DICT DIS: Dictionary Constrained Disambiguation for Improved NMT

Ayush Maheshwari, Preethi Jyothi, Ganesh Ramakrishnan
Indian Institute of Technology Bombay, India
{ayusham, pjyothi, ganesh}@cse.iitb.ac.in

Abstract

Domain-specific neural machine translation (NMT) systems (e.g., in educational applications) are socially significant with the potential to help make information accessible to a diverse set of users in multilingual societies. Such NMT systems should be lexically constrained and draw from domain-specific dictionaries. Dictionaries could present multiple candidate translations for a source word/phrase due to the polysemous nature of words. The onus is then on the NMT model to choose the contextually most appropriate candidate. Prior work has largely ignored this problem and focused on the single candidate constraint setting wherein the target word or phrase is replaced by a single constraint. In this work, we present DICT DIS, a lexically constrained NMT system that disambiguates between multiple candidate translations derived from dictionaries. We achieve this by augmenting training data with multiple dictionary candidates to actively encourage disambiguation during training by implicitly aligning multiple candidate constraints. We demonstrate the utility of DICT DIS via extensive experiments on English-Hindi, English-German, and English-French datasets across a variety of domains including regulatory, finance, engineering, health and standard benchmark test datasets. In comparison with existing approaches for lexically constrained and unconstrained NMT, we demonstrate superior performance for the copy constraint and disambiguation-related measures on all domains, while also obtaining improved fluency of up to 2-3 BLEU points on some domains. We also release our test set consisting of 4K English-Hindi sentences in multiple domains. The source code is present at https://github.com/ayushbits/dictdis-multi.

1 Introduction

Neural machine translation (NMT) systems have seen great success in achieving state-of-the-art translations across several language pairs (Barraud et al., 2018). However, the default NMT pipeline does not guarantee the inclusion of specific terms in the translation output which is extremely crucial in domain-specific scenarios such as translation of technical content. While adding domain-specific terms has been relatively easier in phrase-based statistical MT, such an intervention poses a challenge in NMT owing to the difficulty of directly manipulating output representations from the decoder (Susanto et al., 2020). Alternatively, domain-specific NMT systems have been proposed to generate domain-aware translations by fine-tuning generic NMT models on domain-specific parallel text. However, such fine-tuning would require curating translation pairs for each domain, entailing significant human effort and increasing the cost of maintaining models trained separately for each domain.

Therefore, the MT output must adhere to the source domain by adopting domain-specific terminology, thus reducing and perhaps even guiding the translation post-editing effort. This is achieved in NMT via lexically constrained tech-
niques that incorporate pre-specified words and phrases in the NMT output (Hokamp and Liu, 2017; Dinu et al., 2019; Chen et al., 2021b). In addition to the source sentence, word or phrasal constraints in the target language are provided as input. The constraints could be derived either from (i) in-domain source-target dictionaries or (ii) user-provided source-target constraints during interactive machine translation. Often, such constraints could encode multiple potential translations for a given source phrase. For example, the word ‘speed’ can be translated into 4 different Hindi phrases teja, gati, raphiār, cāla in the physics domain. However, existing constrained translation approaches do not accommodate such ambiguity in the constraints.

In this work, we propose a lexically constrained disambiguation framework, DICTDIS, wherein we train the NMT model to choose the most appropriate word or phrase, from among multiple constraints that are provided for a given source word or phrase. In Figure 1, we present the output of our model in contrast to another lexically constrained approach (Leca (Chen et al., 2021b)) for a sample test instance.

DICTDIS is a copy-and-disambiguation method that accepts a source sentence and multiple source-target constraints as input and chooses an appropriate phrase from multiple candidate translations depending on the context. During training, we sample constraints from domain-agnostic dictionaries, in which, each source phrase could have multiple candidate translations. Toward this, we create a parallel corpus by appending each source sentence with multiple candidate translations delimited by appropriate separation symbols (c.f., Figure 2). The model is trained in a soft manner such that no constraint is forced to appear in the predicted sentence — the constraint injected should (i) be contextually relevant and (ii) not impact the fluency of the sentence. During inference, we show how DICTDIS effectively disambiguates among multiple candidate translations in domain-specific dictionaries specified at test-time (that were not available to the model during training). Our main contributions are as follows:

1) Given a source sentence and domain-specific dictionary constraints consisting of multiple source-target phrases, our proposed approach DICTDIS, can either pick the most relevant constraint or abstain from picking any constraint to least disrupt the fluency of the translation (c.f. Section 3.2).

2) DICTDIS is trained to do ‘soft’ disambiguation of phrases in the dictionary by automatically augmenting a domain agnostic corpus using synthetic domain-agnostic constraints (c.f. Section 3.3). Inspired by controlled text generation approaches (Pascual et al., 2021; Dathathri et al., 2019), we also introduce a user-controllable parameter in the decoding layer that can further improve the copy rate of constraints and provides a way of trading off the copy rate of constraints for fluency.

3) We present an extensive evaluation of DICTDIS on several datasets that we manually curated across finance, engineering, and regulatory domains as well as standard benchmarks (c.f. Section 4.2).

