in the parallel training sentences.[2] This presents difficulties in identifying whether the presence of target terms in the output is attributed to the learned alignment, or the proposed components in previous studies. Therefore, it is important to control lexical constraints not exposed during training to examine the model's ability to cope with "unseen" lexical constraints.

As a response, we present a *test benchmark* for evaluating the LNMT models under these two critical issues. Our benchmark is specifically crafted not only to evaluate the performance of LNMT models but also to assess its ability to discern whether given lexical constraints are semantically appropriate or not. To the best of our knowledge, we are the first to release a hand-curated high-quality test benchmark for LNMT. Concurrently, we suggest a pipeline that allows researchers in LNMT communities to simulate realistic test conditions that consider the homograph issue and assign "unseen" lexical constraints.

To this end, we propose a *two-stage framework* to deal with these issues. We first develop a homograph disambiguation module that determines whether LNMT models should apply a given lexical constraint by evaluating its semantic appropriateness. Further, we propose an LNMT model that integrates provided lexical constraints more effectively by learning when and how to apply these lexical constraints. Our contributions are summarized as follows:

- We formulate the task of semantically appropriate application of lexical constraints and release a high-quality test benchmark to encourage LNMT researchers to consider real-world test conditions.
- We propose a novel homograph disambiguation module to detect semantically inappropriate lexical constraints.
- We present an LNMT model which shows the best translation quality and copy success rate in unseen lexical constraints.

2 HOLLY Benchmark

Here, we introduce HOLLY (**ho**mograph disambiguation evaluation for lexical**lly** constrained NMT), a novel benchmark for evaluating LNMT systems in two circumstances; either the assigned lexical constraints are semantically appropriate or not, as illustrated in Table 2. The entire test data includes 600 test examples on 150 Korean → English lexical constraints.

2.1 Test Examples

Each test example consists of three main elements, as presented in Table 2: (1) a lexical constraint (양수 → amniotic fluid), (2) a source sentence containing the source term (양수) of the lexical constraint, and (3) its reference translation.[3]

[2] We observe that the vanilla Transformer achieves a 66.67% copy success rate.

[3] We outsourced the translation process to a professional translation company, and each translation was manually reviewed.

Source Term		Test Example	Lexical Constraint
양수	(a)	Src. 양수 파열로 출산이 임박한 산모가 공군의 도움으로 건강한 아이를 출산했다. Ref. A pregnant woman on the verge of labor due to amniotic fluid breaking gave birth to a healthy child thanks to the help of the airforce.	양수 → amniotic fluid ✔
	(b)	Src. 평소 다니던 산부인과 의사 선생님이 양수 검사를 권해서 하고 왔습니다. Ref. As my regular gynecologist recommended an amniotic fluid test, I took the test and came back.	양수 → amniotic fluid ✔
	(c)	Src. 수학에서는 양수를 나타낼 때는 '+' 기호를 생략해도 되지만 음수를 나타낼 때에는 반드시 '-' 기호를 숫자 앞에 붙여야 한다. Ref. In mathematics, while you can omit the '+' symbol when indicating positive, you must mark the '-' one before numbers when meaning negative.	~~양수 → amniotic fluid~~ ✗
	(d)	Src. 상장 법인 대주주들의 주식 양도 양수가 최근 들어 활발한 것으로 나타났다. Ref. It turns out that recently, major shareholders of the listed corporates have active handovers and takeovers.	~~양수 → amniotic fluid~~ ✗

Table 2: Test examples are bound together with a lexical constraint. (a) and (b) are positive examples. (c) and (d) are negative examples. The source term can be pronounced as "yang-soo". Four examples are assigned to each lexical constraint.

Lexical Constraint	Positive Reference
양수 → amniotic fluid ✔	Src. 출산 예정일보다 일찍 양수가 터지는 경우가 있다. Ref. (There are cases where the amniotic fluid bursts sooner than the expected date of birth.)
	Src. 태아의 염색체 이상 여부를 알아보기 위해 양수를 검사했다. Ref. (The amniotic fluid was tested to find if there were any abnormalities with the fetal chromosomes.)

Table 3: Positive References. Homograph (양수) in positive references means the amniotic fluid.

While the source term is a homograph with multiple meanings, one of them is chosen to serve as its lexical constraint.[4] Then, based on the meaning of the source term in the source sentence, each test example is classified into one of two groups:

- ***Positive Example*** where the source term in its source sentence is semantically aligned to the lexical constraint (See test examples (a) and (b) in Table 2). For positive test examples, we expect lexical constraints should always be applied.
- ***Negative Example*** where the given lexical constraint is semantically improper to impose (See test examples (c) and (d) in Table 2). Negative test examples allow us to evaluate how LNMT models respond to inappropriate lexical constraints.

2.2 Positive References

As seen in Table 3, we provide two auxiliary source-side example sentences demonstrating the specific use of the source term of its lexical constraint, assuming that the meaning can be differentiated by the context used in the sentences rather than the terminology itself. Hereafter, we name these example sentences as *positive references*.

[4]Out of multiple different meanings of a homograph, we select the least frequent one as its lexical constraint. We describe the data construction details in Appendix A.

3 Methodology

Our methodology for semantically appropriate application of lexical constraints consists of two stages. Initially, we propose a *homograph disambiguation module* that can differentiate the semantics of lexical constraints. This module determines whether LNMT models should incorporate a lexical constraint or not. Subsequently, LNMT models, PLUMCOT in our case, perform the translation, either with or without the given lexical constraints.

3.1 Homograph Disambiguation

Given a few example sentences demonstrating how to specifically use a word, humans can infer the proper meaning. Likewise, our conjecture is that we can fulfill the homograph disambiguation task by leveraging these inter-sentential relationships.

3.1.1 Task Specification

Given n example sentences illustrating one specific meaning of a homograph, our *homograph disambiguation module* aims to determine whether the same word in a newly given sentence, denoted as 'New Sentence' in Fig. 1, carries the same meaning (**label: 1**) or not (**label: 0**). We conducted experiments with two example sentences (i.e., $n = 2$),[5] and the corresponding model architecture is described in Section 3.1.2.

