Categorizing Semantic Representations for Neural Machine Translation

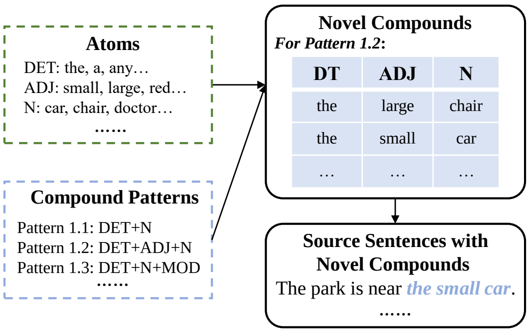
Yongjing Yin[1,2]*, **Yafu Li**[1,2], **Fandong Meng**[3], **Jie Zhou**[3], **Yue Zhang**[2,4]
[1] Zhejiang University
[2] School of Engineering, Westlake University
[3] Pattern Recognition Center, WeChat AI, Tencent Inc
[4] Institute of Advanced Technology, Westlake Institute for Advanced Study
{yinyongjing,liyafu}@westlake.edu.cn {fandongmeng,withtomzhou}@tencent.com
yue.zhang@wias.org.cn

Abstract

Modern neural machine translation (NMT) models have achieved competitive performance in standard benchmarks. However, they have recently been shown to suffer limitation in compositional generalization, failing to effectively learn the translation of atoms (e.g., words) and their semantic composition (e.g., modification) from seen compounds (e.g., phrases), and thus suffering from significantly weakened translation performance on unseen compounds during inference. We address this issue by introducing categorization to the source contextualized representations. The main idea is to enhance generalization by reducing sparsity and overfitting, which is achieved by finding prototypes of token representations over the training set and integrating their embeddings into the source encoding. Experiments on a dedicated MT dataset (i.e., CoGnition) show that our method reduces compositional generalization error rates by 24% error reduction. In addition, our conceptually simple method gives consistently better results than the Transformer baseline on a range of general MT datasets.

1 Introduction

Neural machine translation (NMT) has achieved competitive performance on benchmark datasets such as WMT (Vaswani et al., 2017; Edunov et al., 2018; So et al., 2019). However, the generalizaiton to low-resource domains (Bapna and Firat, 2019b; Zeng et al., 2019; Bapna and Firat, 2019a; Khandelwal et al., 2021) and robustness to slight input perturbations (Belinkov and Bisk, 2018; Xu et al., 2021b) are relatively low for NMT models. In addition, recent studies show that NMT systems are vulnerable to *compositional generalization* (Lake and Baroni, 2018; Raunak et al., 2019; Guo et al., 2020; Li et al., 2021; Dankers et al., 2021; Chaabouni et al., 2021), namely the ability to understand and produce a potentially infinite (formally exponential to the input size) number of novel combinations of known atoms (Chomsky, 2009; Montague and Thomason, 1975; Janssen and Partee, 1997; Lake and Baroni, 2018; Keysers et al., 2020a).



DT	ADJ	N
the	large	chair
the	small	car
…	…	…

Figure 1: The novel compounds in the CoGnition test set are constructed by composing a few basis semantic atoms (e.g., determiners (DET), nouns (N), and adjectives (ADJ)) according to the composition patterns. The compounds are then put into corresponding source contexts extracted from the training data.

Take CoGnition (Li et al., 2021), a dedicated MT dataset, for example (Figure 1). Despite that certain instances of translation atoms (e.g., *small*, *large*, *car*, and *chair*) and their semantic compositions (e.g., *small chair* and *large car*) are frequent in training data, unseen compositions of the same atoms (e.g., *large chair*) during testing can suffer from large translation error rates. Compositionality is also a fundamental issue in language understanding and motivated for translation (Janssen and Partee, 1997; Janssen, 1998), which has been suggested as being essential for robust translation (Raunak et al., 2019; Li et al., 2021) and efficient low-resource learning (Chaabouni et al., 2021).

The current dominant method to NMT employs a sequence-to-sequence architecture (Sutskever et al., 2014; Vaswani et al., 2017), where an encoder is used to find representations of each input token that thoroughly integrates its sequence-level context in-

[*] This work was done as an intern at Pattern Recognition Center, WeChat AI, Tencent Inc, China.

Proceedings of the 29th International Conference on Computational Linguistics, pages 5227–5239
October 12–17, 2022.

formation, and a decoder refers to such contextualized representations for generating a translation sequence. A key reason of failure on compositional generalization is that the correspondence between pairs of *token sequences* is modeled as a whole. Specifically, NMT models are trained end-to-end over large parallel data without disentangling the representation of individual words or phrases from that of whole token sequences. At the sequence level, the source input sample space is highly sparse mainly due to semantic composition, and small changes to a sentence can lead to out-of-distribution issues (Sagawa et al., 2020; Conklin et al., 2021; Liu et al., 2021).

Intuitively, one way to solve this problem is to decouple token-level information from the source sequence by injecting token-level translation distribution (e.g., $P(petit|small)$) into the source representation. Given the fact that the source-side contextualized representations encode rich token-level translation information (Kasai et al., 2021; Xu et al., 2021a), we categorize sparse token-level contextualized source representations into a few representative prototypes over training instances, and make use of them to enrich source encoding. In this way, when encoding a sequence, the model observes less sparse prototypes of each token, thus alleviating excessively memorizing the sequence-level information.

We propose a two-stage framework to train prototype-based Transformer models (Proto-Transformer). In the first stage, we warm up an initial Transformer model which can generate reasonable representations. In the second stage, for each token, we run the trained model to extract all contextualized representations over the training corpus. Then, we perform clustering (e.g., K-Means) to obtain the prototype representations for each token. Take Figure 2 as an example, for the token "Toy", we collect all the contextualized representations and cluster them into 3 prototypes. Finally, we extend the base model by fusing the prototype information back into the encoding process through a prototype-attention module, and continue to train the whole model until convergence.