4) We compare against two state-of-the-art constrained approaches, viz., VDBA (Hu et al., 2019), Leca (Chen et al., 2021b) as well as an unconstrained baseline. We report results using the standard BLEU (Papineni et al., 2002) metric, and a semantic matching-based metric COMET (Rei et al., 2020). Further, we estimate constraint copying rate using copy success rate (CSR) and also analyze CSR for different polysemous degrees of constraints. We observe improvements in both CSR scores coupled with fluency improvement of up to 4 BLEU score points, unlike other baselines which forgo fluency while improving CSR (c.f. Section 5).

2 Related Work

Lexically constrained NMT approaches can be broadly divided into two categories: hard constraints and soft constraints. Hard-alignment methods copy lexical constraints in the exact same form and might therefore result in the ingestion of incorrect morphological forms (Hokamp and Liu, 2017; Post and Vilar, 2018). Moreover, it often results in lower BLEU scores on out-of-domain test sets (c.f. Section 5). In the hard constraints approaches, all constraints are guaranteed to appear in the output sentence. Chatterjee et al. (2022); Chen et al. (2021a) propose a mix of alignment and soft-constraint decoding by formulating the loss function as the joint probability of token probability from the decoder and alignment probability induced from intermediate transformer layers.

Soft constraint approaches modify the NMT training algorithm to bias token prediction probabilities without forcing constraints to appear in the output. Ri et al. (2021); Michon et al. (2020) replace the source word with placeholders, such as a named entity type or part-of-speech tag of the tar-
get constraint. Similarly, Song et al. (2019); Dinu et al. (2019) employ a bilingual dictionary to build a code-switching corpus by replacing each source phrase with the corresponding target constraint during training. However, this often interferes with the meaning of the source words, leading to poorer translation quality. (Baek et al., 2023; Chen et al., 2021b) propose a constraint-aware approach that bundles the source sentence with a single constraint using a separator token. The training data is constructed by aligning source and target sentences and randomly creating a constraint during training using the aligned pairs. D-LCNMT (Zhang et al., 2023) uses a 2-stage approach where the disambiguation module is separate from the ingestion module. It assumes that source lexicon has an additional context and attempts to align target candidates with the source lexicon. With this additional information, it uses contrastive learning to align the representation using source constraint context embedding and constraint embeddings. Then, it ingests the identified constraints using hard-constraints approach. In contrast, our approach is an end-to-end copy and disambiguation approach. Secondly, we do not assume source lexicon context is available which is an additional annotation that requires significant effort for each domain.

3 Approach

3.1 Problem Statement

Let each source sentence $X = (x_1, x_2, \ldots, x_S)$ of length $S$ be associated with a constraint set $C$ derived from a dictionary. A word (or phrase) is often linked with a single constraint (Chen et al., 2021b). However, in general, a source constraint can contain one or more target constraints. Each source word or phrase can be mapped to multiple constraints in $C$. For example, $(x_i, C_i^1, \ldots, C_i^k)$ denotes $k$ possible dictionary translations associated with the word $x_i$. Let the corresponding translated sentence $Y = (y_1, y_2, \ldots, y_T)$ be of length $T$. Conventional NMT systems are trained on the source and target sentences, unaware of constraints $C$. Our lexically constrained NMT system is trained with each source and target sentence, in conjunction with the possible set of constraints on the source sentence $X$.

Given a triplet of the source sentence, constraints, and target sentence $(X, C, Y)$, our objective is to choose constraint words or phrases from $C$ and potentially include them in the target sentence $Y$. (As depicted in Figure 3, the model may choose not to include any of the candidate constraints so that the flow or fluency of the predicted sentence is not adversely affected.)

3.2 DictDIS: Dictionary Disambiguation in conjunction with NMT

The source sentence is appended with $C$ constraints, where each inter-phrase constraint $C_i$ is separated by a symbol ‘<sep>’. In turn, candidate translations $C_i^j$ for this $i$th intra-phrase constraint are separated by another symbol ‘<sep>’. $\hat{X} = [X, <sep>, C_1^1, <sep>, C_1^2, <sep>, C_1^3, \ldots, <eos>]$ where $<eos>$ is the end of sentence token (c.f. Figure 2).

In our lexically constrained NMT approach, the model is trained by maximizing a log-likelihood objective similar to NMT, viz.,

$$p(Y|\hat{X}; \theta) = \prod_{t=1}^{T+1} p(y_t|y_{0:t-1}, x_{1:S}, C; \theta)$$

(1)

Unlike Hokamp and Liu (2017); Post and Vilar
Within training such that the constraints occur in the beam (within the specified size) while decoding.

3.2.1 Encoder

We employ the standard Transformer-based architecture for NMT (Vaswani et al., 2017) that uses self-attention networks for both encoding and decoding. The encoder is a stack of \( N \) identical layers, each of which contains two sub-layers. Each layer consists of a multi-head self-attention (Self-Att) and a feed-forward neural network (FFNN). For time step \( i \) in layer \( j \), the hidden state \( h_{i,j} \) is computed by employing self-attention over hidden states in layer \((j - 1)\).

We define a segment as the sequence of constraints corresponding to the textual span between two <sep> symbols. Inspired by BERT (Devlin et al., 2018), we append our input source token embedding with the learned embedding for the segment. This is illustrated in Figure 2. Further, positional indices of the constraint tokens begin with a number that is larger than the maximum source sentence length. For a given token, its embedding is the sum of three components, viz., token embedding, positional embedding, and segment embedding.

