Disentangled Sequence to Sequence Learning for Compositional Generalization

Hao Zheng and **Mirella Lapata**
Institute for Language, Cognition and Computation
School of Informatics, University of Edinburgh
10 Crichton Street, Edinburgh EH8 9AB
Hao.Zheng@ed.ac.uk mlap@inf.ed.ac.uk

Abstract

There is mounting evidence that existing neural network models, in particular the very popular sequence-to-sequence architecture, struggle to systematically generalize to unseen compositions of seen components. We demonstrate that one of the reasons hindering compositional generalization relates to representations being *entangled*. We propose an extension to sequence-to-sequence models which encourages disentanglement by adaptively re-encoding (at each time step) the source input. Specifically, we condition the source representations on the newly decoded target context which makes it easier for the encoder to exploit specialized information for each prediction rather than capturing it all in a single forward pass. Experimental results on semantic parsing and machine translation empirically show that our proposal delivers more disentangled representations and better generalization. [1]

1 Introduction

When humans use language, they exhibit compositional generalization; they are able to produce and understand a potentially infinite number of novel linguistic expressions by systematically combining known atomic components (Chomsky, 2014; Montague, 1970). For example, if a person knows the meaning of the utterance "A boy ate the cake on the table in a house" and the verb "like", it is natural for them to understand the utterance "A boy likes the cake on the table in a house" when they encounter it for the first time (see Table 1). Humans are also adept at recognizing novel combinations of familiar syntactic structure, e.g., they would have no trouble processing the above sentence if the preposition "beside the tree" were added to it, despite not having previously seen the phrase "in a house beside the tree" (see Table 1).

[1]Our code is available at https://github.com/mswellhao/Dangle.

Training Set
A boy ate the cake on the table in a house. *cake(x_4); *table(x_7); boy(x_1) AND eat.agent(x_2, x_1) AND eat.theme(x_2, x_4) AND cake.nmod.on(x_4, x_7) AND table.nmod.in(x_7, x_{10}) AND house(x_{10})
Test Set (Lexical Generalization)
A boy likes the cake on the table in a house. *cake(x_4); *table(x_7); boy(x_1) AND like.agent(x_2, x_1) AND like.theme(x_2, x_4) AND cake.nmod.on(x_4, x_7) AND table.nmod.in(x_7, x_{10}) AND house(x_{10})
Test Set (Structural Generalization)
A boy ate the cake on the table in a house beside the tree. *cake(x_4); *table(x_7); *tree(x_{13}); boy(x_1) AND eat.agent(x_2, x_1) AND eat.theme(x_2, x_4) AND cake.nmod.on(x_4, x_7) AND table.nmod.in(x_7, x_{10}) AND house(x_{10}) AND house.nmod.beside(x_{10}, x_{13})

Table 1: Examples from COGS (Kim and Linzen, 2020) showcasing lexical and structural generalization. In lexical generalization, a familiar word (e.g., *like*) is attested in a familiar syntactic structure but the resulting combination has not been seen before. In structural generalization, familiar syntactic components give rise to novel combinations (e.g., only prepositional phrases with nesting depth 2 have been previously seen whereas new combinations show nestings of depth 3 or 4). All PP modifiers are assumed to have an NP-attachment reading and all modifications are nested rather than sequential. Definite descriptions are marked with * and appear to the leftmost of the logical form.

There has been a long standing debate whether this systematicity can be captured by connectionist architectures (Fodor and Pylyshyn, 1988; Marcus, 2003; Lake and Baroni, 2018) and recent years have witnessed a resurgence of interest thanks to the tremendous success of neural networks at various natural language understanding and generation tasks (Sutskever et al., 2014; Vaswani et al., 2017; Dong and Lapata, 2016; Jia and Liang, 2016). Mounting evidence, however, suggests that existing models, in particular the very popular sequence-to-sequence architecture, struggle with compositional generalization (Finegan-Dollak et al., 2018; Lake and Baroni, 2018; Keysers et al., 2020; Herzig and Berant, 2021). This failure may be due to spurious

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 4256 - 4268
May 22-27, 2022

correlations which hinder out-of-distribution generalization (Gururangan et al., 2018; Arjovsky et al., 2019; Sagawa et al., 2020) or limited robustness to perturbations in the input (Cheng et al., 2018).

In this paper, we identify an *entanglement problem* with how different semantic factors (e.g., lexical meaning and semantic relations) are represented in neural sequence models that hurts generalization. In theory, neural networks should represent semantic factors in a disentangled way by virtue of the principle of compositionality (Frege, 1884; Partee, 1995) which implies that semantic properties of syntactic constituents are to a certain extent context invariant and the semantic primitives they express are conditionally independent.

Disentangled meaning representations ought to preserve this conditional independence, and neural units modeling a particular semantic factor should be relatively invariant to changes in other factors (Bengio et al., 2013). For example, the relation between "table" and "house" in Table 1 and its representation should not be affected by whether there is a PP modifying "house". However, in a standard neural encoder (e.g., transformer-based) semantic factors tend to be entangled so that changes in one factor affect the representation of others. We further illustrate this problem in an artificial setting and find that a simple marking strategy enhances the learning of disentangled representations.

Motivated by this finding, we propose an extension to sequence-to-sequence (seq2seq) models which allows us to learn disentangled representations for compositional generalization. Specifically, at each time step of the decoding, we adaptively re-encode the source input by conditioning the source representations on the newly decoded target context. We therefore build specialized representations which make it easier for the encoder to exploit relevant-only information for each prediction. Experiments on three benchmarks, namely COGS (Kim and Linzen, 2020), CFQ (Keysers et al., 2020), and CoGnition (Li et al., 2021), empirically verify that our proposal leads to better generalization, outperforming competitive baselines and more specialized techniques.

2 Disentanglement in a Toy Experiment

We first shed light on the problem of entangled representations with a toy experiment and then move on to describe our modeling solution. For simplicity, we only focus on relations as the kind of semantic factors a model aims to represent, but the entanglement issue could also exist in representations of other factors, such as lexical meaning.