Experimental results on CoGnition show that our method significantly improves novel composition translation by over 24% error reduction, demonstrating the effectiveness for tackling the compositional generalization problem. To further verify the effectiveness on more datasets, we conduct experiments on 10 commonly used MT benchmarks and our method gives consistent BLEU improvement. We also present empirical analysis for prototypes and quantitative analysis on compositional generalizaiton. The comparison between the one-pass and the two-pass training procedure shows that the one-pass method is both faster and more accurate than the two-pass one, demonstrating that more generalizable prototypes extracted from early training phrase are more beneficial to compositional generalization. Additionally, quantitative analysis demonstrates that our proposed model is better at handling longer compounds and more difficult composition patterns. The code is publicly available at https://github.com/ARIES-LM/CatMT4CG.git.

2 Related Work

Compositional Generalization Recent work (Lake and Baroni, 2018; Keysers et al., 2020b) has demonstrated weak compositionality of neural models using dedicated datasets. Various methods haven been proposed to solve the issue of compositional generalization such as encoding more inductive bias (Li et al., 2019; Korrel et al., 2019; Baan et al., 2019; Chen et al., 2020a; Gordon et al., 2020; Herzig and Berant, 2021), meta-learning (Lake, 2019; Conklin et al., 2021), and data augmentation (Andreas, 2020; Akyürek et al., 2021). Recently, Ontañón et al. (2021) and Csordás et al. (2021) show that the Transformer architecture can perform better on compositional generalization with some modifications. Although these methods have demonstrated better generalization or interpretability, most of them are limited small vocabulary and limited samples semantic parsing datasets. In the context of machine translation, Lake and Baroni (2018) construct a small dataset where the training data contains a word *daxy* along with its parallel sentences of a single pattern (e.g., *I am daxy*, *je suis daxist*) while the test set contains novel patterns (e.g., *He is daxy*). However, the experiment is limited in that the test set only consists of 8 samples. Different from existing work, Li et al. (2021) propose a large dataset (CoGnition) and construct a large-scale test set that contains newly constructed constituents as novel compounds, so that generalization ability can be evaluated directly based on compound translation error rate. We proposed a method enhancing compositional generalization on the dedicated dataset of Li et al. (2021), while at the same time gives improvements to the machine

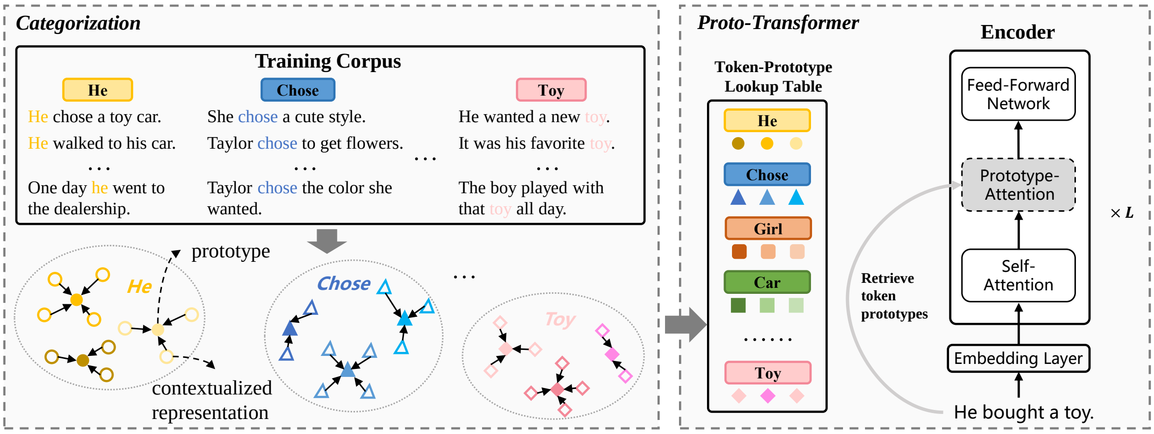


Figure 2: Architecture of Proto-Transformer. The dotted box denotes the prototype-attention introduced in stage 2.

translation quality at practical test cases.

Neural Machine Translation Recent research on NMT has paid increasing attention to robustness (Cheng et al., 2018, 2020; Xu et al., 2021b), domain adaptation (Bapna and Firat, 2019b; Zeng et al., 2019; Bapna and Firat, 2019a; Khandelwal et al., 2021), and compositional generalization (Lake and Baroni, 2018; Raunak et al., 2019; Fadaee and Monz, 2020; Guo et al., 2020; Li et al., 2021). Lake and Baroni (2018) propose a simple toy experiment to first show the problem of compositionality. Raunak et al. (2019) find that NMT models behave poorly on recombining known parts and generalizing on samples beyond the observed length during training, Fadaee and Monz (2020) find that NMT models are vulnerable to modifications such as removal of ad-verbs and number substitutions. More recently, Li et al. (2021) observe significant compositional generalization issues on CoGnition, and Dankers et al. (2021) argue that MT is a suitable testing ground to ask how compositional models trained on natural data are. Our work is in line with the above methods, but we consider a method to address the issue rather than analyse the problem. Technically, Raunak et al. (2019) propose to use bag-of-word regularization to refine encoder and Guo et al. (2020) propose sequence-level mixup to create synthetic samples. Different from them, we propose to enhance models' compositional generalization by categorizing contextualized representations, which turns out more effective.

3 Method

3.1 Transformer Baseline

Given a sequence of source sentence $X = \{x_1, ..., x_T\}$, where T denotes the number of tokens, the Transformer encoder (Vaswani et al., 2017) first maps X to embeddings, packing them as a matrix H^0, and then takes H^0 as input and outputs a contextualized sequence representation $H^L \in \mathbb{R}^{d \times T}$, where d and L denote dimension size and the number of layers respectively.

Attention. Formally, given a set of packed query, key, and value matrices Q, K, and V, the dot product attention mechanism are defined as

$$\text{Attention}(Q,K,V) = \text{Softmax}(\frac{Q^TK}{\sqrt{d}})V, \quad (1)$$

where d is the dimension of the key vector.

A typical extension of the above is multi-head attention (MHA), where multiple linear projections are executed in parallel, and the outputs of all heads are concatenated:

$$\text{MHA}(Q,K,V) = W^O[head_1; ...; head_h], \quad (2)$$

$$head_i = \text{Attention}(W_i^Q Q, W_i^K K, W_i^V V), \quad (3)$$

where W^O, W_i^Q, W_i^K, and W_i^V are model parameters.