[5]We experiment with $n = 1, 2,$ and 3. The effect of varying the number of example sentences is analyzed in Appendix C.2.

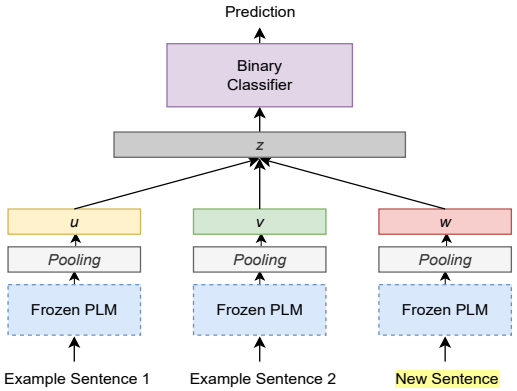

Figure 1: Structure of the homograph disambiguation module.

3.1.2 Model Architecture

Input Representations As illustrated in Fig. 1, sentence embeddings of example sentences and the new sentence are individually obtained from the PLM and fed into the classifier. Embedding vectors for all the sentences are extracted from the averaged hidden representations of the last K layers of frozen PLM.[6] Here, the embedding vector is obtained by the average of hidden representations for the tokens that make up a homograph within the sentence. We denote this averaging operation as *Pooling* in Fig. 1.

Binary Classifier Similar to Sentence-BERT (Reimers and Gurevych, 2019), we use the concatenation ($z \in \mathbb{R}^{6m+3}$) of the following as an input to the classifier:

- Contextualized representation of a homograph ($u, v, w \in \mathbb{R}^m$),
- element-wise difference for each pair ($|u-v|, |v-w|, |u-w| \in \mathbb{R}^m$,)
- pair-wise cosine similarity scores ($sim(u,v),\ sim(v,w),\ sim(u,w) \in \mathbb{R}$),

where m is the dimension of the embeddings and $sim(\cdot,\cdot)$ denotes the cosine similarity function. Our prediction $o \in [0,1]$ for a "New Sentence" is calculated as

$$o = \sigma(\max(0, zW_r + b_r)W + b), \qquad (1)$$

where $W_r \in \mathbb{R}^{(6m+3)\times m}$ and b_r are the weight matrix and bias vector of an intermediate layer, respectively. $W \in \mathbb{R}^{m\times 1}$ and b are the weight matrix and bias vector for the final prediction layer followed by $\sigma(\cdot)$, which represents the sigmoid function.

[6]We utilize the last 16 layers of *klue/roberta-large*, a RoBERTa-based PLM trained on Korean corpus. See https://huggingface.co/klue/roberta-large for details.

3.2 PLUMCOT

In this subsection, we introduce our LNMT model, PLUMCOT, which stands for leveraging pre-trained language model with direct supervision on a copying score for LNMT, and its detailed implementation. To better incorporate target terms into the translations, PLUMCOT combines LeCA (Chen et al., 2021b) with PLM and strengthens a pointer network with supervised learning of the copying score.

3.2.1 Problem Statement

Lexically-constrained NMT Suppose $X = (x_1, x_2, \cdots, x_{|X|})$ as a source sentence and $Y = (y_1, y_2, \cdots, y_{|Y|})$ as a target sentence. Given the constraints $C = (C_1, C_2, \cdots, C_n)$ where each constraint $C_i = (C_{i,S}, C_{i,T})$ consists of the source term $C_{i,S}$ and corresponding target term $C_{i,T}$, LNMT aims to incorporate $C_{1:n,T}$ into its generation. The conditional probability of LNMT can be defined as

$$p(Y|\ X, C; \theta) = \prod_{t=1}^{|Y|} p(y_t | y_{<t}, X, C;\ \theta). \qquad (2)$$

Input Data As in Chen et al. (2021b), we modify X as $\hat{X} = (X, \text{<sep>}, C_{1,T}, \cdots, \text{<sep>}, C_{n,T})$ by appending <sep> tokens followed by target terms, as illustrated in Table 4.[7] If there are no lexical constraints, a source sentence remains the same, i.e., $\hat{X} = X$.[8] Combining a source sentence with target terms leads to the modification of Eq. (2) as the following:

$$p(Y|\ X, C; \theta) = \prod_{t=1}^{|Y|} p(y_t | y_{<t}, \hat{X};\ \theta). \qquad (3)$$

3.2.2 Integration of PLM

As PLM such as BERT (Devlin et al., 2019) is trained on large amounts of unlabeled data, leveraging PLM for LNMT can provide rich contextualized information of X, even in controlled unseen lexical constraint scenarios.

We first feed the source sentence X to a frozen PLM to obtain a representation B of a source sentence, where B is the output of the last layer of the PLM. Conversely, our NMT model based on Vaswani et al. (2017) receives a modified source sentence $\hat{X}$ as input.

[7]In our training, we randomly sample target terms from the target sentence. Please refer to Appendix E for details.

[8]At test time, we append target terms only when lexical constraints are determined to be used by the homograph disambiguation module (as indicated in Table 4).

(Lexical Constraint) 코로나 → Covid-19 ✓Approved by homograph disambiguation module	
Source Sentence	코로나 이전 수준으로 경기가 완전히 회복하는 날이 올까요? (Will the economy ever fully recover to before **Covid-19** levels?)
Modified Source Sentence	코로나 이전 수준으로 경기가 완전히 회복하는 날이 올까요? <sep> **Covid-19**

(Lexical Constraint) 코로나 → Covid-19 ✗NOT approved by homograph disambiguation module	
Source Sentence	코로나 엑스트라는 1998년 이후 미국에서 가장 많이 팔린 수입 음료이다. (Corona Extra has been the top-selling imported drink in the U.S. since 1998.)
Modified Source Sentence	코로나 엑스트라는 1998년 이후 미국에서 가장 많이 팔린 수입 음료이다.

Table 4: Input modification. Expected target terms are appended to the end of the source sentence. The source term can be pronounced as "co-ro-na".

