Can Constructions "SCAN" Compositionality ?

Ganesh Katrapati and **Manish Shrivastava**
International Institute of Information Technology Hyderabad
ganesh.katrapati@research.iiit.ac.in m.shrivastava@iiit.ac.in

Abstract

Sequence to Sequence models struggle at compositionality and systematic generalisation even while they excel at many other tasks. We attribute this limitation to their failure to internalise *constructions*—conventionalised form–meaning pairings that license productive recombination. Building on these insights, we introduce an unsupervised procedure for mining *pseudo-constructions*: variable-slot templates automatically extracted from training data. When applied to the SCAN dataset, our method yields large gains out-of-distribution splits: accuracy rises to **47.8%** on ADD JUMP and to **20.3%** on AROUND RIGHT without any architectural changes or additional supervision. The model also attains competitive performance with $\leq 40\%$ of the original training data, demonstrating strong data efficiency. Our findings highlight the promise of construction-aware preprocessing as an alternative to heavy architectural or training-regime interventions.

1 Introduction

Compositionality is the principle that the meaning of a complex expression is determined by the meanings of its parts and the rules used to combine them (Fodor and Pylyshyn, 1988; Marcus, 2003; Partee et al., 1990). It enables systematic generalisation: the ability to understand and produce novel combinations of familiar elements, a hallmark of human language competence.

Despite the impressive empirical performance of sequence to sequence models such as RNNs, LSTMs, and Transformers, studies have consistently found that they struggle with tasks requiring compositional generalisation (Lake and Baroni, 2018; Hupkes et al., 2020; Keysers et al., 2020). When faced with inputs that combine known primitives in unseen ways, these models frequently fail to extrapolate correctly.

Cognitive and Construction Grammar treat *constructions* as form–meaning pairs composed of conventionalised components that combine with lexical items (Goldberg, 1995; Langacker, 1987; Croft, 2001). For successful communication, speakers must have access to these conventionalised constructions shared within their linguistic community. The degree of conventionalisation varies across construction types: for example, idiomatic expressions like "kick the bucket" are fully fossilised and resist internal modification, whereas partially filled constructions such as "the Xer the Yer" contain open slots that can be flexibly filled to produce complete surface forms (Fillmore et al., 1988; Goldberg, 2006).

Inspired by this notion, we propose that modelling constructions is essential to solving the problem of compositionality. We choose the SCAN dataset - a canonical testbed for evaluating compositionality in neural models - to demonstrate our approach. We introduce a simple yet effective method of mining *pseudo-constructions* and show that models trained on segmented data achieve significant improvements over standard baselines on SCAN's ADD JUMP and AROUND RIGHT splits.

Furthermore, we demonstrate strong data efficiency: by leveraging the compositional structure, our method requires substantially less data to achieve competitive performance, especially on simpler splits. Our results suggest that carefully exposing compositional patterns during training can yield robust improvements without resorting to complex interventions.

2 Related Work

There have been a number of benchmarks and tasks to evaluate whether modern NLP methods including deep neural networks such as RNNs (Elman, 1990), LSTMs (Hochreiter and Schmidhuber,

Proceedings of the Second International Workshop on Construction Grammars and NLP, pages 165–171

1997) and Transformers (Vaswani et al., 2017) exhibit compositional behaviour. *SCAN* (Lake and Baroni, 2018), *COGS* (Kim and Linzen, 2020) , *CFQ* (Keysers et al., 2020), *PCFG* (Hupkes et al., 2020) and similar benchmarks focus on sequence prediction tasks where input sequence must be processed in a compositional manner to yield the correct sequence on the target side.

They showed that the models do *not* generalise systematically: when confronted with new combinations of words or phrases that were absent from the training data, their performance breaks down. Subsequent studies on a variety of datasets (Li and colleagues, 2021; Sinha et al., 2019; Liška et al., 2018), have reported similar findings. Informed by these limitations, recent work has led to multiple methods to improve compositional generalisation abilities of neural network models.

Multiple studies have focused on disentangling syntax and semantics - (Russin et al., 2019) introduced a dedicated syntactic channel boosts SCAN accuracy dramatically , separating primitive–function pathways pushes performance to near-perfect levels (Li et al., 2019; Jiang and Bansal, 2021). Rather than separating syntax and semantics, some studies have focused on syntactic guidance. (Hupkes et al., 2019; Baan et al., 2019; Kim et al., 2021; Zanzotto et al., 2020).

Data-centric approaches improve compositionality by augmenting the training corpus with systematically recombined examples: GECA (Andreas, 2020), automatically mined lexical symmetries (LEXSYM; Akyürek and Andreas, 2022), and grammar-based generators such as CSL (Qiu et al., 2022) all substantially cut error rates on SCAN, COGS, and CLEVR. Herzig et al. (2021) insert a reversible or lossy intermediate representation between the input and the target program, doubling accuracy on CFQ MCD splits and adding 15–20 points on text-to-SQL.

Treating compositionality as a transferable skill, Meta learning approaches (Zhu et al., 2021; Lake, 2019; Lake and Baroni, 2023) push transformers beyond 70 % accuracy on the hardest SCAN and COGS splits.

Apart from this, several studies have proposed significant modifications to the neural network architecture (Csordás et al., 2022; Huang et al., 2024) and neural–symbolic designs such as NMN, MAC, NLM, LANE, program-synthesis grammars, and the Neural-Symbolic Recursive Machine (Andreas et al., 2017; Hudson and Manning, 2018; Dong et al., 2019; Liu et al., 2022; Nye et al., 2020; Li et al., 2022) which achieve (near-)perfect compositional generalisation on datasets like SCAN, COGS and CFQ.