Data Creation Let $x = [e_1, r_1, e_c, r_2, e_2]$ denote a sequence of symbols. We want to predict the relation between e_1 and e_c, and e_c and e_2, which we denote by $y = (y_1, y_2)$, with $y_1 \in L_1$ and $y_2 \in L_2$ where L_1 are a set of relation labels for y_1 and L_2 are a set of relation labels for y_2. For simplicity, we set e_1, e_c, and e_2 to the same symbol e (i.e., $e_1, e_c, e_2 \in \{e\}$) whereas $r_1 \in R_1$ and $r_2 \in R_2$ denote different relation symbols, and R_1 and R_2 are the corresponding sets of relation candidates. In this toy setting, we will further assume that different relation symbols determine different relation labels (e.g., for the phrases "cat in house" and "cat with house", "in" and "with" represent two distinct relations between "cat" and "house"). In reality, relations between words could be dependent on broader context or not verbalized at all. We also assume that there is a one-to-one mapping between relation symbols and relation labels (i.e., between L_1 and R_1 and L_2 and R_2).

We construct a training set by including examples $[e_1, r_1, e_c, r_2, e_2]$ where r_1 is the same relation symbol throughout while r_2 can be any relation symbol in R_2 ($r_1 \in \{r_{train}\}$, $r_2 \in R_2$). We also include examples $[e_1, r_1, e_c]$ with all relation symbols from R_1 occurring in isolation ($r_1 \in R_1$). This way, the training set covers all primitive relations, but contains only a particular type of relation composition (i.e., $\{r_{train}\} \times R_2$). In contrast, the test set contains all unseen compositions $[e_1, r_1, e_c, r_2, e_2]$ (i.e., $r_1 \in R_1 \backslash \{r_{train}\}, r_2 \in R_2$) which will allow us to evaluate a model's ability to generalize. We set each relation set to include 10 relation symbols ($|R_1| = |R_2| = 10$).

Finally, we simplistically only consider the relations of target word e_c with its left and right words e_1 and e_2. In reality, a model would be expected to capture sentence-level semantics, i.e., a word's relation to *all* context words in a sentence (including no relation).

Modeling For each input symbol, we sample a vector from a Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{0.2^2 I})$ and freeze it during training. We then embed each example x into a sequence of vectors $[w_1, w_2, ..., w_n]$ (where $n = 3$ or $n = 5$) and transform them into contextualized representations $[h_1, h_2, .., h_n]$ using a Transformer encoder

(Vaswani et al., 2017). To predict the relation between two symbols, we concatenate their corresponding representations and feed the resulting vector to an MLP for classification.

To study how changes in relation y_1 affect the prediction of y_2 at test time, we explore two training methods. One is joint training where a model learns to predict both y_1 and y_2 (i.e., h_1 and h_3 are concatenated to predict y_1 or h_3 and h_5 are concatenated to predict y_2). The other method is separate training where a model is trained to only predict y_2 (i.e., only h_3 and h_5 are concatenated to predict y_2). For separate training, we basically ignore examples $[e_1, r_1, e_c]$ which only include r_1, as they have no bearing on the prediction of y_2.

Observation With separate training, the model learns to ignore r_1, the accuracy of predicting y_2 on the test set is 100%, regardless of which value r_1 takes. This indicates that random perturbation of r_1 alone does not lead to generalization failure. It also follows that there is no spurious correlation between r_1 and y_2. However, when the model is trained to predict both relations (which is what happens in realistic settings since we need to capture all possible relations) r_1 has a huge impact on the prediction of y_2 whose accuracy drops to approximately 55%. Taken together, these results suggest that the model fails to generalize to new relation compositions due to its internal representations being entangled and as a result changes in one relation affect the representation of others.

Why is there a wide performance gap between joint and separate training? At test time the model processes the same utterance (no matter whether it is trained jointly or separately), and could in theory be susceptible to both r_1 and r_2. However, the induced representations show fundamentally different behaviors, and remain invariant to r_1 with separate training. A possible explanation is that modern neural networks trained with SGD have a learning bias towards *simple* functions (Shah et al., 2020). When r_1 is not predictive of y_2, relying only on r_2 whilst remaining invariant to r_1 constitutes a simpler function than making use of both r_1 and r_2. As a result, in separate training the model learns to ignore extraneous information, focusing exclusively on r_2. On the contrary, in joint training the target of predicting both y_1 and y_2 forces the hidden states (e.g., h_3) to capture information about both relations, leading to the entanglement problem discussed above.

A Simple Solution Although separate training presents a solution to entanglement, it is unrealistic for real-wold data as it would be extremely inefficient to train separate models for each relation (the number of relations is quadratic with respect to sentence length). Instead, we explore a simple but effective approach where a single model takes as input an utterance enriched with different indicator features for different targets. Specifically, given utterance $[e_1, r_1, e_c, r_2, e_2]$, and assuming we wish to predict relation y_1, we add indicator feature 1 for symbols e_1, r_1, and e_c (marking the relation and its immediate context), and 0 for all other symbols. The model then takes as input the utterance *and* relation indicators, i.e., $[1, 1, 1, 0, 0]$ for y_1 and $[0, 0, 1, 1, 1]$ for y_2, and learns embeddings for indicators during training. It thus learns specialized representations for *each* prediction rather than shared representations for *all* predictions. Based on the simplicity bias, the two representations will guide the model towards exclusively relying on r_1 and r_2, naturally disentangling different relations by encoding them separately. Such a model predicts y_1 with 100% test accuracy and y_2 with 97%.