Let L denote the number of encoder and decoder layers of NMT, H^l be the output of the encoder of NMT at the l-th layer, and h_t^l denote the t-th element of H^l. For each layer $l \in [1, L]$, we employ multi-head attention with the output of PLM as in Zhu et al. (2019), denoted as MHA_B. This maps the output of the NMT encoder at $l-1$th layer into queries and output of PLM, B, into keys and values.[9] The output of the t-th element of the NMT encoder at the l-th layer is given by

$$\begin{aligned}\tilde{h}_t^l &= \frac{1}{2}\big(\text{MHA}(h_t^{l-1}, H^{l-1}, H^{l-1}) \\ &\quad + \text{MHA}_\text{B}(h_t^{l-1}, B, B)\big) + h_t^{l-1}, \\ h_t^l &= \text{LN}\big(\text{FFN}(\text{LN}(\tilde{h}_t^l)) + \tilde{h}_t^l\big),\end{aligned} \tag{4}$$

where $\text{LN}(\cdot)$ denotes Layer normalization in Ba et al. (2016) and MHA and $\text{FFN}(\cdot)$ are the multi-head attention and feed-forward network, respectively.

Similar to the encoder, multi-head attention with PLM is introduced for each decoder layer.[10] Combined with Section 3.2.3, a highly contextualized representation is given to the pointer network.

3.2.3 Supervision on a Copying Score

Pointer Network To copy target terms from $\hat{X}$, we introduce a pointer network (Gu et al., 2016) as in Song et al. (2019); Chen et al. (2021b). For each time step, a pointer network takes in the output of the encoder and outputs a copying score $g_t^{\text{copy}} \in [0, 1]$, which controls how much to copy. The output probability of the target word y_t can be calculated as

$$p(y_t|y_{<t}, \hat{X}; \theta) = (1 - g_t^{\text{copy}}) \times p_t^{\text{word}} + g_t^{\text{copy}} \times p_t^{\text{copy}}, \tag{5}$$

where p_t^{copy} is a probability of copying, and p_t^{word} is a probability of the target word y_t in the vocabulary.[11]

Copying Score As implied by Eq. (5), inaccurately predicted g_t^{copy} results in the failure of copying target terms. However, in previous research on LNMT, the importance of a copying score was relatively understated. Despite the high probability of copying p_t^{copy}, an incorrect copying score can even lower the output probability of the target terms. Therefore, we propose a novel supervised learning of the copying score g_t^{copy} to obtain a more accurate value.

Our supervision of the copying score strengthens the copy mechanism of the pointer network by allowing the model to learn exactly when to copy. Since target terms are in the source sentence, we can determine which words should be copied from the source sentence. For example, when translating a source sentence in Table 4, the appended target term, Covid-19, must be copied. Thus, the copying score g_t^{copy} of the target term Covid-19 should be higher, and g_t^{copy} should be lower for the remaining words in the target sentence. Our training objective can be defined as

$$\begin{aligned}L(\theta) &= -\sum_{t=1}^{|Y|} \log p(y_t|y_{<t}, \hat{X}; \theta) - \lambda J(\theta), \\ J(\theta) &= \alpha \sum_{t \notin C_{1:n,T}} (1 - g_t) \times \log\,(1 - g_t^{\text{copy}}) \\ &\quad + \beta \sum_{t \in C_{1:n,T}} g_t \times \log\, g_t^{\text{copy}},\end{aligned} \tag{6}$$

where a gold copying score g_t is set to zero for $t \in \{t|y_t \notin C_{1:n,T}\}$; otherwise, g_t is set to one for $t \in \{t|y_t \in C_{1:n,T}\}$. To mitigate the length imbalance between the target terms and remaining words in the target sentence, we set α and β to the

[9] Please refer to Appendix D for more details.
[10] Please refer to Appendix F for more details.
[11] Please refer to Appendix G for more details.

value obtained by dividing their respective lengths from the total length.

4 Experiments on the HOLLY benchmark

In this section, we report the performance of our methodology when tested on the HOLLY benchmark. In Section 4.1, we evaluate the performance of our *homograph disambiguation module* in determining the semantic appropriateness of a lexical constraint. In Section 4.2, we assess the performance of LNMT models using ***positive examples*** from the HOLLY benchmark under conventional settings. Subsequently, we investigate the potential advantages that the homograph disambiguation module might bring when applied to the ***negative examples*** from the HOLLY benchmark.

4.1 Homograph Disambiguation

4.1.1 Data

Here, we present our dataset for training the homograph disambiguation module. Our training data was collected from the Korean dictionary[12] and we manually inspected the quality of each sentence. In line with Fig. 1, each example consists of a triplet of example sentences containing a common homograph. Depending on the inter-sentential relationships between each input sentence, the homograph disambiguation module outputs a binary label: "1" is assigned if the homograph carries the same meaning in all sentences, and "0" if used differently in one example sentence. The brief data statistics of the training data are reported in Table 5. Note that any homographs are not allowed to be overlapped across train, validation, and test datasets.

	# of words (homograph)	# of examples		
		Class 1	Class 0	Total
Train	434	13,128	35,708	48,836
Validation	39	1,500	1,500	3,000

Table 5: Data statistics for the homograph disambiguation task.

At test time, we evaluated our model on the HOLLY benchmark. Specifically, for each lexical constraint, two positive references (refer to Table 3) and one of the four test example sentences ((a), (b), (c), or (d) in Table 2) is given as a triplet.

4.1.2 Results

We conducted experiments with two well-known variants of PLM trained on Korean corpora: *klue/roberta-base*, and *klue/roberta-large*. Our homograph disambiguation module achieved a test accuracy of 88.7%, and 92.3% when using *klue/roberta-base*, and *klue/roberta-large*, respectively. In spite of the imbalanced data distribution shown in Table 5, the values of precision and recall are balanced in both classes, as shown in Table 6.

Class	Precision	Recall	F1
1	0.924	0.933	0.929
0	0.933	0.923	0.928

Table 6: Test accuracy of homograph disambiguation module leveraging *klue/roberta-large* on the HOLLY benchmark. The accuracy is reported in terms of precision, recall, and F1 on each class.