3.2.2 Decoder

With a structure similar to the encoder, the decoder consists of a stack of \( N \) identical layers. In addition to two sub-layers, it also contains a cross-attention (CrossAtt) sub-layer to capture information from the encoder. Cross-attention is computed between the last layer of the encoder hidden state \( h = h_{1,n}, h_{2,n} \ldots h_{m,n} \) and the output of the self-attention sub-layer, \( \hat{s}_t \). Each sub-layer output is followed by a layer normalization step (Ba et al., 2016). The probability for the token \( t \) is computed through a softmax over the target-side vocabulary, applied to the final decoder state \( s_t \) as

\[
P_t^{pred}(y_t | (y_{<t}, \hat{X})) = \text{softmax}(s_{t,N}, W)
\]

where \( W \) is the learnable weight matrix, \( \hat{X} \) is the source sentence with target constraints and \( y_1, y_2 \ldots y_t \) represent the target phrases and \( N \) is the final decoder block in the stack. Inspired by (Gulıçehre et al., 2016) and (Chen et al., 2021b), we introduce a pointer network that adds logit values over the target-side constraints. Specifically, the token probability over the target vocabulary is the weighted sum of decoder probabilities from the predictive model \( P_t^{pred} \) and copy probabilities \( P_t^{copy} \).

This strengthens copying by identifying the target constraint (provided as an input) that needs to be copied. Copy probability for time step \( t \) is computed as the average multi-head attention weights of the last decoder layer. Intuitively, \( P_t^{copy} \) is the
attention weight for the corresponding source position $s$.

$$P^\text{copy}_t = \text{avg}_k \left( \text{CrossAtt}(\tilde{s}_{t,N}, h^k, h^k) \right)$$ (3)

$\text{avg}_k$ denotes averaging attention weights obtained via cross-attention across all the attention heads (indexed by $k$) and $\tilde{s}_{t,N}$ is the output of $N - th$ self-attention layer for token $t$.

We further add a disambiguation network to disambiguate between multiple inter-phrase constraints. This network learns to differentiate between multiple senses of a constraint based on the sentence context. We define the disambiguation score for the $j^{th}$ inter-phrase component of the $i^{th}$ constraint, $P^\text{dis}_{ij}$ as the dot product of the attention-weighted context vector and the $j^{th}$ inter-phrase component’s contextual embedding (normalized overall $j$ inter-phrase constraints). We anticipate that constraint-component embedding combined with source context and cross-attention can distinguish between multiple inter-phrase constraints.

$$\text{score}^\text{dis}_{ij} = c_t, e^j_s = \sum_{s=1}^S \alpha_{t,s} h_s e^j_s$$

$$P^\text{dis}_{ij} = \text{normalize}_j(\text{score}^\text{dis}_{ij})$$ (4)

where $c_t = \sum_{s=1}^S \alpha_{t,s} h_s$ is the attention-weighted context vector and $e^j_s$ is the contextual embedding of inter-phrase constraints. Recall that $h_s$ is the encoder hidden state at position $s$ of the last layer, $\alpha_{t,s}$ is the averaged attention weight at the last decoder layer for source position $s$ at decoding time step $t$.

$P^\text{copy}_t + P^\text{dis}_t$ is a normalised probability distribution so that the weighted combination remains a probability distribution. The final distribution over the target vocabulary is defined as the weighted sum of $P^\text{pred}_t$ and the normalized probability sum of $P^\text{copy}_t + P^\text{dis}_t$ (refer Fig 3):.

$$p(y_t|\langle y_\text{<t}, \bar{X} \rangle) = g_t P^\text{pred}_t + (1 - g_t)(P^\text{copy}_t + P^\text{dis}_t)$$ (5)

where $g_t \in [0, 1]$ is a learnable gate controlling the weightage of the two probability distributions. It decides whether to keep the provided user constraint in translation or let the model predict (as can be seen in Fig. 3). We compute $g_t$ as $g_t = \text{FeedForward}(c_t, s_t)$ where $c_t$ is the weighted context vector, $s_t$ is the hidden state of the last layer of the decoder until timestep $t$, and FeedForward is a single layer FFNN.

### 3.3 Domain-agnostic Training

**Data.** We conduct experiments on English-Hindi, English-German, and English-French translation tasks. To train DICTDIS we construct constraints using a domain-agnostic source-target dictionary such that each source phrase can contain one or more target phrases. We use a bilingual dictionary to append constraints if a phrase in the source sentence matches the source side of the dictionary. We train our model on the Samanantar dataset (Ramesh et al., 2022) consisting of 8.4 million English-Hindi parallel sentences. We use a generic English-Hindi dictionary containing 11.5K phrases. In Appendix B, we present the distribution of phrases (in percentage) in this domain-agnostic dictionary, with respect to the number of constraints associated with each phrase. During pre-processing, we match source-side dictionary phrases with the parallel corpus and find that 96% of the sentences contain at least one constraint pair. To avoid adversely affecting the translation performance (in terms of fluency), we leave the remaining 4% sentences as unconstrained.