Layer Structure. The Transformer encoder has L identical layers, each of which is composed of two sublayers (i.e., self-attention and feed-forward networks). In the l-th self-attention layer, the query, key, and value matrices are all the hidden states from the previous layer H^{l-1}:

$$H_a^l = \text{MHA}(H^{l-1}, H^{l-1}, H^{l-1}). \quad (4)$$

The feed-forward sublayer is a two-layer transformation with ReLU activation:

$$H^l = W_2^l\text{ReLU}(W_1^l H_a^l + b_1^l) + b_2^l, \quad (5)$$

where W_1^l, b_1^l, W_2^l, and b_2^l are trainable parameters. The layer normalization and residual connection are omitted for brevity.

3.2 Proto-Transformer

Our proposed Proto-Transformer extends the base Transformer by introducing a prototype-attention module on top of the self-attention module in each encoder layer, which aggregates token-level prototype representations for each token.

3.2.1 Prototype-attention

Assuming that each token x_t in the input sequence is assigned a set of prototype vectors packed as a matrix $C^{x_t} \in \mathbb{R}^{d \times k}$ where k is the number of prototypes. The prototype attention aggregates the global prototype information and refines the contextualized representations for each token by the multi-head attention mechanism:

$$H_p^l = \text{MHA}(H_a^l, C, C), \tag{6}$$

where C is all the prototype representations of the sequence (i.e., $\{C^{x_1}, C^{x_1}, ..., C^{x_T}\}$), and the localness of prototype attention is implemented by mask mechanisms. The output H_p^l is fed into the feed-forward network (Eq. 5).

The difference between the Proto-Transformer encoder and the Transformer encoder is illustrated by the dotted box in Figure 2. By stacking multiple self-attention layers with prototype-attention, the encoder is able to exploit less-sparse prototype representations, preventing the model from over-memorizing the local context.

3.2.2 Training

Proto-Transformer takes a two-stage training process, which is summarized in Algorithm 1. Here, $\mathcal{D}$ denotes a training corpus, $\mathcal{V}$ denotes the vocabulary of $\mathcal{D}$, $\theta^{(0)}$ denotes initial parameters of a Transformer model, θ_+ denotes parameters of the prototype-attention module, and k and N denote the number of prototypes and training epochs of stage 1 respectively.

Stage 1. We first train a base Transformer model for N epochs until it is able to generate reasonable translations. Given the training corpus $\mathcal{D} = \{(X, Y)\}$, where X and Y denote a source sentence and target translation, respectively, the model is optimized by minimizing cross-entropy loss. After training for N epochs we obtain a model $\theta^{(N)}$, which has acquired some translation knowledge.

Algorithm 1 The training procedure of Proto-Transformer

```
Input: Training set $\mathcal{D} = \{(X, Y)\}$, vocabulary $\mathcal{V}$, NMT model $\theta^{(0)}$, prototype-attention modules $\theta_+$, number of prototype $k$, training epochs of stage one $N$, token-representation lookup table $\mathcal{Q}$, token-prototype lookup table $\mathcal{P}$
Output: Trained NMT model $\theta^*$
Initialize $\theta^{(0)}$ and $\theta_+$
Stage 1
for $i = 1$ to $N$ do
    $\theta^{(i)} \leftarrow$ TrainModel($\mathcal{D}$, $\theta^{(i-1)}$)
end for
Stage 2
for $X \in \mathcal{D}$ do
    for each token $x_i$ in $X$ do
        Add $H_i$ to the corresponding contextualized representation list $\mathcal{Q}[x_i]$ for token $x_i$
    end for
end for
Apply K-Means to each token $v_j \in \mathcal{V}$ to obtain prototype representations $C^{v_j}$ and add $(v_j, C^{v_j})$ to $\mathcal{P}$
Introduce prototype-attention modules to the base model: $\theta^{(N)} \leftarrow \theta^{(N)} \cup \theta_+$
Continue training to convergence:
$\theta^* \leftarrow$TrainModel($\mathcal{D}$, $\theta^{(N)}$,$\mathcal{P}$)
```

Stage 2. We cluster the contextualized representations of each token to obtain its prototype representations. In particular, using the model $\theta^{(N)}$, we build a token-representation lookup table $\mathcal{Q}$. We iterate through the whole training corpus and calculate contextualized representations $\{H_1^{v_i}, ..., H_{R(v_i)}^{v_i}\}$ of each source token v_i in the vocabulary except punctuation, where $R(v_i)$ is the number of contextualized representations of token v_i. Here we omit the superscript L for simplicity. Next, for each token v_i, we use K-Means (Lloyd, 1982) [1] to cluster the contextualized representations due to its efficiency for large number of samples in high dimensions:

$$C^{v_i} = \text{K-Means}(H_1^{v_i}, ..., H_{R(v_i)}^{v_i}). \tag{7}$$

All token-prototype pairs are saved in a lookup table $\mathcal{P}$. Finally, we introduce the prototype-attention modules with parameters θ^+ to the Transformer encoder, and for each source sentence we retrieve the prototype representations from table $\mathcal{P}$. We continue to optimize the whole parameter set $\{\theta^{(N)} \cup \theta^+\}$ to obtain a final model θ^*.

As mentioned earlier in the introduction, we compare the above one-pass method to a two-pass method, where we train the base model to convergence instead of for N iterations, before training

[1] https://scikit-learn.org/stable/modules/classes.html

a Proto-Transformer from scratch which uses clustered prototypes from the base model. The relative advantage is that the prototypes are taken from a fully trained NMT model, but the disadvantage is that the training time doubles. In addition, it remains a empirical question whether fully-trained NMT models give prototypes that are more suitable for guiding a final model. We discuss this in the Section 5.1.

It is worth noting that a limitation of K-Means is the pre-specification of cluster numbers, which can be different for different tokens in practice. Non-parametric clustering algorithms such as DBSCAN (Ester et al., 1996) can potentially solve the problem. However, their runtime complexity is typically $O(n^2)$ or worse, which prevents us from choosing them for large-scale NMT data.