4.2 Lexically-constrained NMT

4.2.1 Training Data

We used 1.83M sentence pairs from two publicly available Korean-English datasets as training corpora: IWSLT 17 training data and AI Hub parallel data.[13] We pre-tokenized the Korean corpora with Mecab and built a joint vocabulary for both languages by learning a Byte Pair Encoding (Sennrich et al., 2016) model in sentencepiece (Kudo and Richardson, 2018) with 32K merge operations.

To simulate the unseen lexical constraints, we filtered out about 160K training sentence pairs with test lexical constraints on both sides when experimenting with the HOLLY benchmark. This filtering process is crucial for examining how the models cope with any lexical constraints that users might introduce.

4.2.2 Evaluation

Metrics We evaluated the performance of our model in terms of BLEU[14] and copy success rate (CSR). CSR is a metric for investigating the ratio of imposed lexical constraints met in translations. For a statistical significance test, we use `compare-mt` (Neubig et al., 2019) with $p = 0.05$ and 1,000 bootstraps.

Test Scenarios There were two important test cases, as shown in Table 7. Given a source sentence, we can consider the ***Soft Matching*** test case,

[12]To release data, we collected the examples that are controlled by the appropriate license. CC BY-SA 2.0 KR.

[13]The AI HUB data can be found here: https://www.aihub.or.kr/aihubdata/data/view.do?currMenu=115&topMenu=100&aihubDataSe=realm&dataSetSn=126.

[14]We measure the BLEU scores using sacreBLEU (Post, 2018) with the signature nrefs:1|case:mixed|eff:no|tok:13a|smooth:exp|version:2.0.

Test Case	Lexical Constraint	Test Example	Expected Target Term(s)
Soft Matching	소화 → digest	Src. 사람의 이는 음식물을 잘게 부숴 삼키기 좋게 하여 소화를 돕는 역할을 한다. Ref. The human teeth function to break down food items into comfortably swallowable pieces, helping digestion.	digest | digestion | digestive
Hard Matching	소화 → digestion		digestion

Table 7: Two test scenarios are described with one of test example in the HOLLY benchmark. The source term can be pronounced as "so-hwa".

which allows some morphological variations, as introduced in Dinu et al. (2019). As illustrated in Table 7, since the Korean word 소화 can be used in multiple different forms via inflection, any one of the expected candidates (digest, digestion, and digestive) presented in the translation is considered to be correct in terms of CSR.

We also have the ***Hard Matching***[15] test case where the exact target term (e.g., digestion) presented in its reference has to be incorporated in the translation. Note that, this cannot be tested on negative examples since the target terms in lexical constraints do not appear in their references.

Baselines

- Code-Switching (CS) (Song et al., 2019) replaces source terms with aligned target terms and learns to copy them via pointer network.
- LeCA (Chen et al., 2021b) modifies the source sentence as described in Table 4, and utilizes pointer network during training.
- Cdalign (Chen et al., 2021a) proposes constrained decoding based on alignment.[16]

4.2.3 Main Results

Simulating Unseen Lexical Constraints Table 8 shows the importance of simulating unseen lexical constraints. When lexical constraints are exposed during training, the vanilla Transformer already achieves 66.67% of CSR by mere memorization. We observe that eliminating around 160K overlapping training examples results in a significant reduction of CSR (66.67% → 11.97%), indicating that we manage to simulate the conditions where lexical constraints are nothing short of unseen.

Results on Positive Examples The performances of LNMT models on positive examples[17] are compared in Table 9. It is shown that PLUMCOT outperforms all the baselines in both metrics by a large margin. Since we simulate the unseen lexical constraints, the external information from the PLM contributes to the increase in BLEU. Combined with the supervision on a copying score, PLUMCOT achieves the highest CSR.[18] The overall BLEU scores of *Hard Matching* are shown to be greater than *Soft Matching* as the expected target terms drawn from *reference translations* are given to the models.

Method	Soft Matching		Hard Matching	
	BLEU	CSR (%)	BLEU	CSR (%)
Trained w/ *filtered* data	18.44	11.97	18.44	9.06
Trained w/ *full* data	21.65	66.67	21.65	58.90

Table 8: Performance of the vanilla Transformer on positive examples. Without filtering, lexical constraints can be memorized by the network during training.

Method	Soft Matching		Hard Matching	
	BLEU	CSR (%)	BLEU	CSR (%)
CS	18.52	93.20	20.21	92.56
LeCA	19.33	94.17	20.53	93.85
Cdalign	17.02	95.47	17.52	94.82
PLUMCOT (Ours)	**20.91***	**98.06**	**22.07***	**98.71**

Table 9: LNMT performances on positive examples. All the models are trained with the *filtered* data as stated in Section 4.2.1. "$\star$" demonstrates that our method achieves statistically significant performance over baselines on positive examples.

Benefits of Homograph Disambiguation Here, we analyze beneficial effects of homograph disambiguation on LNMT. Since lexical constraints in negative examples[19] are semantically improper, the homograph disambiguation module determines whether they should be imposed or not. Corrections are made with its decisions; the corresponding lexical constraints are removed. The effects of the *correction* are shown in Fig. 2.

We observed significant drops in CSR across all of the models, which is desirable since the lex-

[15] This test case is suggested by previous work (Chen et al., 2021b; Song et al., 2019; Chen et al., 2021a).

[16] We compare our model to the ATT-INPUT approach, which suffers from high time complexity but guarantees a high CSR.

[17] Recall that positive examples are bound together with semantically appropriate lexical constraints. Refer to (a) and (b) in Table 2 for details.

[18] Instead of using the HOLLY benchmark, we also tested the performance of PLUMCOT in a test benchmark used in previous studies (Chen et al., 2021a,b) to compare its effectiveness, as shown in Appendix C.3.

[19] Refer to examples (c) and (d) in Table 2.