Discussion Fodor and Pylyshyn (1988) have argued that failure to capture systematicity is a major deficiency of neural architectures, contrasting human learners who can readily apply known grammatical rules to arbitrary novel word combinations to individually memorizing an exponential number of sentences. However, our toy experiment shows that neural networks are not just memorizing sentences but implicitly capturing structure. With separate training or joint training enhanced with the marking strategy, the neural model manages to remain robust to interference from r_1 and properly represent r_2 even for unseen examples, i.e., new compositions of r_1 and r_2. This generalization ability implies that neural models do not need to see all exponential compositions in order to produce plausible representations of them. Instead, with appropriate training and model design, they could uncover and represent the structure underlying systematically related sentences.

3 Learning to Disentangle

While the marking strategy offers substantial benefits in learning disentangled relation representations, we typically do not have access to explicit labels indicating which words are helpful for predicting a specific relation. Nevertheless, the idea

of learning representations specialized for different predictions (albeit with shared parameters) is general and could potentially alleviate the entanglement problem for compositional generalization.

Let $[x_1, x_2, ..., x_n]$ denote a source sequence. Canonical seq2seq models like the Transformer (Vaswani et al., 2017) first encode it into a sequence of contextualized representations which are then used to decode target symbols $[y_1, y_2, ..., y_m]$ one by one. The same source encodings are used to predict all target symbols, and are therefore expected to capture all semantic factors in the input. However, these could be entangled as demonstrated in our analysis above. To alleviate this issue, we propose to learn specialized source representations for different predictions by adaptively re-encoding the source input at every step of the decoding.

Specifically, at the t-th time step, we concatenate the source input with the previously decoded target and obtain the context for the current prediction $C_t = [x_1, x_2, ..., x_n, y_1, ..., y_{t-1}, [\text{PH}]]$ where [PH] is a placeholder (e.g., a mask token when using a pretrained encoder). C_t is then fed to a standard encoder (e.g., the Transformer encoder) to obtain the contextualized representations $H_t = [h_{t,1}, h_{t,2}, ..., h_{t,n}, h_{t,n+1}, ..., h_{t,n+t}]$:

$$H_t = f_{\texttt{Encoder}}(C_t) \tag{1}$$

The key difference from the encoder in standard seq2seq models is that at each time step we adaptively *re-compute* source encodings $H_{t,n} = [h_{t,1}, ..., h_{t,n}]$ that condition on the newly decoded target $[y_1, ..., y_{t-1}]$. This way, target context informs the encoder of predictions of interest at each time step. This simple modification unburdens the model from capturing all source information through a forward pass of encoding. Instead, based on the simplicity bias, the model tends to zero in on information relevant for the current prediction, remaining invariant to irrelevant details, thereby improving disentanglement. One might argue that the decoder in standard seq2seq models could also extract specialized information for each prediction (through the cross attention mechanism). However, it would fail to do so when working with an entangled encoder that produces problematic representations for out-of-distribution examples and breaks down the decoding process.

We propose two strategies for exploiting the target-informed encoder. Firstly, we use a multilayer perceptron (MLP) to predict y_t based on the encoder's output, i.e., the last hidden states $h_{t,n+t}$:

$$p(y_t|x, y_{<t}) = f_{\texttt{MLP}}(h_{t,n+t}) \tag{2}$$

Secondly, we incorporate the proposed encoder into the standard encoder-decoder architecture: we take source encodings $H_{t,n}$ and feed them together with the previous target $[y_1, ..., y_{t-1}]$ to a standard decoder (e.g., Transformer-based) to predict y_t:

$$p(y_t|x, y_{<t}) = f_{\texttt{Decoder}}(H_{t,n}, y_{<t}) \tag{3}$$

For complex tasks like machine translation, preserving the encoder-decoder architecture is essential to achieving good performance.

We adopt the Transformer architecture to instantiate the encoder and decoder, however, the proposed method is generally applicable to any seq2seq model. We maintain separate position encodings for source and target symbols (e.g., x_1 and y_1 correspond to the same position). To differentiate between source and target content, we also add a source(target) type embedding to all source(target) token embeddings. Compared to the classical Transformer, our proposal increases running time from $\mathcal{O}(n^2 + m^2)$ to $\mathcal{O}(m(n^2 + m^2))$ where n is input length and m is output length. Improving the efficiency of our approach is deferred to future work.

4 Experiments: Semantic Parsing

In this section, we present our experiments for evaluating the proposed **D**isent**angle**d seq2seq model which we call DANGLE. We refer to the two variants of DANGLE as DANGLE-ENC and DANGLE-ENCDEC. We first focus on semantic parsing benchmarks which target compositional generalization. Our second suite of experiments reports results on compositional generalization for machine translation.

4.1 Datasets

Our semantic parsing experiments focus on two benchmarks. The first one is COGS (Kim and Linzen, 2020) which contains natural language sentences paired with logical forms based on lambda calculus (see the examples in Table 1). In addition to the standard splits of Train/Dev/Test, COGS provides a generalization (Gen) set that covers five types of compositional generalization: interpreting novel combinations of primitives and grammatical roles, verb argument structure alternation, and

sensitivity to verb class, interpreting novel combinations of modified phrases and grammatical roles, generalizing phrase nesting to unseen depths.

The former three fall into lexical generalization while the latter two require structural generalization. Interpreting novel combinations of modified phrases and grammatical roles involves generalizing from examples with PP modifiers within object NPs to PP modifiers within subject NPs. The generalization of phrase nesting to unseen depths is concerned with two types of recursive constructions: nested CPs (e.g., [*Mary knows that* [*John knows* [*that Emma cooks*]$_{\text{CP}}$]$_{\text{CP}}$]$_{\text{CP}}$) and nested PPs (e.g., *Ava saw the ball* [*in the bottle* [*on the table*]$_{\text{PP}}$]$_{\text{PP}}$). The training set only contains nestings of depth 0–2, where depth 0 is a phrase without nesting. The generalization set contains nestings of strictly greater depths (3–12). The Train set includes 24,155 examples and the Gen set includes 21,000 examples.