4 Experiments

4.1 Experimental Settings

Datasets. We use CoGnition (Li et al., 2021) to systematically evaluate compositional generalization in MT scenarios, an English→Chinese (En→Zh) translation dataset. In consists of a training set of 196,246 sentence pairs, a valid set and a test set of 10,000 samples. In particular, it has a dedicated test set (i.e., CG-test set) consisting of 10,800 sentences containing novel compounds, so that the model's ability of compositional generalization can be measured by the ratio of compounds that are correctly translated. In addition, we choose 9 machine translation tasks from IWSLT, WMT and JRC-Acquis to verify the general effectiveness of our methods. The dataset statistics are shown in Appendix B.

Setup. We use Transformer (Vaswani et al., 2017) as our baseline models implemented using the Fairseq toolkit (Ott et al., 2019). For CoGnition and IWSLT, we use the Transformer iwslt_de_en setting while for the others we use the transformer_base setting. Following previous work, for IWSLT and JRC-Acquis En↔Es, we use beam search with width 5 and length penalty 0.6 for inference, whereas for the other datasets we set the beam width as 4. For CoGnition, N and k are set as 8 and 3 based on the validation set, and N and k for the other datasets are shown in Appendix B. Empirically, N is recommended to chosen from 10% to 25% of the total training epochs. We conduct categorization for all tokens except punctuation and low-frequent words, and the token on CoGniton is word and is subword on other datasets For reproducibility and stability, we train 6 models with seeds provided in (Li et al., 2021) for each method and report the average performance.

Evaluation Metrics. We use compound translation error rate (CTER; (Li et al., 2021)) to measure model performance on CoGnition. Specifically, *instance-level* CTER denotes the ratio of samples where the novel compounds are translated incorrectly, while *aggregate-level* CTER denotes the ratio of compounds that suffer at least one incorrect translation in corresponding contexts. To calculate CTER, Li et al. (2021) provide a manually collected dictionary for all the atoms based on the training set, since each word may have different translations. We also conduct human evaluation (Appendix A) as a supplement. We conduct evaluation using BLEU (Papineni et al., 2002) for the other datasets.

4.2 Baseline Methods

We compare our method with following baselines: (1) Transformer (Vaswani et al., 2017), which uses the same settings as Li et al. (2021); (2) Transformer-Small, a more compact Transformer model with 4 layers and 256 hidden size; (3) Transformer-Deep, which increases the number of encoder layers to 8 to take the same parameters as Proto-Transformer; (4) Transformer-Rela, which replaces absolute positional encoding with a relative one, an important component for compositional generalizaiton demonstrated in (Csordás et al., 2021; Ontañón et al., 2021); (5) Bow (Raunak et al., 2019): which uses bag-of-words loss to regularize the encoder based on the observation that the encoder representations are much weaker on unseen composition; and (6) SeqMix (Guo et al., 2020): which synthesizes examples by interpolating embeddings[2].

4.3 Results on CoGnition

The main results on CoGnition are shown in Table 1. Transformer gives instance-level and aggregate-level CTERs of 28.42% and 62.88%, respectively. In comparison, Proto-Transformer gives a score of 21.69% and 51.84%, respectively, with a significant improvement of 6.73% and 11.04% accordingly.

[2] The performance is the best when the hyper-parameters of Beta distribution are set to 1.0, which is consistent with (Guo et al., 2020).

Model	Compound Translation Error Rate (CTER) ↓			
	NP	VP	PP	Total
Transformer	24.74%/55.16%	24.82%/59.54%	35.71%/73.94%	28.42%/62.88%
Transformer-Deep	25.11%/54.11%	28.14%/60.43%	37.38%/75.24%	30.21%/63.26%
Transformer-Small	22.14%/47.21%	23.55%/53.60%	32.02%/69.64%	25.91%/56.82%
Bow	22.16%47.89%	24.83%/55.57%	35.04%/73.21%	27.34%/58.89%
SeqMix	24.52%/49.71%	26.88%/58.90%	34.36%/73.09%	28.59%/60.57%
Transformer-Rela	22.69%/51.20%	24.89%/57.17%	34.94%/71.73%	27.50%/60.03%
Proto-Transformer	**14.07%/36.45%**	**22.13%/50.90%**	**28.85%/68.15%**	**21.69%/51.84%**

Table 1: Compound translation error rate (CTER) on CoGnition. We report instance-level and aggregate-level CTERs, separated by "/". NP, VP, and PP denote noun phrases, verb phrases and positional phrases, respectively, three compound types in the compositional generalization test set (CG-test set).

Moreover, Proto-Transformer outperforms all baseline systems significantly, indicating that categorization on the contextualized representations is more beneficial to compositional generalization. We also calculate the BLEU scores, and Proto-Transformer and Transformer obtain 60.1 and 59.5, respectively. The BLEU scores are all relatively high since the sentences on the CG test set are similar to the sentences in training data except the novel compounds. For CoGnition, CTER is more accurate and suitable than BLEU because the translation dictionary processes multiple accurate translations of each compound better.

Since Proto-Transformer brings some extra parameters (approximately 6M), we investigate whether the performance improvement is derived from the increase of model parameters. As can be seen, Transformer-Deep performs poorly on the CG-test set, indicating that only increasing model capacity may be harmful since it leads to worse over-fitting to sequence-level distributions. Besides, the more compact model Transformer-Small yields better results than Transformer and Transformer-Deep but lags far behind Proto-Transformer. This shows that model size is useful but not sufficient for solving compositional generalization.

Proto-Transformer performs better than Bow, indicating that the encoder representations refined by categorization are more adequate than the regularization technique. Compared to SeqMix, the improvement of Proto-Transformer is more significant (2.31% vs 11.04% aggregate-level CTER). SeqMix reduces representation sparsity via linear interpolation in the input embedding space, and we conjecture that the stochastically synthetic samples may be unreasonable and harmful to model training. Relative positional embedding (Shaw et al., 2018) is demonstrated to be important for compositional generalization. Specifically, Transformer-Rela reduces CTERs by 0.92% and 2.85% but is inferior to ours, indicating that the prototypes bring more than positional information.