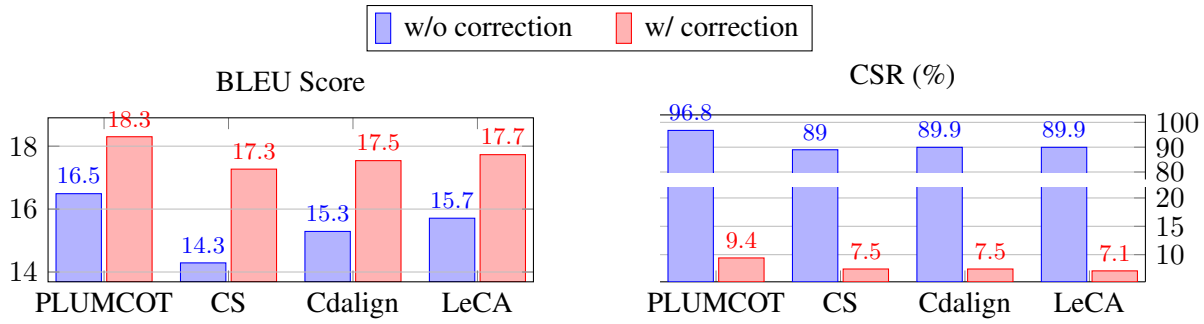

Figure 2: Effect of homograph disambiguation tested on negative examples. "w/ correction" refers to the removal of semantically inappropriate lexical constraints determined by the homograph disambiguation module. CSR was evaluated on *Soft Matching*.

ical constraints are irrelevant to the context. By removing inappropriate constraints, all the models achieve a consistent and statistically significant improvement in translation quality by a large margin.

4.2.4 Ablation Study

We study the effect of each component of PLUMCOT, and the results are provided in Table 10. Compared to the PLUMCOT without supervision, the supervision on a copying score significantly improves CSR (93.85% vs. 98.06%). We find that leveraging rich contextual representation of PLM can further improve the translation quality (18.51 → 20.91). The BLEU score of a model that combines only PLM without supervision on a copying score is lower than that of the PLUMCOT model. This may simply be due to a higher BLEU score from better reflecting the target terms in the positive examples (an increase in CSR from 93.85% to 98.06%). Combining the two components yields the best performance in both metrics. More ablations can be found in Appendix C.

Method	Soft Matching		Hard Matching	
	BLEU	CSR (%)	BLEU	CSR (%)
PLUMCOT	**20.91**	**98.06**	**22.07**	**98.71**
(−) PLM	18.51	98.06	19.58	98.38
(−) Supervision	19.22	93.85	20.54	93.20

Table 10: Ablation studies performed on positive examples *w/o* correction. "PLM": integration of PLM. "Supervision": supervised learning of a copying score.

4.3 Qualitative Analysis

Table 11 provides translated examples. Given a lexical constraint, PLUMCOT incorporates the target term correctly. In a negative example, the meaning of 세제 is properly translated into detergent by PLUMCOT with *correction*.[20] We provide more examples in Table 15.

5 Related Work

5.1 Lexically-constrained NMT

Recent work on LNMT broadly falls into two categories: *decoding algorithms* and *inline annotation*. During beam search, decoding algorithms enforce target terms to appear in the output (Hokamp and Liu, 2017; Anderson et al., 2017; Chatterjee et al., 2017; Hasler et al., 2018). This approach ensures a high CSR, but the decoding speed is significantly degraded. To alleviate this issue, Post and Vilar (2018) suggests a decoding algorithm with a complexity of $O(1)$ in the number of constraints. Another variation on decoding algorithms utilizes word alignments between source and target terms (Song et al., 2020; Chen et al., 2021a).

In *inline annotation* studies, the model is trained to copy target terms via modification of training data. Either a source term is replaced with the corresponding target term, or the target term is appended to the source sentence (Song et al., 2019; Dinu et al., 2019; Chen et al., 2021b). Concurrently, Bergmanis and Pinnis (2021); Niehues (2021); Xu and Carpuat (2021b) consider the morphological inflection of lexical constraints during the integration of target terms. While these methods incur a slight computational cost and provide better translation quality, target terms are not guaranteed to appear (Chen et al., 2021a; Wang et al., 2022a). To better copy target terms in a source sentence, a pointer network (Vinyals et al., 2015; Gulçehre et al., 2016) that uses attention weights to copy elements from a source sentence is intro-

[20] Note that the correction is made by homograph disambiguation.

Positive Example (Lexical Constraint: 세제 → tax system)	
Source	거래세를 줄이고 보유세를 강화하는 게 부동산 세제의 대원칙이지만 이를 적용하기도 어렵다.
Reference	Reducing transaction taxes and raising possession taxes are the core principles of the real estate tax system, but it is challenging to get them applied.
Vanilla	Reducing transaction taxes and strengthening holding taxes are the grand principles of real estate taxes, but it is also difficult to apply them. ✗
LeCA	Reducing transaction taxes and strengthening holding taxes are the main principles of real estate tax, but it is difficult to apply them. ✗
PLUMCOT	The main principle of the real estate tax system is to reduce transaction taxes and strengthen holding taxes, but it is difficult to apply them. ✓

Negative Example (Lexical Constraint: 세제 → tax system)	
Source	이번 행사 기간 동안 5만 원 이상 구입하시는 고객에게는 주방 세제를 경품으로 드립니다.
Reference	For those who spend more than 50 thousand won for purchasing items during this event, kitchen detergents will be given as a gift.
PLUMCOT w/o correction	The kitchen tax system will be given as a prize to customers who purchase more than 50,000 won during this event.
PLUMCOT w/ correction	Customers who purchase more than 50,000 won during this event will receive a gift of kitchen detergent.

Table 11: Example translations for positive and negative examples. The source term can be pronounced as "se-je".

duced (Gū et al., 2019; Song et al., 2019; Chen et al., 2021b). In this work, we further enhance the copying mechanism of a pointer network via supervised learning of a copying score that achieves better performance in terms of BLEU and CSR.

5.2 Homograph Issue in LNMT

Michon et al. (2020) points out the homograph issue in LNMT in an in-depth error analysis of their model. To the best of our knowledge, the homograph issue was explicitly addressed first in Öz and Sukhareva (2021). In their work, given a source homographic term, the most frequent alignment is selected as its correct lexical constraint, while the other alignments are treated as negative terms that should be avoided in the translation. However, low-frequency meanings are important for LNMT since it is not guaranteed that users always bring up generic terminology.