In the case of English-German, we exactly follow the setup of Leca (Chen et al., 2021b) and employ the WMT16 news data as a training corpus consisting of 1.8M sentences, newstest2013 as the development set, and newstest2014 as the test set. For English-French, we use the Europarl corpus for training consisting of 2M sentences and terminology tasks from WMT 21 (Alam et al., 2021) as validation and test set. To train DICTDIS, we use En-De and En-Fr bilingual dictionary from Muse dataset (Lample et al., 2018). We use Moses tokenizer (Koehn et al., 2007) for pre-processing En-De, En-Fr, and the Indic-NLP (Kunchukuttan, 2020) for pre-processing En-Hi sentences. The tokenized sentences are then processed using BPE (Sennrich et al., 2016) with 32K merge operations for the language pairs. We detokenize the predictions before computing BLEU, COMET, and CSR. We describe implementation details in Section C in the Appendix.

### 4 Experiments

#### 4.1 Evaluation

The performance of constrained machine translation is evaluated using the following three metrics, viz., (1) BLEU: The BLEU score (Papineni et al., 2002) available at [https://sanskritdocuments.org/hindi/dict/eng-hin_unic.html](https://sanskritdocuments.org/hindi/dict/eng-hin_unic.html).
for Indian languages. Two of our corpora (viz., Banking and Regulatory) were curated by aligning sentences from publicly available reports. The aerospace corpus was developed by manually translating sentences from an undergraduate textbook of aerospace engineering by a team of translators and reviewers. In addition to the curated corpus, we also present comparisons with popular benchmarks viz., Flores-101 (Goyal et al., 2022) and WAT 2021. We use administrative dictionaries on these two datasets during constrained inference. We provide details on the annotation effort of each of our in-house developed datasets in Appendix A. For En-De, following Chatterjee et al. (2022), we use Wiktionary,\textsuperscript{975} as our constraint dictionary. For En-Fr, we use the Covid-19 dictionary as provided in the WMT shared task on terminologies (Alam et al., 2021).

### 4.3 Baselines

We compare against state-of-the-art constrained as well as unconstrained approaches.

1. **Leca** (Chen et al., 2021b): This is a constrained approach that accepts a single target constraint and the source sentences as input. In case of multiple constraints for a phrase, Leca chooses a random phrase for insertion.

2. **Vectorized Dynamic Beam Allocation** (VDBA) (Hu et al., 2019) extends beam search to include pre-specified lexical constraints in the generated translation. We use a constrained decoding implementation of fairseq to run the inference over an unconstrained transformer.

3. **Transformer** refers to an unconstrained base transformer trained using fairseq.

4. **DictDIS(Uncons)** refers to the DictDIS model (trained with dictionary constraints), however during inference, we do not provide additional constraints. This helps us in comparison with the DictDIS and assess whether there is any degradation/improvement in performance as a result of adding constraints with the input. On the ‘with constraint set’ in Table 2, the objective of DictDIS(Uncons) is to compare with the Transformer which is also devoid of constraints during inference. On the ‘without constraint test sets’, DictDIS and DictDIS(Uncons) have similar input, therefore, numbers are left blank for the DictDIS(Uncons). The aim of adding

\textsuperscript{975}http://lotus.kuee.kyoto-u.ac.jp/WAT/indic-multilingual/index.html

\textsuperscript{2}https://github.com/mtresearcher/terminology_dataset/
We present BLEU scores of DICTDIS and all baselines in Table 2. We split our test set into constrained sets by matching at least a pair of source and target constraints on both source and target sentences. An unconstrained test set refers to those sentences where constraints from a bilingual dictionary are not present in both source and target sentences. For unconstrained test sets, our method DICTDIS outperforms all the baselines. We observe that all models preserve their ability to translate unconstrained test sentences.

DICTDIS(Uncons) is to a) compare with the Transformer approach where no constraints are added as input either during training and inference, and b) compare with DICTDIS to assess the impact of passing constraints during inference on the BLEU score.

### 5 Results

We present BLEU scores of DICTDIS and all baselines in Table 2. We split our test set into constrained sets by matching at least a pair of source and target constraints on both source and target sentences. An unconstrained test set refers to those sentences where constraints from a bilingual dictionary are not present in both source and target sentences. For unconstrained test sets, our method DICTDIS outperforms all the baselines. We observe that all models preserve their ability to translate unconstrained test sentences.

On the constrained test set, DICTDIS and its unconstrained variants achieve the best BLEU performance on 5 test sets. DICTDIS yields the best performance on 2 test sets and near-best scores on the other 4 test sets. For the Aerospace and Flores test set, DICTDIS achieves the best performance. On Banking, WAT 2021, and WMT 14 test sets, the un-constrained variant of DICTDIS achieves the best performance while the constrained variant trails by 0.4, 0.3, and 0.2 points respectively. On 5 out of 7 test sets, unconstrained DICTDIS performs marginally better than the constrained version. The BLEU score difference between DICTDIS(uncons) and DICTDIS(con) is 0.2-0.5 on the test sets. This could be possibly due to minor impact on fluency due to constraint ingestion. The greedy beam search decoding approach is affected by the constraints ingested in the previous step(s) due to which marginal degradation in BLEU scores are observed. Despite achieving similar scores, DICTDIS(con) reports better CSR that unconstrained on all the test sets in Table 3. On the other hand, DICTDIS(uncons) demonstrates the advantage of training with a constrained decoding method.