Our second benchmark is CFQ (Keysers et al., 2020), a large-scale dataset specifically designed to measure compositional generalization. It contains 239,357 compositional Freebase questions paired with SPARQL queries. CFQ was automatically generated from a set of rules in a way that precisely tracks which rules (atoms) and rule combinations (compounds) were used to generate each example. Using this information, the authors generate three splits with *maximum compound divergence* (MCD) while guaranteeing a small atom divergence between train and test sets. In this dataset atoms refer to entities and relations and compounds to combinations thereof. Large compound divergence indicates the test set contains many examples with unseen syntactic structures. We evaluate our model on all three splits. Each split consists of 95,743/11,968/11,968 train/dev/test examples.

4.2 Comparison Models

On COGS, we trained a baseline TRANSFORMER (Vaswani et al., 2017) with sinusoidal (absolute) and relative position embeddings (Shaw et al., 2018; Huang et al., 2020). We assessed the effect of pretraining on compositional generalization, by also fine-tuning T5-BASE (Raffel et al., 2020) on the same dataset. We created disentangled versions of these models adopting an encoder-only architecture (i.e., +DANGLE-ENC). The pretrained version of our model used ROBERTA (Liu et al., 2019).[2]

We also compared with two models specifically designed for compositional generalization on COGS. The first one is TREE-MAML (Conklin et al., 2021), a meta-learning approach whose objective directly optimizes for out-of-distribution generalization. Their best performing model uses tree kernel similarity to construct meta-train and meta-test task pairs. The second approach is LEXLSTM (Akyurek and Andreas, 2021), an LSTM-based seq2seq model whose decoder is augmented with a lexical translation mechanism that generalizes existing copy mechanisms to incorporate learned, decontextualized, token-level translation rules. The lexical translation module is intended to disentangle lexical phenomena from syntactic ones.

Furrer et al. (2020) showed that pretrained seq2seq models are key to achieving good performance on CFQ. We compared against their T5-11B-MOD model which obtained best results among various pretrained models. This is essentially a T5 model with 11B parameters fine-tuned on CFQ with intermediate representations (i.e., SPARQL queries are simplified to be structurally more aligned to the input for training and then post-processed to obtain the original valid SPARQL at inference time). We also built our model on top of ROBERTA due to the effectiveness of pre-training on this dataset (ROBERTA+DANGLE-ENC), again adopting an encoder-only architecture. To tease apart the effect of pretraining and the proposed approach, we also implemented a baseline that makes use of the ROBERTA-BASE model as the encoder and a vanilla Transformer decoder. The Transformer decoder was initialized randomly and trained from scratch. Finally, we compared against HPD (Guo et al., 2020), a hierarchical poset decoding architecture which consists of three components: sketch prediction, primitive prediction, and traversal path prediction. This model is highly optimized for the CFQ dataset and achieves competitive performance.

We implemented comparison models and DANGLE with fairseq (Ott et al., 2019); for T5-BASE we used HuggingFace Transformers (Wolf et al., 2020). We provide details on model configuration, and various experimental settings in the Appendix.

[2]Note that we use T5-BASE instead of ROBERTA as our pretrained baseline on COGS because in initial experiments we found that having a pretrained *decoder* is critical for good performance, possibly due to the relatively small size of COGS and large vocabulary which includes many rare words.

Model	2		3		4		5	
	CP	PP	CP	PP	CP	PP	CP	PP
TRANSFORMER (abs)	3.4	8.9	1.2	6.6	0.8	5.5	3.1	8.2
+DANGLE-ENC	11.4	5.7	10.3	8.8	14.3	8.6	12.7	13.4
TRANSFORMER (rel)	0.0	0.0	0.0	0.6	0.1	2.5	1.4	4.6
+DANGLE-ENC	13.8	13.5	18.2	19.4	24.7	31.9	27.2	44.3

Table 2: Exact-match accuracy for CP and PP recursion on **different splits of COGS** (recursion depth with $[2-5]$ range).

Model	MCD1	MCD2	MCD3	Mean
T5-11B-MOD	61.6	31.3	33.3	42.1
HPD	72.0	66.1	63.9	67.3
ROBERTA	60.6	33.6	36.0	43.4
+DANGLE-ENC	78.3	59.5	60.4	66.1

Table 3: Exact-match accuracy on **CFQ**, Maximum Compound divergence (MCD) splits.

Model	OSM	CP	PP	Overall
TREE-MAML	0.0	0.0	0.0	66.7
LEXLSTM	0.0	0.0	1.3	82.1
TRANSFORMER (abs)	0.0	3.4	8.9	85.5
+DANGLE-ENC	0.0	11.4	5.7	85.9
TRANSFORMER (rel)	0.0	0.0	0.0	83.3
+DANGLE-ENC	0.0	13.8	13.5	85.4
T5-BASE	0.0	12.5	18.0	85.9
ROBERTA + DANGLE-ENC	0.0	**24.6**	**34.7**	**87.6**

Table 4: Exact-match accuracy on **COGS** by type of structural generalization and overall. OSM refers to generalizing from object modifier PPs to subject modifier PPs; CP and PP are recursion depth generalization for sentential complements and prepositional phrases.

4.3 Results

Table 4 shows our results on COGS broken down by type of structural generalization and overall. All models achieve 0 accuracy on generalizing from PP object modifiers to PP subject modifiers. We find this is due to a predicate order bias. In all training examples, "agent" or "theme" come before preposition predicates like "in", so the models learn this spurious correlation and cannot generalize to cases where the preposition precedes the predicate.

Interestingly, a vanilla TRANSFORMER outperforms more complex approaches like TREE-MAML and LEXLSTM. We conjecture the large discrepancy is mostly due to our use of Glove embeddings, which comparison systems do not use. Pretraining in general substantially benefits lexical generalization, our TRANSFORMER and T5-BASE models achieve nearly perfect accuracy on all such cases in COGS. An intuitive explanation is that pretrained embeddings effectively capture common syntactic roles for tokens of the same type (e.g., "cat" and "dog") and facilitate the generalization of the same decoding strategy to all of them. DANGLE-ENC significantly improves generalization performance on CP and PP recursion when combined with our base TRANSFORMER and ROBERTA.