Different from their method, our homograph disambiguation module infers the meaning of lexical constraints and makes decisions to impose them or not. Furthermore, we confirm that our method works equally well on "unseen" homographs.

5.3 Integration of PLM with NMT

Followed by the success of PLM, researchers attempted to distill the knowledge of PLM into NMT (Zhu et al., 2019; Weng et al., 2020; Xu et al., 2021). BERT-fused (Zhu et al., 2019) is one such method; it plugs the output of BERT into the encoder and decoder via multi-head attention. We borrowed the idea from BERT-fused, and for the first time, combined LNMT and PLM, which works well even in "unseen" lexical constraints by leveraging the rich contextual information of PLM.

6 Conclusions

In this paper, we investigate two unexplored issues in LNMT and propose a new benchmark named HOLLY. To address the homograph issue of the source terms, we built a homograph disambiguation module to infer the exact meaning of the source terms. We confirm that our homograph disambiguation module alleviates mistranslation led by semantically inappropriate lexical constraints. PLUMCOT is also proposed to improve LNMT by using the rich information of PLM and ameliorating its copy mechanism via direct supervision of a copying score. Experiments on our HOLLY benchmark show that PLUMCOT significantly outperforms existing baselines in terms of BLEU and CSR.

7 Limitations

Our study includes some limitations that must be addressed. Some test examples might have wrong predictions made by the *homograph disambiguation module*. Specifically, in positive examples where lexical constraints should be imposed, its errors result in wrong corrections (i.e., the elimination of necessary lexical constraints). Table 12 shows how these erroneous corrections affect the results.

Method	w/o correction		w/ correction	
	BLEU	CSR (%)	BLEU	CSR (%)
CS	18.52	93.20	18.88	85.44
LeCA	19.33	94.17	19.24	88.03
Cdalign	17.02	95.47	17.03	89.97
PLUMCOT (Ours)	**20.91**	**98.06**	**20.80**	**91.59**

Table 12: Effect of homograph disambiguation tested on positive examples on ***Soft Matching***.

We can observe an overall decline in CSR; however, it does not hurt the translation quality. We

verify that the differences in BLEU resulting from wrong corrections are not statistically significant for all the methods. Considering the gain achieved in negative examples, as seen in Fig. 2, our proposed homograph disambiguation might serve as a useful starting point to address homographs in LNMT; however, there is still room for improvement. Our current *homograph disambiguation module* is designed as a stand-alone system outside the LNMT. However, building an end-to-end system can be beneficial, which can be addressed in future work.

Acknowledgments

Authors would like to thank all Papago team members for the insightful discussions. Also, we sincerely appreciate the fruitful feedback from Won Ik Cho and CheolSu Kim. We thank the anonymous reviewers for their valuable suggestions for enhancing this work. This work was supported by Papago, NAVER Corp, the Institute of Information communications Technology Planning Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2019-0-00075, Artificial Intelligence Graduate School Program (KAIST)), and the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. NRF-2022R1A2B5B02001913).

References

Melissa Ailem, Jingshu Liu, and Raheel Qader. 2021. Lingua custodia's participation at the wmt 2021 machine translation using terminologies shared task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 799–803.

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2017. Guided open vocabulary image captioning with constrained beam search. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 936–945, Copenhagen, Denmark. Association for Computational Linguistics.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

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.

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

Guanhua Chen, Yun Chen, and Victor OK Li. 2021a. Lexically constrained neural machine translation with explicit alignment guidance. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12630–12638.

Guanhua Chen, Yun Chen, Yong Wang, and Victor OK Li. 2021b. Lexical-constraint-aware neural machine translation via data augmentation. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 3587–3593.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186.

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.

Jetic Gū, Hassan S Shavarani, and Anoop Sarkar. 2019. Pointer-based fusion of bilingual lexicons into neural machine translation. *arXiv preprint arXiv:1909.07907*.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1631–1640.

Çağlar Gulçehre, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio. 2016. Pointing the unknown words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 140–149.

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.

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.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *EMNLP 2018*, page 66.

Elise Michon, Josep M 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.

Graham Neubig, Zi-Yi Dou, Junjie Hu, Paul Michel, Danish Pruthi, and Xinyi Wang. 2019. compare-mt: A tool for holistic comparison of language generation systems. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 35–41.

Jan Niehues. 2021. Continuous learning in neural machine translation using bilingual dictionaries. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 830–840.

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.

Ogün Öz and Maria Sukhareva. 2021. Towards precise lexicon integration in neural machine translation. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 1084–1095.

Sungjoon Park, Jihyung Moon, Sungdong Kim, Won Ik Cho, Ji Yoon Han, Jangwon Park, Chisung Song, Junseong Kim, Youngsook Song, Taehwan Oh, et al. 2021. Klue: Korean language understanding evaluation. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

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.

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.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. 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 3982–3992.

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.

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.

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.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. *Advances in neural information processing systems*, 28.

Shuo Wang, Peng Li, Zhixing Tan, Zhaopeng Tu, Maosong Sun, and Yang Liu. 2022a. A template-based method for constrained neural machine translation. *arXiv preprint arXiv:2205.11255*.

Shuo Wang, Zhixing Tan, and Yang Liu. 2022b. 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.

Rongxiang Weng, Heng Yu, Shujian Huang, Shanbo Cheng, and Weihua Luo. 2020. Acquiring knowledge from pre-trained model to neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9266–9273.

Haoran Xu, Benjamin Van Durme, and Kenton Murray. 2021. Bert, mbert, or bibert? a study on contextualized embeddings for neural machine translation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6663–6675.

Weijia Xu and Marine Carpuat. 2021a. 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 and Marine Carpuat. 2021b. Rule-based morphological inflection improves neural terminology translation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5902–5914.