Leca performs poorly on all the constrained test sets. The difference between Leca and the best-performing method on the Banking, Aerospace, Regulatory, Flores, WAT2021, and WMT14 set is 5.6, 11.4, 1.4, 5.1, 7.2, and 3.1 BLEU scores respectively. VDBA (Hu et al., 2019) enforces constraints only considering the target tokens of the lexicons which reduce the fluency of the sentence. VDBA algorithm enforces constraints to appear in the output which is reflected in close to 99% CSR scores for the test sets but very low BLEU scores on
out-of-domain test sets. Leca has far better BLEU scores than VDBA due to its training framework which focuses on learning a gating value of constraints when the decoder-encoder attention over the constraints has some significant value.

We perform a paired significance test between DICTDIS and Transformer baseline at $p < 0.05$ (Koehn, 2004) with bootstrap resamples as 1000. In Table 2, we highlight statistical significant scores with an underline.

### 5.1 Copy Success rate (CSR)

In Table 3, we present CSR of various baselines and DICTDIS. DICTDIS yields better CSR scores for Aerospace and WAT2021, whereas Leca has better CSR on Regulatory, Banking, and WMT14 test sets. Leca achieves better CSR scores on test sets where single target constraints form a major part of a constrained set (c.f., Table 1) such as Banking, Regulatory, and WMT14. On the contrary, it performs poorly on constraints having a polysemous degree $>$1. DICTDIS constrained setting performs second best on the Banking, Regulatory, and WMT14 datasets and yields the overall best performance on other test sets.

While Leca only utilizes a copy network, DICTDIS takes advantage of both copy and disambiguation networks. In Table 4, we present polysemous degree-wise CSR scores for different test sets and constrained approaches. As pointed out in the preceding discussion, Leca ingests single-degree constraints aggressively; however, it is unable to pick appropriate constraints when the polysemy degree is 2 or higher. DICTDIS performs consistently better for polysemy degree $>$2 and yields an overall higher BLEU score on most datasets.

Low disambiguation performance on Regulatory and Banking can be explained by the inability of DICTDIS to ingest unambiguous (i.e., polysemy degree 1) constraints aggressively in the predicted sentence. The Regulatory dataset includes around 90% unambiguous constraints from the banking dictionary while other engineering and medical datasets have a sizeable proportion of constraints with degree $>$1. In the case of Regulatory, Leca ingests unambiguous constraints aggressively, resulting in much better disambiguation performance than other approaches. On the Banking dataset, Leca is unable to disambiguate ambiguous phrases with degree 2 polysemy. On other datasets, DICTDIS has better disambiguation performance and superior BLEU scores.

### 5.2 Controlled Text Generation

We observe that DICTDIS is not aggressive in ingesting single-degree constraint in favor of fluency. Therefore, inspired by controlled text generation approaches (Pascual et al., 2021; Dathathri et al., 2019), we introduce a user-controllable parameter $\alpha$ which controls the aggressiveness of ingesting single-degree constraints. $\alpha$ is governed by the normalized cross attention on the constraints for each time step $t$. The final value of logits of constraints is increased by $\alpha \times \text{CrossAtt}_t$. In Table 5, we present results for $\alpha = 0.1$ which is the optimum value governing the tradeoff between CSR and BLEU. We observe that DICTDIS + $\alpha$ achieves better CSR than DICTDIS on these test sets without sacrificing fluency, unlike Leca. We observe that higher values of $\alpha$ result in high CSR but at the expense of lower BLEU scores. In Table 8, 9 and 10 in Appendix E, we present examples of translations with constrained approaches, viz., DICTDIS and Leca.

### 6 Conclusion

We present a copy-and-disambiguation approach, viz., DICTDIS to translate under dictionary constraints provided at run-time, that potentially include multiple target candidates for each source language phrase. We present a recipe for training DICTDIS on a generic parallel corpus by synthesizing constraints during training by leveraging domain-agnostic dictionaries. We present an exten-

| Test Sets | Banking | Aerospace | Regulatory | Flores | WAT2021 | WMT14 (En-De) | WMT21 (En-Fr) |
|-----------|---------|-----------|------------|--------|---------|---------------|---------------|
| Polysemy  | 1 2     | 1 2 3 4 5 | 1 2 3 4    | 1 2    | 1 2 3 4 5 | 1 2 3 4 5     | 1 2 3 4       |
| Leca      | 95 70   | 95 57 68  | 95 70 74   | 94 74  | 95 74 65 54 | 94 74 64 65   | 93 67 68 59   |
| DictDis   | 92 90   | 89 79 82  | 92 87 89   | 93 92  | 91 82 79 83 | 91 82 79 82   | 91 82 79 82   |
| DictDis + $\alpha$ | 32.9 87.8 | 25.9 86.7 | 25.9 86.7 | 25.9 86.7 | 25.9 86.7 | 25.9 86.7 | 25.9 86.7 |

Table 4: Results for polysemous degree-wise CSR for different test sets and constrained approaches.

| Model   | Banking | Regulatory |
|---------|---------|------------|
| Leca    | 28.5 91.7 | 25.4 94.1 |
| DictDis | 33.7 82.1 | 26.3 85.3 |
| DictDis + $\alpha$ | 32.9 87.8 | 25.9 86.7 |

Table 5: BLEU and CSR for Banking and Regulatory dataset with constraint ingestion parameter $\alpha = 0.1$. 