Dataset	Direction	BLEU	
		Transformer	Proto-TF
IWSLT	En⇒De	28.44	**28.96**
	En⇒It	28.24	**28.87**
	En⇒Vi	30.19	**30.96**
WMT	En⇒Ro	32.46	**33.37**
	Ro⇒En	32.49	**33.27**
	En⇒Fi	21.39	**22.11**
	En⇒De	27.95	**28.49**
JRC-Acquis	En⇒Es	60.90/60.32	**61.56/60.90**
	Es⇒En	63.21/62.59	**63.73/63.10**

Table 2: BLEU on commonly used MT datasets. For JRC-Acquis, we follow previous work to report results on both the valid and test sets, separated by "/". The performance of our model is significantly better than Transformer ($p < 0.05$) (Koehn, 2004).

4.4 Results on General MT Datasets

We conduct experiments on several general MT benchmarks, where the translation compound error rate (CTER) cannot be calculated directly. The performances on IWSLT, WMT and JRC-Acquis are presented in Table 2. Compared with Transformer, our model achieves consistent improvement (0.67 BLEU), demonstrating its effectiveness under general evaluation settings. Proto-Transformer outperforms Transformer by 0.91 and 0.78 BLEU scores on WMT'16 En→Ro and Ro→En respectively. For JRC-Acquis, our model achieves an average improvement of 0.62 BLEU score on En→Es and 0.56 BLEU score on Es→En. On the largest dataset WMT'16 En→De, our model also performs better than Transformer by 0.54 BLEU score. In addition, the large datasets possibly benefit from the additional parameters (about 6M). We run an experiment of a model with 8 encoder layers on EN-DE,

k	CTER	k	CTER
0	28.42%/62.88%	3	21.69%/51.84%
1	23.60%/54.33%	4	23.04%/52.70%
2	22.58%/53.19%	5	24.98%/54.71%

Table 3: CTERs against different number of prototypes on the CG-test set. “0” denotes to the Transformer baseline.

and it achieves 28.02 BLEU, inferior to our model with same number of parameters.

Note that the BLEU scores in the general datasets can reflect translation quality, but has several limitations in the measurement of generalization or robustness. For example, when a sentence is almost translated correctly but has few serious errors, it can achieve a high BLEU score but suffers severe semantic distortion. For instance, as shown in (Li et al., 2021), the model mis-translates “He became sick from eating all of the peanut butter on the ball“ into “He became sick from eating all of the peanut butter on the field“. With a minor mistake on the compound “on the ball“, the model achieves a sentence-level BLEU of 61.4, but the full sentence meaning is largely affected.

5 Analysis

5.1 Prototypes

Effect of the Number of Prototypes. The number of prototypes k controls the granularity of context used to refine the representation learning, and is determined it based on the model loss on the development set. In this experiment, we investigate its influence on generalization performance. As shown in Table 3, incorporating prototypes can reduce the translation error caused by unseen compounds, and the model obtains the best generalization performance when k is 3. Intuitively, using too many prototypes dilutes the concentration effect, leading to overfitting again, while too few prototypes limits the expressiveness for polysemous words, e.g., one prototype represents a single sense.

Effect on Token-level Translation Consistency. We hypothesize that token alignments induced by Proto-Transformer are less ambiguous for tokens having settled meaning. To verify it, we approximately measure the number of distinct translations for 10 selected English tokens in the WMT Ro→En dataset, by counting the number of Romanian tokens that are aligned to the English tokens in the test set with the method of Chen et al. (2020b).

Tgt tokens	PT	TF	Src tokens
yesterday	1	3	**ieri**,psd,victor
limited	2	5	**limitata**,**limitat**,comert,apel, ...
six	1	6	**sase**,procente,saptamani,din, ...
agriculture	1	4	**agricultura**,plasa, ...
culture	2	5	**cultura**,**culturii**, profesion...
republic	1	3	**republica**,liderului, ...
simply	1	3	**simplu**,nu,individuala, ...
november	1	6	**noiembrie**,propaganda,2008, ...
tomorrow	1	3	**maine**,dimineata,amiaza, ...
saturday	1	3	**sambata**,seara,dimineata, ...

Table 4: Word translation variations induced from Transformer (TF) and Proto-Transformer (PT). The columns PT and TF denote the number of aligned Romanian tokens, and the bold characters display the tokens induced from Proto-Transformer.

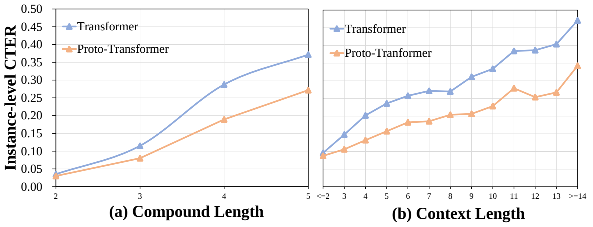


Figure 3: CTERs of Proto-Transformer and Transformer over different compound and context lengths.

As can be seen in Table 4, Proto-Transformer attends to much fewer token types than Transformer. This shows that Proto-Transformer’s translations are less prone to context changes, which indicates the sparsity reduction of input sample space. The finding also partly explains the improvement on the Ro→En dataset. The above observations verify our intuition in the introduction that adding prototypes to the source representation enhances the model’s knowledge on token-level translation consistency, which leads to better compositional generalizaiton.

It is noteworthy that increase translation consistency may increase translationese effects (Vanmassenhove et al., 2021). We argue that there can be a trade-off between the consistency modeling and translationese. Our approach does not force the model to generate highly consistent translation, but aims to alleviate the vulnerability to context changes by introducing several prototypes.

One-pass vs Two-pass. The possible advantage of the two-pass training procedure (Two-Pass) mentioned in Introduction is that the quality of the prototype representations generated by a converged model may be better. Specifically, Two-Pass achieves 24.36% instance-level and 55.69% aggregate-level CTER, outperforming Transformer by 4.06% and 7.19%, respectively, but is noticeably inferior to Proto-Transformer (21.69% and 51.84%). This observation may relate to the observation that neural models learn more generalizable features in the early phase before memorization (Arpit et al., 2017; Stephenson et al., 2021). The one-pass approach can not only reduce the training cost (11,103 seconds vs 6,286 seconds) but also leverage more generalizable features to better improve models' compositional generalization. In addition, the prototypes can be initialized randomly and trained along with the model, and this variant obtains 26.24% and 57.94% CTERs. It preforms much worse than Proto-Transformer (21.69% and 51.84%) though they have identical number of parameters and architecture, indicating the effectiveness of explicit categorization during training.