To further show the potential of our proposal, we evaluated TRANSFORMER+DANGLE-ENC on additional COGS splits. Table 2 shows how model performance changes with exposure to progressively larger recursion depths. Given recursion depth n, we created a split by moving all examples with depth $\leq n$ from Gen to Train set. As can be seen, TRANSFORMER+DANGLE-ENC, especially the variant with relative embeddings, is continuously improving with exposure to additional training examples. In contrast, vanilla TRANSFORMER does not seem to benefit from additional examples, even when relative position encodings are used. We can also explain why adding more recursion in training boosts generalization performance. In the original split, many nouns never occur in examples with recursion depth 2, which could tempt the model to exploit this kind of dataset bias for predictions. In contrast, seeing words in different contexts (e.g., different nesting depth) effectively reduces the possibility of learning these spurious correlations and therefore improves compositional generalization.

CFQ results are shown in Table 3. ROBERTA+DANGLE-ENC substantially boosts the performance of ROBERTA-BASE, and is in fact superior to T5-11B-MOD. This result highlights the limitations of pretraining as a solution to compositional generalization underscoring the benefits of our approach. ROBERTA+DANGLE-ENC is comparable to HPD which is a special-purpose architecture highly optimized for the CFQ dataset. On the contrary, DANGLE is generally applicable to any seq2seq task including machine translation, as we will show in Section 5.

4.4 Analysis

As discussed in Section 2, we hypothesize that a neural model's inability to perform compositional generalization partly arises from its internal representations being entangled. To verify this, we visualize the hidden representations for a TRANSFORMER model with and without DANGLE. Specifically, we train both models on the 4th split of COGS (i.e., data with maximum PP

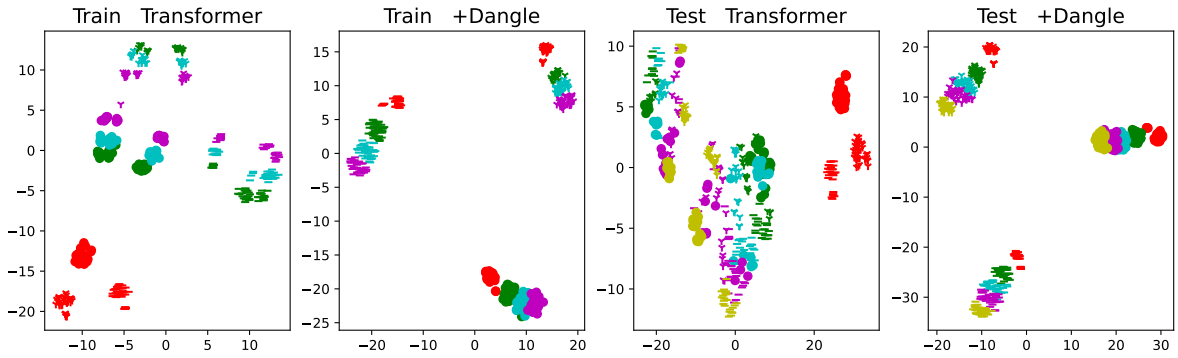

Figure 1: t-SNE visualization of hidden states corresponding to predicates "in", "on", and "beside" on training examples with PP recursion depth 4 and test examples with PP recursion depth 5. Different colors denote different recursion contexts and different shape of markers correspond to different predicates.

recursion depth 4) and test on examples with PP recursion depth 5. Then, we extract the hidden states before the softmax layer used to predict the preposition predicates "in", "beside", and "on" and use t-SNE (van der Maaten and Hinton, 2008) to visualize them. Ideally, the representations of these prepositions should be invariant to the contexts accompanying them so that their prediction is not influenced by distribution shifts (e.g., contextual changes from PP recursion 4 to PP recursion 5).

The visualization is shown in Figure 1. Different colors correspond to different recursion depths while different shape of markers denote different prepositions (e.g., for a training example like "NP in NP in NP in NP in NP", the hidden states corresponding to the four "in" prepositions have the same marker but different colors). In training, TRANSFORMER's hidden states within the same preposition scatter more widely compared to those of DANGLE, which implies that its internal representations conflate information about a preposition's context with itself. In other words, TRANSFORMER's hidden states capture more context variations *in addition to* variations corresponding to the predicate of interest. This in turn causes catastrophic breakdown on the test examples, where TRANSFORMER's hidden states cannot discriminate context from predicate information at all. This is in stark contrast with DANGLE, where information about predicates is preserved even in the presence of unseen contexts.

We further design a metric to quantify entanglement in neural representations drawing inspiration from Kim and Mnih (2018). Their metric assumes the ground-truth factors of a dataset are given, and is applied to images with one factor fixed and all other factors varying randomly; if the representation is perfectly disentangled, the dimension with the lowest variance should correspond to the fixed factor. Since in our setting we do not have access to ground-truth factors, we assume the variable-length target token sequence is the factor of interest. We also do not need to perform a mapping between neurons and factors, because their correspondence is hard-coded in seq2seq models (e.g., a predicate and the hidden units used to predict it).

Model	COGS IntraV	COGS InterV	COGS ↓R	CFQ IntraV	CFQ InterV	CFQ ↓R
TRANSFORMER	0.24	0.64	0.37	0.25	1.13	0.22
+DANGLE-ENC	0.19	0.73	0.26	0.01	0.52	0.01
TRANSFORMER	0.28	0.44	0.63	0.32	1.06	0.30
+DANGLE-ENC	0.23	0.54	0.42	0.04	0.48	0.08

Table 5: Entanglement for TRANSFORMER and our approach (+DANGLE-ENC) on COGS and CFQ (for which both models employ a ROBERTA encoder). Results for training/test set in first/second block. Intra/InterV denotes intra/inter-class variance and R is their ratio.