| Test Sets | Banking | Aerospace | Regulatory | Flores | WAT2021 | WMT14 (En-De) | WMT21 (En-Fr) |
|-----------|---------|-----------|------------|--------|---------|---------------|---------------|
| Polysemy  | 1 2     | 1 2 3 4 5 | 1 2 3 4    | 1 2    | 1 2 3 4 5 | 1 2 3 4 5     | 1 2 3 4       |
| Leca      | 95 70   | 95 57 68  | 95 70 74   | 94 74  | 95 74 65 54 | 94 74 64 65   | 93 67 68 59   |
| DictDis   | 92 90   | 89 79 82  | 92 87 89   | 93 92  | 91 82 79 83 | 91 82 79 82   | 91 82 79 82   |
| DictDis + $\alpha$ | 32.9 87.8 | 25.9 86.7 | 25.9 86.7 | 25.9 86.7 | 25.9 86.7 | 25.9 86.7 | 25.9 86.7 |

Table 4: Results for polysemous degree-wise CSR for different test sets and constrained approaches.
sive evaluation of DICTDIS on existing datasets as well as on several new datasets, that we manually curated from the finance and engineering domains.

7 Limitations

A major limitation with training the current framework is the need for bilingual synonym dictionaries to have good coverage of words/phrases in the source sentences. Although synonym dictionaries are available for high-resource languages\(^4\), they are not readily available for low-resource languages. Secondly, our framework is limited by the maximum length of constraints. In cases where the length of constraints is greater than 1024 characters, our model chooses to ignore such constraints. Thirdly, model training time is 3x times higher than the base transformer primarily due to higher input length.

Acknowledgements

We thank Piyush Sharma for his contributions in the initial experiments of the paper. We thank Anuja Dumada, Pranita Harpale and Atul Kumar Singh for preparing few evaluation datasets. Ayush Maheshwari did this work as part of the PhD at IIT Bombay and was supported by a fellowship Ekal Foundation during his PhD. Ganesh Ramakrishnan is grateful to the National Language Translation Mission (NLTM): Bhashini project by Government of India and IIT Bombay Institute Chair Professorship for their support and sponsorship.

References

Md Mahfuz Ibn Alam, Ivana Kwapilíková, Antonios Anastasopoulos, Laurent Besacier, Georgiana Dinu, Marcello Federico, Matthias Gallé, Kweonwoo Jung, Philipp Koehn, and Vassilina Nikouliina. 2021. Findings of the wmt shared task on machine translation using terminologies. In Proceedings of the Sixth Conference on Machine Translation, pages 652–663.

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

Yujin Baek, Koanho Lee, Dacyeon Ki, Cheonbok Park, Hyoung-Gyu Lee, and Jaegul Choo. 2023. Towards accurate translation via semantically appropriate application of lexical constraints. In Findings of the Association for Computational Linguistics: ACL 2023, pages 4839–4855.

Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, and Matthias andigital Humanities Huck. 2018. Findings of the 2019 conference on machine translation (WMT19). page 9. Frontiers.

Niccolò Campolungo, Federico Martelli, Francesco Saina, and RobertoNavigli. 2022. Dibimt: A novel benchmark for measuring word sense disambiguation biases in machine translation. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 4331–4352.

Soumya Chatterjee, Sunita Sarawagi, and Preethi Jyothi. 2022. Accurate online posterior alignments for principled lexically-constrained decoding. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 6675–6689.

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.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2019. Plug and play language models: A simple approach to controlled text generation. In International Conference on Learning Representations.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

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.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc’Aurelio Ranzato, Francisco Guzman, and Angela Fan. 2022. The flores-101 evaluation benchmark for low-resource and multilingual machine translation. Transactions of the Association for Computational Linguistics, 10:522–538.

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.

\(^4\)https://github.com/facebookresearch/MUSE
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.

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

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In Proceedings of the 2004 conference on empirical methods in natural language processing, pages 388–395.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In Proceedings of the 45th annual meeting of the Association for Computational Linguistics companion volume proceedings of the demo and poster sessions, pages 177–180.

Anoop Kunchukuttan. 2020. The indicnlp library.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc’Aurelio Ranzato. 2018. Unsupervised machine translation using monolingual corpora only. In International Conference on Learning Representations.

Ayush Maheshwari, Ajay Ravindran, Venkatapathy Subramanian, and Ganesh Ramakrishnan. 2023. Udama: machine learning based post-editing tool for document translation. In Proceedings of the 6th Joint International Conference on Data Science & Management of Data (10th ACM IKDD CODS and 28th COMAD), pages 263–267.

Ayush Maheshwari, Nikhil Singh, Amrith Krishna, and Ganesh Ramakrishnan. 2022. A benchmark and dataset for post-ocr text correction in sanskrit. In Findings of the Association for Computational Linguistics: EMNLP 2022, pages 6258–6265.

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.

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.

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.

Damian Pascual, Beni Egressy, Clara Meister, Ryan Cotterell, and Roger Wattenhofer. 2021. A plug-and-play method for controlled text generation. In Findings of the Association for Computational Linguistics: EMNLP 2021, pages 3973–3997.

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.

Gowtham Ramesh, Sumanth Doddapaneni, Aravindh Bheemaraj, Mayank Jobanputra, Raghavan AK, Ajitesh Sharma, Sujit Sahoo, Harshita Diddee, Divyanshu Kakwani, Navneet Kumar, et al. 2022. Samanantar: The largest publicly available parallel corpora collection for 11 indic languages. Transactions of the Association for Computational Linguistics, 10:145–162.