Chun Zeng, Jiangjie Chen, Tianyi Zhuang, Rui Xu, Hao Yang, Qin Ying, Shimin Tao, and Yanghua Xiao. 2022. Neighbors are not strangers: Improving non-autoregressive translation under low-frequency lexical constraints. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5777–5790, Seattle, United States. Association for Computational Linguistics.

Jinhua Zhu, Yingce Xia, Lijun Wu, Di He, Tao Qin, Wengang Zhou, Houqiang Li, and Tieyan Liu. 2019. Incorporating bert into neural machine translation. In *International Conference on Learning Representations*.

A HOLLY benchmark

We collected monolingual example sentences that contain one of the pre-specified homographs from the Korean dictionary. For each homograph, retrieved example sentences are classified into multiple groups according to their meanings. We chose one group with the least frequent meaning and used its examples as positive references to determine a lexical constraint for the homograph. Examples from the other groups are considered as negative references. Eventually, six reference sentences were collected for each homograph; more specifically, four positive references and two negative references.

Setting aside two positive references for homograph disambiguation, as stated in Table 3, we outsourced the translation of two positive and negative examples, as introduced in Table 2. For positive examples, professional translators were requested to translate source terms of lexical constraints into pre-defined target terms. We guide the translators to carefully translate negative examples by focusing on the exact meaning of lexical constraints.

B Implementation details

B.1 Configuration of PLUMCOT

We implemented PLUMCOT and all the models based on fairseq (Ott et al., 2019). We matched the embedding dimensions, the number of layers, and the number of attention heads of all models for a fair comparisons. PLUMCOT was trained from scratch and *klue/roberta-large* (Park et al., 2021) was used for our PLM.[21]

B.2 Computational Cost

All the experiments were conducted on a single A100 GPU. It takes about 84 hours to train PLUMCOT and 5 hours to train the homograph disambiguation module. The number of training / total parameters for PLUMCOT is 156M and 493M. The number of training / total parameters for the homograph disambiguation module is 6M and 343M.

C Ablation Studies

C.1 Weights of the supervised learning of a copying score

The results in Table 13 were reported according to the different weights λ of the supervised learning of the copying score in Eq. (6). Based on experimental results, we were able to find a compromise where λ is 0.2.

Method	Soft Matching		Hard Matching	
	BLEU	CSR (%)	BLEU	CSR (%)
PLUMCOT (λ = 1)	19.00	99.68	20.49	99.35
PLUMCOT (λ = 0.2)	20.91	98.06	22.07	98.71

Table 13: Results of BLEU and CSR according to different λ.

C.2 Number of example sentences

We experimented with a varying number of example sentences. As we use more example sentences, the information from the inter-sentential relationships becomes richer, eventually improving homograph disambiguation performance. Experiments with n = 1, 2, and 3 show an accuracy of 91.33%, 92.33%, and 92.67%, respectively. Although an experiment with n = 3 provides the best accuracy, collecting positive references can sometimes be burdensome to users. Therefore, we conclude that n should be decided by considering its trade-off.

C.3 Randomly Sampled Test Constraints

Different from our HOLLY benchmark, at test time, lexical constraints were *randomly* sampled from the alignments in each sentence pair in previous studies (Dinu et al., 2019; Song et al., 2019; Chen et al., 2021b,a; Wang et al., 2022b). Ten different

[21] Please refer to 16 for more details.

test sets were built based on ten randomly sampled sets of lexical constraints, as described in Chen et al. (2021a,b). Test statistics are reported in Table 14. It is shown that PLUMCOT achieves the highest BLEU. The CSR is slightly lower than Cdalign, indicating that the gain for "seen" constraints is insignificant.[22]

Method	BLEU		CSR (%)	
	Average	STDEV	Average	STDEV
Vanilla	19.14	0.00	81.67	0.00
CS	20.95	0.13	94.66	0.00
LeCA	22.10	0.05	96.33	0.00
Cdalign	21.45	0.07	**98.03**	0.00
PLUMCOT (λ = 0.2)	**22.50**	0.07	97.84	0.00

Table 14: Results on randomly sampled test lexical constraints. Statistics are drawn from five randomly constructed test datasets.

D Equation Details

Let Q, K, and V be the query, key, and value in (Vaswani et al., 2017), respectively. Then MHA in Eq. (4) and Eq. (8) can be calculated as

$$\begin{aligned} \text{MHA}(Q,K,V) &= \text{Cat}_{h=1}^{H}[\text{head}_1,\cdots,\text{head}_n]W_o, \\ \text{head}_i &= \text{Attn}(QW_i^Q, KW_i^K, VW_i^V), \\ \text{Attn}(q,k,v) &= \text{softmax}(\frac{qk^T}{\sqrt{d_k}})v, \end{aligned} \tag{7}$$

where the projection matrices are parameters $W^O \in \mathbb{R}^{Hd_v \times d_{\text{model}}}$, W_i^Q, W_i^K, $W_i^V \in \mathbb{R}^{d_{\text{model}} \times d_k}$ for MHA. In this paper, we employ $d_{\text{model}} = 768$, $H = 12$, $d_k = d_v = d_{\text{model}}/H = 64$. Note that all baselines and our model PLUMCOT use the same number of heads and the same projection matrices size. We use additional multi-head attention, MHA_B, which only differs in projection matrices size where W_i^Q, W_i^K, $W_i^V \in \mathbb{R}^{d_{\text{PLM}} \times d_k}$ for MHA_B.[23]

E Input data augmentation

As illustrated in Table 4, we modify a source sentence X as $\hat{X}$ by appending <sep> tokens followed by target terms. Since lexical constraints are domain-specific or user-provided terminology, we exclude the top 1,500 frequent words from a 32K joint dictionary. In our training, we randomly sample at most 3 target terms from the target sentence. For each sentence, from 0 to 3 target terms are sampled following the distribution [0.3, 0.2, 0.25, 0.25].