5.2 Effects on Compositional Generalization

Composition Length. Longer compounds are harder to generalize as they contain richer semantic information (Li et al., 2021). We classify the test samples by compound length and context length, and calculate the instance-level CTER. In Figure 3, we can observe that the advantage of Proto-Transformer grows larger in generalizing longer compounds and context. In particular, Proto-Transformer gives a lower CTER by 12.80% over samples with context longer than 13 tokens. The underlying reason can be that longer compounds or contexts are more sparse in input space, and Proto-Transformer alleviates sparseness by putting token distributions into representations via prototypes.

Modifier. One challenging type of novel compounds in CoGnition is the postpositive modifier atom (MOD), which is constructed to enrich the information of its preceding word (e.g., *he liked* in the sentence *he bought the car he liked*). The difficulty of translating compounds with MOD lies in word reordering from English to Chinese. We divide the test samples into two groups according to compounds with or without MOD (Figure 4). Proto-Transformer demonstrates larger advantage in translating the compounds with MOD, showing its superiority in processing complex semantic composition. To further understand the effectiveness, we choose the token *liked*, a core part of the MOD atom *he liked*, and visualize its representations with t-SNE (van der Maaten and Hinton, 2008). In Figure 5, we can see that the representations of *he liked* serving as MOD are concentrated at the leftmost prototype, while the representations of *he liked* not serving as MOD scatter on the other prototypes. Through representation categorization, Proto-Transformer finds a specialized prototype which abstracts the knowledge of *liked* serving as a part of MOD, and refers to the MOD prototype when processing compounds with MOD.

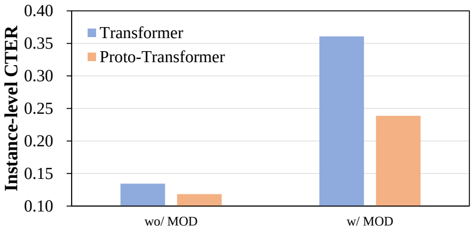


Figure 4: CTERs on compounds w/o and w/ MOD.

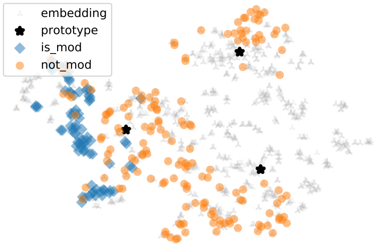


Figure 5: Visualization of contextualized embeddings (grey tri-down markers) and 3 prototypes (black stars) for the token *liked*. The blue diamonds and orange circles denote the case where *liked* appear in the context *he liked*, and the former one denotes the case where *he liked* serves as MOD.

Case Study We present 3 source samples containing a novel compound *woke the silly boyfriend up* and 5 atoms, i.e., *woke*, *the*, *silly*, *boyfriend* and *up*, and their translations in Table 5. For all samples, correct translations should contain complete semantic meaning of the compound. Proto-Transformer correctly translates the compound along with different contexts across all samples, while Transformer suffers various mistakes. In the first sample, the translation of *something terrible* is omitted by Transformer. In the second sample, Transformer omits *silly boyfriend* and mistranslates *woke the silly boyfriend up* into *a silly bird woke up*. Similar mistake can be observed in the third example, where Transformer mistranslates *his dog woke the silly boyfriend up* into *his dog woke up*. Transformer overfits contexts and makes errors on unseen compositions, while our model is more stable.

Source	Transformer	Proto-Transformer
Yet to her dismay something terrible **woke the silly boyfriend up**.	然而，使她沮丧的是， 一个可怕的男友醒来了。 (Yet to her dismay **a silly boyfriend woke up**.)	但令她沮丧的是， 某种可怕的东西把那个傻男友吵醒了。 (Yet to her dismay something terrible **woke the silly boyfriend up**.)
Every morning , a bird **woke the silly boyfriend up** for work at 6 am.	每天早上6点， 一只愚蠢的鸟叫醒去上班。 (every morning , **a silly bird woke up** for work at 6 am.)	每天早上6点，一只鸟 叫醒那个愚蠢的男朋友去上班。 (Every morning , a bird **woke the silly boyfriend up** for work at 6 am.)
His dog **woke the silly boyfriend up** in the middle of the night.	他的狗半夜醒来了那个傻男友。 (His dog **woke up** in the middle of the night, **the silly boyfriend**.)	他的狗半夜把那个傻男友吵醒了。 (His dog **woke the silly boyfriend up** in the middle of the night.)

Table 5: Example translations. The bold characters denoting the novel compounds and corresponding translations.

6 Conclusion

We investigated a conceptually simple method for enhancing compositional generalizaiton of NMT models, proposing a two-stage training framework to fuse prototype representations into the encoding process of Transformer. Experiments on CoGnition show the effectiveness of our method on compositional generalization, and extensive results over 9 translation tasks verify the generality of our method. To our knowledge, we are the first to propose token-level categorization for NMT, achieving promising performance on both a large-scale compositional generalization dataset and general datasets.

Acknowledgements

The work is funded by the Zhejiang Province Key Project 2022SDXHDX0003. We would like to thank all of the anonymous reviewers and Jiali Zeng for the helpful comments. Yue Zhang is the corresponding author.

References

Ekin Akyürek, Afra Feyza Akyürek, and Jacob Andreas. 2021. Learning to recombine and resample data for compositional generalization. In *Proc. of ICLR*.

Jacob Andreas. 2020. Good-enough compositional data augmentation. In *Proc. of ACL*, pages 7556–7566.

Devansh Arpit, Stanislaw Jastrzebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron C. Courville, Yoshua Bengio, and Simon Lacoste-Julien. 2017. A closer look at memorization in deep networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 233–242. PMLR.

Joris Baan, Jana Leible, Mitja Nikolaus, David Rau, Dennis Ulmer, Tim Baumgärtner, Dieuwke Hupkes, and Elia Bruni. 2019. On the realization of compositionality in neural networks. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 127–137.

Ankur Bapna and Orhan Firat. 2019a. Non-parametric adaptation for neural machine translation. In *Proc. of NAACL-HLT*, pages 1921–1931.