For each predicate y occurring in different examples e, we extract all corresponding representations $\{\mathbf{v}_{e,y}\}$, i.e., the last layer of the hidden states used to predict y, and compute the empirical variance $\mathrm{Var}_e(\mathbf{v}^i_{e,y})$ for each y; we compute *intra-class* variance as the average of all predicates' variances weighted by their respective frequency:

$$V_{intra} = \frac{1}{d}\sum_{i=1}^{d} \mathrm{E}_y \,\mathrm{Var}_e(\mathbf{v}^i_{e,y}) \quad (4)$$

where d is the dimension of hidden states and E is the weighted average of their variances. Intuitively, if the representations are perfectly disentangled, they should remain invariant to context changes and intra-class variance should be zero.

We also measure *inter-class* variance by taking

Training Set
en: That winter, Taylor barely moved from the fire.
zh: 那年冬天，泰勒几乎没有从大火中挪动过。
Test Set
en: That winter, the dog he liked barely moved from the fire.
zh: 那年冬天，他喜欢的狗狗几乎没有从火堆里挪动过。

Table 6: A training and test example from the CoGnition dataset. The test example is constructed by embedding the synthesized novel compound "the dog he liked" into the template extracted from the training example "That winter, [NP] barely moved from the fire.".

Model	↓ $\text{ErrR}_{\text{Inst}}$	↓ $\text{ErrR}_{\text{Aggr}}$	↑ BLEU
TRANSFORMER (abs)	29.4	63.8	59.4
+DANGLE-ENCDEC	24.4	55.5	59.7
TRANSFORMER (rel)	30.5	63.8	59.4
+DANGLE-ENCDEC	**22.8**	**50.6**	**60.6**

Table 7: BLEU and compound translation error rates (ErrR) on the compositional generalization test set. Subscript Inst denotes instance-wise error rate while Aggr denotes aggregate error over 5 contexts. All results are averaged over 3 random seeds.

the mean of $\mathbf{v}_{e,y}$ for each predicate y and then computing the variance of the means:

$$V_{inter} = \frac{1}{d}\sum_{i=1}^{d} \text{Var}_y \, \text{E}_e(\mathbf{v}^i_{e,y}) \quad (5)$$

Inter-class variance, on the contrary, should be relatively large for these hidden states, because they are intended to capture class variations. The ratio of intra- and inter-class variance collectively measures entanglement.

As shown in Table 5, representations in DANGLE consistently obtain lower intra- to inter-class ratios than baseline models on both COGS and CFQ on both training and test sets.

5 Experiments: Machine Translation

5.1 Dataset

We also applied our approach to CoGnition (Li et al., 2021), a recently released realistic compositional generalization dataset targeting machine translation. This benchmark includes 216K English-Chinese sentence pairs; source sentences were taken from the Story Cloze Test and ROCStories Corpora (Mostafazadeh et al., 2016, 2017) and target sentences were constructed by post-editing the output of a machine translation engine. It also contains a synthetic test set to quantify and analyze compositional generalization of neural MT models. This test set includes 10,800 sentence pairs, which were constructed by embedding synthesized novel compounds into training sentence templates. Table 6 shows an example. Each newly constructed compound is combined with 5 different sentence templates, so that every compound can be evaluated under 5 different contexts.

5.2 Comparison Models

We compared our model to a TRANSFORMER translation model following the same setting and configuration of Li et al. (2021). Again, we experimented with sinusoidal (absolute) and relative position embeddings. We adopted the encoder-decoder architecture variant of our approach (i.e., DANGLE-ENCDEC), as the encoder-only architecture performed poorly possibly due to the complexity of the machine translation task. The number of parameters was kept approximately identical to the TRANSFORMER baseline for a fair comparison. All models were implemented using fairseq (Ott et al., 2019). More modeling details are provided in the Appendix.

5.3 Results

As shown in Table 7, +DANGLE-ENCDEC improves over the base TRANSFORMER model by 1.2 BLEU points when relative position embeddings are taken into account. In addition to BLUE, Li et al. (2021) evaluate compositional generalization using novel compound translation error rate which is computed over instances and aggregated over contexts. +DANGLE-ENCDEC variants significantly reduce novel compound translation errors both across instances and on aggregate by as much as 10 absolute accuracy points (see first two column in Table 7). Across metrics, our results show that +DANGLE-ENCDEC variants handle compositional generalization better than the vanilla. TRANSFORMER model.

5.4 Analysis

Two natural questions emerge given the substantial gain achieved by DANGLE on the compositional generalization (CG) test set: (a) Is this gain related to our treatment of the entanglement problem? and (b) How does entanglement manifest itself in machine translation? We attempt to answer these questions with an example.

In the CG test set, five new utterances are constructed by embedding the novel compound "behind the small doctor on the floor" into five sen-

tence templates. In the training set, the phrases "behind the [ADJ] [NOUN]" and "the [ADJ] [NOUN] on the floor" appear frequently, but the phrase "behind the [ADJ] [NOUN] the [ADJ] [NOUN]" is very rare. This poses a serious challenge for the baseline encoder-decoder model, which mistakenly translates the compound phrase into 地板后面的小医生 (the small doctor behind the floor), or 地板上的小医生 (the small doctor on the floor), or altogether ignores the translation of some content words like 地板后面 (behind the floor). It seems the baseline model cannot simultaneously represent the relation between "behind" and "the small doctor" and the relation between "the small doctor" and "the floor", even though the two are conditionally independent. In contrast, DANGLE generates the correct translation 地板上的小医生后面 in all five contexts. We believe this is due to the proposed adaptive encoding mechanism and its ability to decompose the representation problem of an unfamiliar compound phrase into sub-problems of familiar phrases (i.e, "behind the small doctor" and "the small doctor on the floor").