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

Ryokan Ri, Toshiaki Nakazawa, and Yoshimasa Tsuruoka. 2021. Modeling target-side inflection in place-holder translation. In Proceedings of the 18th Biennial Machine Translation Summit (Volume 1: Research Track), pages 231–242.

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

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.
Appendix

A Dataset

We adopted following two methods to create the parallel corpora for English-Hindi test set:

1. **Alignment**: We align parallel data available in the finance domain. Several banking organizations in India produce their work reports in both Hindi and English. We semi-automatically align around 3K sentences using our in-house developed tool (Maheshwari et al., 2023).

2. **Manual Translation**: We obtained manual translations for engineering curriculum books from English into Hindi with the help of professional translators. This was with requisite permissions from the corresponding publishers and authors.

   i) **Regulatory**: These are manually translated sentence pairs of the annual report of a central bank. We extract the sentences using OCR and automatically align sentences using multilingual sentence embeddings. The aligned sentences are then manually reviewed for alignment and OCR errors (as adopted in Maheshwari et al. (2022)). The dictionary here primarily belongs to the finance domain. We used the Universal Sentence Encoder Model\(^5\) to compute sentence embedding for both English and Hindi sentences. Then, we compute the dot product and pair the closest matching sentence above the heuristically defined threshold. Secondly, we ask a human annotator to verify the automatically aligned sentences. We observe around 95% accuracy (with a drop of around 15% sentences due to thresholding).

   ii) **Aerospace**: This is an undergraduate aerospace engineering book. Since aerospace is a multi-disciplinary field, we choose dictionaries from physics, chemistry, mechanical, and math domains to use with DICTDIS. Any single dictionary may not provide sufficient enough coverage over domain-specific terms.

   iii) **Banking**: These are manually translated sentence pairs of the annual report of a banking organization. We adopt the same method as in the regulatory dataset to derive the test set. We recruited expert linguistic translators who are domain experts and worked as part of at least 3 book translation project. We provided source phrases in Microsoft Excel and asked translators to check the corresponding translations. All the translators were

---

5https://tfhub.dev/google/universal-sentence-encoder/4
duly compensated as per industry standards.

**B  Training Data**

We use a generic English-Hindi synonym dictionary containing 11.5K phrases. In Table 6, we present the distribution of phrases (in percentage) in this domain-agnostic dictionary across different numbers of constraints associated with a phrase. During pre-processing, we match source-side dictionary phrases with the parallel corpus and find that 96% of the sentences contain at least one constraint pair. To avoid adversely affecting the translation performance (in terms of fluency), we leave the remaining 4% sentences as unconstrained.

![Table 6: Percentage of phrases having different numbers of constraints in the constraint dictionary during training.](image)

**C  Implementation Details**

We implement DICTDIS using fairseq toolkit (Ott et al., 2019) v0.12\(^6\) over base Transformer model (Vaswani et al., 2017) for all our experiments. The results are reported for the single run. We use the Byte-Pair Encoder (BPE) tokenizer () with a maximum vocabulary size of 32000 for all our experiments. We set the maximum token length (including all inter-phrase constraints and intra-phrase constraints) to 2048. The optimizer employed is Adam (Kingma and Ba, 2014) with label smoothing of 0.1, the learning rate is set to 5e-4 with 4000 warm-up steps, the probability dropout is set to 0.3, maximum token length in a batch to 4096 and maximum number of updates to 200,000. The beam size for all experiments and baselines is set to 5. To maintain consistency in the length of constraints in the batch, we pad tokens to make their lengths equal. Training takes approximately 3 hours for 1 epoch on four Nvidia A6000 GPUs in a distributed setup.

**D  COMET scores on Test sets**

In Table 7, we present mean COMET scores for the constrained test sentences i.e. test sentences containing both source and target-side constraints. COMET (Rei et al., 2020) metric is based on a pre-trained language model that accepts source, reference, and predicted sentence as input. COMET has shown a higher correlation with human judgments and its ability to identify the highest-performing MT systems. We use pre-trained wmt20-comet-da model for calculating COMET scores. We observe that DICTDIS achieves the highest rank among various baselines on 5 out of 6 test sets. En-De language pair has a small/negative score for a few baselines which could possibly be due to the different scores scales. However, DICTDIS ranks higher for the majority of test sets.

**E  Examples for qualitative illustration**

In Table 8, 9 and 10, we present example translations from Flores, Aerospace dataset, and Regulatory dataset respectively. We present outputs with constrained approaches, viz., DICTDIS and Leca, and the unconstrained Transformer. For the specified input sentence in English and corresponding dictionary constraints, our approach incorporates constraints without adversely affecting the fluency of the sentence. In the examples, DICTDIS incorporates constraints specified during runtime into the translation and produces better sentence translations. Since our method is trained with multiple polysemy constraints, it learns to choose (or drop) appropriate constraints.