Ankur Bapna and Orhan Firat. 2019b. Simple, scalable adaptation for neural machine translation. In *Proc. of EMNLP*, pages 1538–1548.

Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and natural noise both break neural machine translation. In *Proc. of ICLR*.

Rahma Chaabouni, Roberto Dessì, and Eugene Kharitonov. 2021. Can transformers jump around right in natural language? assessing performance transfer from SCAN. In *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 136–148, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xinyun Chen, Chen Liang, Adams Wei Yu, Dawn Song, and Denny Zhou. 2020a. Compositional generalization via neural-symbolic stack machines. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Yun Chen, Yang Liu, Guanhua Chen, Xin Jiang, and Qun Liu. 2020b. Accurate word alignment induction from neural machine translation. In *In Proc. of EMNLP*, pages 566–576.

Yong Cheng, Lu Jiang, Wolfgang Macherey, and Jacob Eisenstein. 2020. AdvAug: Robust adversarial augmentation for neural machine translation. In *Proc. of ACL*, pages 5961–5970.

Yong Cheng, Zhaopeng Tu, Fandong Meng, Junjie Zhai, and Yang Liu. 2018. Towards robust neural machine translation. In *Proc. of ACL*, pages 1756–1766.

Noam Chomsky. 2009. *Syntactic structures.*

Henry Conklin, Bailin Wang, Kenny Smith, and Ivan Titov. 2021. Meta-learning to compositionally generalize. In *Proc. of ACL*, pages 3322–3335.

Róbert Csordás, Kazuki Irie, and Jürgen Schmidhuber. 2021. The devil is in the detail: Simple tricks improve systematic generalization of transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 619–634. Association for Computational Linguistics.

Verna Dankers, Elia Bruni, and Dieuwke Hupkes. 2021. The paradox of the compositionality of natural language: a neural machine translation case study. *CoRR*, abs/2108.05885.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proc. of EMNLP*, pages 489–500.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96), Portland, Oregon, USA*, pages 226–231.

Marzieh Fadaee and Christof Monz. 2020. The unreasonable volatility of neural machine translation models. In *Proceedings of the Fourth Workshop on Neural Generation and Translation*, pages 88–96.

Jonathan Gordon, David Lopez-Paz, Marco Baroni, and Diane Bouchacourt. 2020. Permutation equivariant models for compositional generalization in language. In *Proc. of ICLR*.

Jiatao Gu, Yong Wang, Kyunghyun Cho, and Victor O. K. Li. 2018. Search engine guided neural machine translation. In *Proc. of AAAI*, pages 5133–5140.

Demi Guo, Yoon Kim, and Alexander Rush. 2020. Sequence-level mixed sample data augmentation. In *Proc. of EMNLP*, pages 5547–5552.

Jonathan Herzig and Jonathan Berant. 2021. Span-based semantic parsing for compositional generalization. In *Proc. of ACL*, pages 908–921.

Theo M. V. Janssen. 1998. Algebraic translations, correctness and algebraic compiler construction. *Theor. Comput. Sci.*, 199(1-2):25–56.

Theo M. V. Janssen and Barbara H. Partee. 1997. Compositionality. In Johan van Benthem and Alice ter Meulen, editors, *Handbook of Logic and Language*, pages 417–473. North Holland / Elsevier.

Jungo Kasai, Nikolaos Pappas, Hao Peng, James Cross, and Noah A. Smith. 2021. Deep encoder, shallow decoder: Reevaluating non-autoregressive machine translation. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Daniel Keysers, Nathanael Schärli, Nathan Scales, Hylke Buisman, Daniel Furrer, Sergii Kashubin, Nikola Momchev, Danila Sinopalnikov, Lukasz Stafiniak, Tibor Tihon, Dmitry Tsarkov, Xiao Wang, Marc van Zee, and Olivier Bousquet. 2020a. Measuring compositional generalization: A comprehensive method on realistic data. In *Proc. of ICLR*.

Daniel Keysers, Nathanael Schärli, Nathan Scales, Hylke Buisman, Daniel Furrer, Sergii Kashubin, Nikola Momchev, Danila Sinopalnikov, Lukasz Stafiniak, Tibor Tihon, Dmitry Tsarkov, Xiao Wang, Marc van Zee, and Olivier Bousquet. 2020b. Measuring compositional generalization: A comprehensive method on realistic data. In *Proc. of ICLR*.

Urvashi Khandelwal, Angela Fan, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2021. Nearest neighbor machine translation. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proc. of EMNLP*, pages 388–395.

Kris Korrel, Dieuwke Hupkes, Verna Dankers, and Elia Bruni. 2019. Transcoding compositionally: Using attention to find more generalizable solutions. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 1–11.

Brenden M. Lake. 2019. Compositional generalization through meta sequence-to-sequence learning. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 9788–9798.

Brenden M. Lake and Marco Baroni. 2018. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *Proc. of ICML*, Proceedings of Machine Learning Research, pages 2879–2888.

Jason Lee, Elman Mansimov, and Kyunghyun Cho. 2018. Deterministic non-autoregressive neural sequence modeling by iterative refinement. In *Proc. of EMNLP*, pages 1173–1182.

Yafu Li, Yongjing Yin, Yulong Chen, and Yue Zhang. 2021. On compositional generalization of neural machine translation. In *Proc. of ACL*, pages 4767–4780.

Yuanpeng Li, Liang Zhao, Jianyu Wang, and Joel Hestness. 2019. Compositional generalization for primitive substitutions. In *Proc. of EMNLP*, pages 4293–4302.

Evan Zheran Liu, Behzad Haghgoo, Annie S. Chen, Aditi Raghunathan, Pang Wei Koh, Shiori Sagawa, Percy Liang, and Chelsea Finn. 2021. Just train twice: Improving group robustness without training group information. In *Proc. of ICML*, Proceedings of Machine Learning Research, pages 6781–6792.

Stuart P. Lloyd. 1982. Least squares quantization in PCM. *IEEE Trans. Inf. Theory*, 28(2):129–136.

Richard Montague and Richmond H Thomason. 1975. Formal philosophy. selected papers of richard montague. *Erkenntnis*, (2).