F Integration of PLM in Decoder

Here, we follow the same notations in Section 3.2.2. Let S^l denote the decoder output at l^{th} layer. s_t^l is the t-th element of S^l, and $S_{1:t}^l$ denotes t number of elements from s_1^l to s_t^l, masking elements from $t+1$ to the end. The output of each layer of the decoder can be calculated as

$$\begin{aligned} \hat{s}_t^l &= \text{LN}\big(\text{MHA}(s_t^{l-1}, S_{1:t}^{l-1}, S_{1:t}^{l-1})\big) + s_t^{l-1}, \\ \tilde{s}_t^l &= \frac{1}{2}\Big(\text{MHA}(\hat{s}_t^l, H^L, H^L) \\ &\quad + \text{MHA}_\text{B}(\hat{s}_t^l, B, B)\Big) + \hat{s}_t^l, \\ s_t^l &= \text{LN}\big(\text{FFN}\big(\text{LN}(\tilde{s}_t^l)\big) + \tilde{s}_t^l\big). \end{aligned} \tag{8}$$

G Pointer Network

We use the same notations in Section 3.2 and Appendix D and F. Let $|\hat{X}|$ denotes the length of the modified source sentence $\hat{X}$, and $(\alpha_{t,1}, \alpha_{t,2}, \cdots, \alpha_{t,|\hat{X}|})$ denotes the averaged attention weight of $\text{MHA}(\hat{s}_t^L, H^L, H^L)$ over the multi-heads in Eq. (8). Then our copying score g_t^{copy} can be calculated as

$$\begin{aligned} g_t^{\text{copy}} &= \sigma(W_g[c_t; s_t^L] + b_g), \\ c_t &= \sum_{i=1}^{|\hat{X}|} \alpha_{t,i} \times h_i^L, \end{aligned} \tag{9}$$

where c_t and s_t^L are concatenated, W_g and b_g are the weight matrix and bias vector, and $\sigma(\cdot)$ is the sigmoid function.

[22] Note that the models are trained with *full* data as we cannot remove training examples that overlap with random test lexical constraints in advance.

[23] Here, we utilize *klue/roberta-large*, a RoBERTa-based PLM trained on Korean corpus. The size of d_{PLM} is 1024.

Positive Example (Lexical Constraint: 내성 ("nae-sung") → resistance)	
Source	항생제에 내성이 있는 새로운 종류의 병원균이 등장해서 국민의 건강을 위협하고 있다.
Reference	New types of pathogens with **resistance** to antibiotics have emerged, threatening public health.
Vanilla	A new type of pathogens that are tolerant of antibiotics have emerged, threatening the health of the people. ✗
LeCA	A new type of pathogen that is tolerant of antibiotics has emerged, threatening the health of the people. ✗
PLUMCOT	A new type of pathogen that has **resistance** to antibiotics has emerged, threatening the health of the people. ✓

Negative Example (Lexical Constraint: 내성 ("nae-sung") → resistance)	
Source	그가 말수가 적은 것은 내성적인 성격에서 연유한다.
Reference	His being quiet is because of his **introverted** personality.
PLUMCOT w/o correction	His low words are based on his **resistance** to introverts.
PLUMCOT w/ correction	His low-level words are related to his **introverted** personality.

Positive Example (Lexical Constraint: 사유 ("sa-yoo") → reason)	
Source	회사 측은 계약 당사자 간 계약의 절차성을 사유로 계약 무효를 결정했다고 설명했다.
Reference	The company explained that the contract cancellation was decided because of the **reason** relevant to contract procedures between the contract parties.
Vanilla	The company explained that it decided to nullify the contract because of the procedurality of the contract between the parties. ✗
LeCA	The company explained that it decided to nullify the contract because of the procedurality of the contract between the parties. ✗
PLUMCOT	The company explained that it decided to nullify the contract on the **reason** of the procedure of the contract between the parties. ✓

Negative Example (Lexical Constraint: 사유 ("sa-yoo") → reason)	
Source	자본주의 국가에서 사유 재산은 소유자의 의사에 따라 처분할 수 있다.
Reference	In capitalist countries, **private** assets can be disposed of according to the will of their owners.
PLUMCOT w/o correction	In capitalist countries, private property can be disposed of according to the **reason** of the owner.
PLUMCOT w/ correction	In capitalist countries, **private** property can be disposed of according to the owner's will.

Table 15: More translations for positive and negative examples.

Table 16: Hyperparameters and model configuration of PLUMCOT.

NMT	Transformer
encoder layers	6
encoder embed dim	768
encoder feed-forward dim	3072
encoder attention heads	12
decoder layers	6
decoder embed dim	768
decoder feed-forward dim	3072
decoder attention heads	12
positional encodings	Sinusoidal
max source positions	1024
max target positions	1024
segment embeddings	True
dropout	0.3

PLM	klue/roberta-large
encoder layers	24
encoder embed dim	1024
encoder feed-forward dim	4096
encoder attention heads	16
positional encodings	learned positional encodings
max source positions	514
max target positions	514
segment embeddings	True

Hyperparameter	Value
optimizer	Adamw
β_1, β_2	(0.9, 0.98)
weight decay	0.0
max updates	130k
learning rate	0.0005
learning rate warmup	4000 steps
warmup init learning rate	1e-7
lr scheduler	inverse sqrt
max tokens	4000
update frequency	8
clip grad norm	1.0

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section 7

☑ A2. Did you discuss any potential risks of your work?
Section 7

☑ A3. Do the abstract and introduction summarize the paper's main claims?
We summarized the paper's main claims both in the 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?

We created a test benchmark. Refer to Section 2 for more details.

☐ B1. Did you cite the creators of artifacts you used?
Not applicable. Left blank.

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Section 2

☑ 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)?
Section 2

☑ 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?
Section 2, Appendix

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Section 2

☑ 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.
Section 2, Section 4

C ☑ Did you run computational experiments?

Section 4

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

☒ 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?
We reported the results of just a single run.

☑ 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 4, Appendix

D ☑ Did you use human annotators (e.g., crowdworkers) or research with human participants?
Section 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.?
Appendix A

☒ 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)?
As we outsourced the translation process to a professional company, only qualified professional translators were participated. We guarantee that our institution paid sufficient amount of cost by signing a contract.

☑ 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?
Appendix A

☐ 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?
We manually review the data. Our data has nothing to do with demographic or geographic issues.