\(^6\)https://github.com/facebookresearch/fairseq/
### Table 7: COMET scores for the constrained test sets. Note that COMET scores are representative of ranking relative to direct assessment as measured by the pre-trained model instead of the translation quality.

| Model          | Banking | Aerospace | Regulatory | Flores | WAT2021 | WMT14 |
|----------------|---------|-----------|------------|--------|---------|-------|
| VDBA           | 71.6    | 65.6      | 62.3       | 51.9   | 94.2    | -3.7  |
| Leca           | 64.4    | 48.9      | 60.9       | 51.4   | 78.5    | -13.1 |
| Transformer    | 71.4    | 66.8      | 71.4       | 62     | 94.6    | 2.9   |
| DictDIS        | 74.5    | 77.3      | 65.7       | 72.8   | 97.5    | 8.2   |

### Table 8: Examples from the test set using DictDIS and Leca (constrained) models. Here, the correctly ingested constraints are highlighted inbold, and incorrectly ingested constraints are highlighted as underlined. Sentences have been written in the IAST format for readability. DictDIS is generally more effective in ingesting the constraints appropriately than Leca while maintaining fluency.

#### Model | Sentences
---|---
Input | Stability and Turn Down Ratio The operation of the combusting device must be stable over the operating range. ('ratio', 'anapātu,anupāta'), (operation, 'saṃkṣirya,pracālana'), ('down', 'adha,nīce'), ('range', 'parāsa,pariṣara,mālā,śrenī,parāsa,pariṣara,mālā,śrenī'), ('stability', 'sthāyitva')
Constraints | ('transformation', 'rūpāntaraṇa') ('precise', 'pratatha,parisuddha')
DictDIS | 
Transformer | kamābhiṣṭinga divālsa kā saṃčālana auparetiṅga remja para sthirā honā cāhe l
Leca | sthāyitva aura tāraṇa dāūna anupāta kambhastinga yuktī kā pracālana pracālana simā para sthirā honā cāhe l
Reference | sthāyitva aura tāraṇa dāūna anupāta kambhastinga yuktī kā pracālana pracālana simā para sthirā honā cāhe l

#### Table 9: Constrained translation examples from the Aerospace test set using DictDIS, Transformer (unconstrained) and Leca (constrained) models. Here, the correctly ingested constraints are highlighted inbold, and incorrectly ingested constraints are highlighted as underlined. Sentences have been written in the IAST format for readability.

#### Model | Sentences
---|---
Input | isa taraha ke parivartana kī satīka yāṃstrī kābhī taka saṃstōṣaṇaṅka dhānga se nāṁhī bantaī gai hā l
Constraints | isa taraha ke rūpāntaraṇa kī yathārtha kriyāvidhi kā bhī taka saṃstōṣaṇaṅka rūpa se pāṭa nāṁhī calā hā l
DictDIS | isa taraṇa ke rūpāntaraṇa kī pravīṣṭha abhī taka saṃstōṣaṇaṅka dhānga se nāḍa-nāḍa nāṁhī honā huā l
Transformer | isa taraṇa ke rūpāntaraṇa kī satīka yāṃstrī kābhī taka saṃstōṣaṇaṅka dhānga se kāma nāṁhī kīyā gāyā huā l
Leca |
Forward exchange contracts are valued half-yearly, and net loss, if any, is provided for in the Exchange Equalisation Account (EEA).

vāyādā vinimaya samvidāem kā mūlyānāṃkāra arddha vāyākā hotā hai aura nivala hāni, yadi koī ho, ekṣecemānāṃkāra akīmūmā (īe) mem pradāna ki jāti hai

The Union Budget aims to keep the economy on the path of fiscal consolidation.

kendrīya bājaṭā kā uddeśya arthavayavasthā ko rājakosiya samekana ke mārga para banāe rakhanā hai

Notable gains in containment of key deficit indicators in 2013-14 are apparent in the provisional accounts (PA).

calū khāṭā samānke para niyatramā se prāpta lābha ke bāre mem anāntima lekhe (pirī) mem spāṭa hai

Notwithstanding deterioration in export performance brought on, inter alia, by weak external market conditions, the current account deficit narrowed in 2014-15 from its level a year ago on terms of trade gains and weak import demand.

niṟiyāṭa pradaṇsā mem giravaṭa ke bāvajūda kamaṃjāra bājāra paristhitīyom ke kārāṇa calū khāṭā eka sālā pahale ke stara se kama hokara eka sālā pahale 2014-15 mem vyāpāra lābha aura kamaṃjāra āyāṭa māṃga ke kārāṇa kama ho gayā thē

anyā bāṭoṃ ke sāṭha-sāṭha kamaṃjāra bāhāyī bājāra paristhitīyom ke kārāṇa niṟiyāṭa ke pradaṇsā mem giravaṭa ke bāvajūda calū khāṭā gūrāhī varṣa 2014-15 ke apanē stara se eka varṣa pahale vyāpāra lābha aura kamaṃjāra āyāṭa sāṭreṃ ke saṃdārba mem kama ho gayā thē

āyī garvāṭa ke bāvajūda varṣa 2014-15 kā calū khāṭā gūrāhī pichale varṣa ke isake stara se kama thē kyunkī vyāpāra āyāṭa sāṭreṃ tathā āyāṭa ki kamaṃjāra māṃga se phāyadā huṣ thē

Table 10: Phrasal constraint examples from Regulatory dataset using DictDis and Transformer (unconstrained). Here, the correctly ingested constraints are highlighted in bold and incorrectly ingested constraints are highlighted as underline.