Santiago Ontañón, Joshua Ainslie, Vaclav Cvicek, and Zachary Fisher. 2021. Making transformers solve compositional tasks. *CoRR*, abs/2108.04378.

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 *Proc. of ACL*, pages 311–318.

Vikas Raunak, Vaibhav Kumar, and Florian Metze. 2019. On compositionality in neural machine translation. *CoRR*, abs/1911.01497.

Shiori Sagawa, Aditi Raghunathan, Pang Wei Koh, and Percy Liang. 2020. An investigation of why overparameterization exacerbates spurious correlations. In *Proc. of ICML*, Proceedings of Machine Learning Research, pages 8346–8356.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proc. of ACL*, pages 1715–1725.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, pages 464–468. Association for Computational Linguistics.

David R. So, Quoc V. Le, and Chen Liang. 2019. The evolved transformer. In *Proc. of ICML*, Proceedings of Machine Learning Research, pages 5877–5886.

Cory Stephenson, Suchismita Padhy, Abhinav Ganesh, Yue Hui, Hanlin Tang, and SueYeon Chung. 2021. On the geometry of generalization and memorization in deep neural networks. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112.

Laurens van der Maaten and Geoffrey E. Hinton. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605.

Eva Vanmassenhove, Dimitar Shterionov, and Matthew Gwilliam. 2021. Machine translationese: Effects of algorithmic bias on linguistic complexity in machine translation. In *Proc. of EACL*, pages 2203–2213.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Hongfei Xu, Josef van Genabith, Qiuhui Liu, and Deyi Xiong. 2021a. Probing word translations in the transformer and trading decoder for encoder layers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 74–85. Association for Computational Linguistics.

Weiwen Xu, Ai Ti Aw, Yang Ding, Kui Wu, and Shafiq Joty. 2021b. Addressing the vulnerability of NMT in input perturbations. In *Proc. of NAACL*, pages 80–88.

Jiali Zeng, Yang Liu, Jinsong Su, Yubing Ge, Yaojie Lu, Yongjing Yin, and Jiebo Luo. 2019. Iterative dual domain adaptation for neural machine translation. In *Proc. of EMNLP*, pages 845–855.

A Human evaluation

Model	Adequacy	Fluency
Transformer	4.32	4.27
Proto-Transformer	4.51	4.54

Table 6: Human evaluation on adequacy and fluency.

We conduct the human evaluation for translations in terms of adequacy and fluency. We randomly sample 100 sentences from the CG-test set of CoGnition, and invite three annotators to evaluate the translation adequacy and fluency ranging from one to five. The five point scale for adequacy indicates how much of the meaning expressed in the reference translation is also expressed in a hypothesis translation: 5 = All, 4 = Most, 3 = Much, 2 = Little, and 1=None. The five point scale for fluency indicates how fluent the translation is: 5 = Flawless, 4 = Good, 3 = Non-native, 2 = Disfluent, and 1 = Incomprehensible.

The average of the scores from the three annotators is taken as the final score, and the results of the baseline and our model are shown in Table 6. Compared with Transformer, Proto-Transformer improves adequacy and fluency by 0.19 and 0.27, respectively. Proto-Transformer achieves more accurate translation of the novel compounds by alleviating the problem of compositional generalization, and possibly make the other part of the sentence to be translated better.

B Dataset Statistics and Hyper-parameters

We list the statistics and the introduced hyper-parameters of all the datasets in Table 7.

For IWSLT'14 English↔German (En↔De), IWSLT'14 English→Italian (En→It), and IWSLT'15 English→Vietnamese (En→Vi), we use Moses tokenizer[3] and apply joint BPE (Sennrich et al., 2016) with 10,000 merge operations. For WMT'16 English↔Romanian (En↔Ro), we use the processed data from Lee et al. (2018). For WMT'16 English→German (En→De) and WMT'17 English→Finnish (En→Fi), we apply joint BPE with 37,000 and 32,000 merge operations, respectively. For JRC-Acquis English↔Spanish (En↔Es) we use the datasets processed by Gu et al. (2018). For WMT'17 En-Fi, we use the concatenation of newstest2015, newsdev2015, newstest2016 and newstestB2016 as the development set, and the newstest2017 as the test set.

C Target-side Prototypes.

Target-side prototypes possibly contain more bilingual translation knowledge since the decoder processes target sentences based on source representations. We are interested in whether target-side prototypes can be used to enhance compositional generalization and/or further improve Proto-Transformer. To answer this question, we first extract prototypes from the decoder and incorporate them back to the decoder using the same mechanism of Proto-Transformer. The model can reduce CTERs to 25.93%/58.21%, largely inferior to using source-side prototypes (21.69%/51.84%). We also try to incorporate target-side prototypes to the decoder based on Proto-Transformer but it gives no noticeable improvement, achieving 21.65% and 51.71% CTERs, respectively. The underlying reason is connected a recent finding that translation already happens in the source encoding and the representations from encoders contain sufficient translation knowledge (Kasai et al., 2021; Xu et al., 2021a).

D Computational Cost

Our framework adds some computational overheads including extracting contextualized representations and conducting the clustering algorithm. The former only requires a single forward pass over the training set, merely amounting to a fraction of the cost of training for one epoch. Thanks to pytorch implementation of K-means algorithms which utilize GPU for faster matrix computations, the clustering is friendly for the large-scale datasets and it can be much faster with parallelization and more powerful hardware. Since the number of prototypes is a constant k, the complexity of prototype-attention is $O(kT)$, linear with respect to sequence length T.

[3] https://github.com/moses-smt/

	CoGnition	IWSLT'14		IWSLT'15	WMT'16			WMT'17	JRC-Acquis	
	En-Zh	En-De	En-It	En-Vi	En-Ro	En-Ro	En-De	En-Fi	En-Es	Es-En
#Train	196k	157k	175k	133k	608k	608k	4.5M	2.6M	679k	679k
#Valid	10k	7k	1k	1.6k	2k	2k	3k	9k	25k	25k
#Test	10k	7k	0.9k	1.3k	2k	2k	3k	3k	26k	26k
N	8	12	10	8	8	11	15	12	8	8
k	3	4	4	3	4	6	5	6	4	4

Table 7: Dataset statistics and hyper-parameters.