6 Related Work

The realization that neural sequence models struggle in settings requiring compositional generalization has led to numerous research efforts aiming to understand why this happens and how to prevent it. One line of research tries to improve compositional generalization by adopting a more conventional grammar-based approach (Herzig and Berant, 2021), incorporating a lexicon or lexicon-style alignments into sequence models (Akyurek and Andreas, 2021; Zheng and Lapata, 2021), and augmenting the standard training objective with attention supervision losses (Oren et al., 2020; Yin et al., 2021). Other work resorts to data augmentation strategies as a way of injecting a compositional inductive bias into neural models (Jia and Liang, 2016; Akyürek et al., 2021; Andreas, 2020) and meta-learning to directly optimize for out-of-distribution generalization (Conklin et al., 2021). There are also several approaches which explore the benefits of large-scale pre-trained language models (Oren et al., 2020; Furrer et al., 2020).

In this work we identify the learning of representations which are not disentangled as one of the reasons why neural sequence models fail to generalize compositionally. Disentanglement, i.e., the ability to uncover explanatory factors from data, is often cited as a key property of good representations (Bengio et al., 2013). For example, a model trained on 3D objects might learn factors such as object identity, position, scale, lighting, or colour. Several types of variational autoencoders (Kingma and Welling, 2014) have been proposed for the unsupervised learning of disentangled representations in images (Higgins et al., 2017; Kim and Mnih, 2018; Chen et al., 2018). However, some of the underlying assumptions of these models have come under scrutiny recently (Locatello et al., 2019).

Disentanglement for linguistic representations remains under-explored, and has mostly focused on separating the style of text from its content (John et al., 2019; Cheng et al., 2020). In the context of sentence-level semantics, disentangled representations should be able to discriminate among lexical meanings and semantic relations between words. We highlight the entanglement problem in neural sequence models when trained with explicit factor supervision which, however, does not cover the entire exponential space of compositions for different factors. Instead of encouraging disentanglement with some regularization (Higgins et al., 2017; Kim and Mnih, 2018), we propose a modification to sequence-to-sequence models which achieves this by re-encoding the source based on newly decoded target context. It may be counter-intuitive that we are disentangling by conditioning on more information, but it is feasible thanks to the inherent simplicity bias in neural models.

7 Conclusions

In this paper we proposed an extension to sequence-to-sequence models which allows us to learn disentangled representations for compositional generalization. We have argued that taking into account the target context makes it easier for the encoder to exploit specialized information for improving its predictions. Experiments on semantic parsing and machine translation have shown that our proposal improves compositional generalization without any model, dataset, or task specific modification.

Acknowledgments We thank Chunchuan Lyu, Bailin Wang, and the anonymous reviewers for their useful feedback and Yafu Li for his help with our machine translation experiments. We gratefully acknowledge the support of the European Research Council (award number 681760).

References

Ekin Akyürek, Afra Feyza Akyurek, and Jacob Andreas. 2021. Learning to recombine and resample data for compositional generalization. In *Proceedings of the 9th International Conference on Learning Representations*, Online.

Ekin Akyurek and Jacob Andreas. 2021. Lexicon learning for few shot sequence modeling. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4934–4946, Online. Association for Computational Linguistics.

Jacob Andreas. 2020. Good-enough compositional data augmentation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7556–7566, Online. Association for Computational Linguistics.

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. 2019. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2013. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828.

Ricky T. Q. Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. 2018. Isolating sources of disentanglement in variational autoencoders. In *Advances in Neural Information Processing Systems*, volume 31, pages 2610–2620. Curran Associates, Inc.

Pengyu Cheng, Martin Renqiang Min, Dinghan Shen, Christopher Malon, Yizhe Zhang, Yitong Li, and Lawrence Carin. 2020. Improving disentangled text representation learning with information-theoretic guidance. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7530–7541, Online. Association for Computational Linguistics.

Yong Cheng, Zhaopeng Tu, Fandong Meng, Junjie Zhai, and Yang Liu. 2018. Towards robust neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1756–1766, Melbourne, Australia. Association for Computational Linguistics.

Noam Chomsky. 2014. *Aspects of the Theory of Syntax*, volume 11. MIT press.

Henry Conklin, Bailin Wang, Kenny Smith, and Ivan Titov. 2021. Meta-learning to compositionally generalize. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3322–3335, Online. Association for Computational Linguistics.

Li Dong and Mirella Lapata. 2016. Language to logical form with neural attention. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 33–43, Berlin, Germany. Association for Computational Linguistics.

Catherine Finegan-Dollak, Jonathan K. Kummerfeld, Li Zhang, Karthik Ramanathan, Sesh Sadasivam, Rui Zhang, and Dragomir Radev. 2018. Improving text-to-SQL evaluation methodology. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 351–360, Melbourne, Australia. Association for Computational Linguistics.

Jerry A Fodor and Zenon W Pylyshyn. 1988. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1-2):3–71.

Gottlob Frege. 1884. *Die Grundlagen der Arithmetik (The Foundations of Arithmetic): eine logisch- mathematische Untersuchung ber den Begriff der Zahl*. W. Koebner, Breslau. Reprint published by: Georg Olms, Hildesheim, 1961.

Daniel Furrer, Marc van Zee, Nathan Scales, and Nathanael Schärli. 2020. Compositional generalization in semantic parsing: Pre-training vs. specialized architectures. *arXiv preprint arXiv:2007.08970*.

Yinuo Guo, Zeqi Lin, Jian-Guang Lou, and Dongmei Zhang. 2020. Hierarchical poset decoding for compositional generalization in language. In *Advances in Neural Information Processing Systems*, volume 33, pages 6913–6924. Curran Associates, Inc.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.

Jonathan Herzig and Jonathan Berant. 2021. Span-based semantic parsing for compositional generalization. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 908–921, Online. Association for Computational Linguistics.

Irina Higgins, Loïc Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. 2017. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *Proceedings of the 5th International Conference on Learning Representations*, Toulon, France.

Zhiheng Huang, Davis Liang, Peng Xu, and Bing Xiang. 2020. Improve transformer models with better relative position embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3327–3335, Online. Association for Computational Linguistics.

Robin Jia and Percy Liang. 2016. Data recombination for neural semantic parsing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12–22, Berlin, Germany. Association for Computational Linguistics.

Vineet John, Lili Mou, Hareesh Bahuleyan, and Olga Vechtomova. 2019. Disentangled representation learning for non-parallel text style transfer. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 424–434, Florence, Italy. Association for Computational Linguistics.

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. 2020. Measuring compositional generalization: A comprehensive method on realistic data. In *Proceedings of the 8th International Conference on Learning Representations*, Addis Ababa, Ethiopia.

Hyunjik Kim and Andriy Mnih. 2018. Disentangling by factorising. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2649–2658, Stockholmsmässan, Stockholm Sweden. PMLR.

Najoung Kim and Tal Linzen. 2020. COGS: A compositional generalization challenge based on semantic interpretation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9087–9105, Online. Association for Computational Linguistics.

Diederik P. Kingma and Max Welling. 2014. Auto-encoding variational Bayes. In *Proceedings of the 2nd International Conference on Learning Representations*, Banff, AB, Canada.

Brenden M. Lake and Marco Baroni. 2018. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2879–2888. PMLR.

Yafu Li, Yongjing Yin, Yulong Chen, and Yue Zhang. 2021. On compositional generalization of neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4767–4780, Online. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized bert pretraining approach.

Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. 2019. Challenging common assumptions in the unsupervised learning of disentangled representations. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 4114–4124, Long Beach, California, USA. PMLR.

Gary F. Marcus. 2003. *The Algebraic Mind: Integrating Connectionism and Cognitive Science*. MIT press.

Richard Montague. 1970. Universal grammar. *Theoria*, 36(3):373–398.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 839–849, San Diego, California. Association for Computational Linguistics.

Nasrin Mostafazadeh, Michael Roth, Annie Louis, Nathanael Chambers, and James Allen. 2017. LSDSem 2017 shared task: The story cloze test. In *Proceedings of the 2nd Workshop on Linking Models of Lexical, Sentential and Discourse-level Semantics*, pages 46–51, Valencia, Spain. Association for Computational Linguistics.

Inbar Oren, Jonathan Herzig, Nitish Gupta, Matt Gardner, and Jonathan Berant. 2020. Improving compositional generalization in semantic parsing. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2482–2495, Online. Association for Computational Linguistics.

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

Barbara Partee. 1995. Lexical semantics and compositionality. In Leila Gleitman and Mark Liberman, editors, *Invitation to Cognitive Science Part I: Language*. MIT Press, Cambridge, MA.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. 2020. Distributionally robust neural networks. In *Proceedings of the 8th International Conference on Learning Representations*, Addis Ababa, Ethiopia.

Harshay Shah, Kaustav Tamuly, Aditi Raghunathan, Prateek Jain, and Praneeth Netrapalli. 2020. The pitfalls of simplicity bias in neural networks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9573–9585. Curran Associates, Inc.

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, Volume 2 (Short Papers)*, pages 464–468, New Orleans, Louisiana. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Pengcheng Yin, Hao Fang, Graham Neubig, Adam Pauls, Emmanouil Antonios Platanios, Yu Su, Sam Thomson, and Jacob Andreas. 2021. Compositional generalization for neural semantic parsing via span-level supervised attention. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2810–2823, Online. Association for Computational Linguistics.

Hao Zheng and Mirella Lapata. 2021. Compositional generalization via semantic tagging. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1022–1032, Punta Cana, Dominican Republic. Association for Computational Linguistics.

A Model Configuration: Semantic Parsing Experiments

In these sections, we describe the configuration of the models evaluated in the experiments of Sections 4 and 5, respectively.

On COGS, the small in-distribution development (Dev) set makes model selection extremely difficult and non-reproducible. We follow Conklin et al. (2021) and sample a small subset from the generalization (Gen) set denoted as 'Gen-Dev' for tuning hyper-parameters. Best hyper-parameters were used to rerun the model with 5 different random seeds for reporting final results on the Gen set. For the baseline TRANSFORMER, the layer number of encoder and decoders are both 2. The embedding dimension is 300. The feedforward embedding dimension is 512. For TRANSFORMER+DANGLE, to maintain approximately identical model size with the baseline, we used the same embedding dimension and set the number of the encoding layers to 4. For both models, we initialized embeddings (on the both source and target side) with Glove (Pennington et al., 2014).

On COGS, for the ROBERTA+DANGLE model, we share the target vocabulary and embedding matrix with the source. On CFQ, we use a separate target vocabulary; the target embedding matrix is randomly initialized and learned from scratch. ROBERTA-BASE on CFQ is combined with a Transformer decoder that has 2 decoder layers with embedding dimension 256 and feedforward embedding dimension 512. All hyper-parameters are chosen based on validation performance. On CFQ, for both ROBERTA-BASE and ROBERTA+DANGLE, results are averaged over 3 randoms seeds.

B Model Configuration: Machine Translation Experiments

We followed the same setting of Li et al. (2021) and adopted a TRANSFORMER translation model consisting of a 6-layer encoder and a 6-layer decoder with hidden size 512. Each training batch includes 8,191 tokens at maximum. This model was trained for 100,000 steps and we chose the best checkpoint on the validation set for evaluation. Again, we experimented with sinusoidal (absolute) and relative position embeddings.

We used the same hyperparameters as the baseline model except for the number of layers which we tuned on the validation set; for relative position embeddings, the encoder has 4 vanilla source-only Transformer encoder layers on top of 4 target-informed Transformer encoder layers (i.e., 8 encoder layers in all) and the decoder has 4 Transformer decoder layers; for absolute position embeddings, the encoder has 4 vanilla source-only Transformer encoder layers on top of 2 target-informed Transformer encoder layers and the decoder has 6 Transformer decoder layers. For a fair comparison, we also experimented with 8 encoder layers and 4 decoder layers for the baseline TRANSFORMER, and found that it performs similarly to the standard 6-layer architecture.