While many of these approaches achieve near-perfect accuracy in datasets like SCAN and COGS, they either require data augmentation, which likely translates into training bigger models for a longer time, or they propose drastic architectural changes which have not been proven to scale beyond these benchmarks. Our method does not employ data augmentation or complex architectural changes. Our aim is show that taking insights from Cognitive Grammar and the notion of *Constructions* leads to building models more capable of compositional generalisation.

Recent work on integrating Construction Grammar (CxG) with neural models has been encouraging: fine-tuning BERT on construction-annotated corpora sharpens its encoding of construction identity and slot fillers (Tayyar Madabushi et al., 2020), a Mandarin CxLM leverages more than ten-thousand schemata to boost cloze accuracy (Tseng et al., 2022). Yet no study has directly shown that construction-aware training itself improves systematic compositional generalisation on classic out-of-distribution tests and bridging that gap remains a challenge.

3 Data

Introduced by (Lake and Baroni, 2018), *SCAN* contains pairs of simple navigation commands with action sequences; primitives like “jump” map to “`I_JUMP`”, while modifiers such as “left”, “right”, “opposite”, and “around” compose these primitives into longer actions.

The original paper showed that models excel on a random split yet falter on novel combinations. In the ADD JUMP split, models see the primitive “jump” during training but must execute composed forms (e.g., “jump twice”) at test time. Loula et al. (2018) extended this with the AROUND RIGHT split: training includes “walk left”, “walk right”, “jump around left”, and so on, while testing requires generalising to “jump around right”, forcing the model to learn that “around” modifies directions and that “left” and “right” are symmetric.

We focus on improving the accuracy for both these splits.

4 Approach

Definition (Pseudo-construction). A *pseudo-construction* is a partially specified template induced from training data, containing fixed words alongside one or more *slots* represented by placeholders (e.g., _ or W_n). Unlike fully conventionalised constructions, pseudo-constructions are derived automatically and capture recurring structural patterns that can generalise to novel inputs when the slots are filled with appropriate lexical items.

4.1 Mining Pseudo-constructions

A SCAN train or test set consists of both a source file, which consists of commands ("jump") and a target file which consists of actions ("I_JUMP"). Given a SCAN split, we take the source file of the training set, and follow a series of steps to obtain partially filled pseudo-constructions.

- **Extracting Candidates:** For every sentence in the train source file, we extract spans of up to length of 4 tokens and add them to the candidate list. We also generate masked spans in which one or more non-consecutive words are replaced by the the token "_", effectively forming a *slot* in a partially filled pseudo-construction. The candidates are then ranked according to their probabilities.

- **Beam Decoding:** We use beam search to segment an input sentence into the best scoring sequence of pseudo-constructions and words. Test source files are not used for mining pseudo-constructions. They are segmented only using the ones induced from the training set.

- **Encouraging Alignment with Target**: Partially filled pseudo-constructions like '_ around _ twice" are advantageous because the same template applies for any fully filled variant - a simple word replacement on the target side works well. However, simple masking also produces "look _ left _" which produces widely different targets for different values of _ and _. Consider,

  look *around* left → I_TURN_LEFT I_LOOK I_TURN_LEFT I_LOOK I_TURN_LEFT I_LOOK I_TURN_LEFT I_LOOK

  look *opposite* left → I_TURN_LEFT I_TURN_LEFT I_LOOK

  To discourage picking candidates like the latter one, we compute an *alignment distance* between the candidate and its equivalent on the target side. For each candidate P, gather the set of source sentences $\mathcal{S}(P) = \{s_1, s_2, \ldots, s_n\}$ in which the pattern occurs, with each source sentence s_i paired to a target sentence t_i. For every $s_i \in \mathcal{S}(P)$, calculate its Levenshtein (edit) distance to every other s_j $(j \neq i)$ in the same set and select the *nearest neighbour*, $\mathrm{NN}(s_i)$—the source sentence that minimises this distance. Let (t_i, t_j) denote the target sentences aligned with $(s_i, \mathrm{NN}(s_i))$.

  Define

  $$\Delta_i = |\mathrm{len}(t_i) - \mathrm{len}(t_j)|$$

  as the absolute difference in their word counts. The resulting *misalignment score* (MS) for pattern P is the average of these differences:

  $$MS(P) = \frac{1}{|\mathcal{S}(P)|} \sum_{i=1}^{|\mathcal{S}(P)|} \Delta_i.$$

  A lower misalignment score indicates that source sequences are more aligned to the target sequences. A pseudo-construction has a low misalignment score when swapping different words into its slots still produces target sentences that look much the same. We add this score as a penalty to the beam search to pick candidates which are more aligned.

Once the source files (train and test) are segmented, we prepare the data for the next stage. For every sentence in the source files, we replace the underscores with slot tokens such as W_n where n refers to the slot number. We save the mapping between the slot tokens and the original words.

The SCAN data consists singleton rules such as *jump* → I_JUMP. We treat this as a bidirectional lexicon. Whenever a token in a target sentence appears in the lexicon, we lookup the source word and then replace it with the associated slot token. For example:

4.2 Training

We use the sequence to sequence transformer architecture as the base model for training purposes, and use the JoeyNMT toolkit (Kreutzer et al., 2019) to train all the models. The model architecture

has an encoder and a decoder each with 4 layers and 4 attention heads with embedding size of 256 and the feed forward layer with the size of 1024. The models are trained for 30 epochs using the NOAM scheduler (Vaswani et al., 2017). Prior to evaluation, we swap back the slot tokens predicted sequence through the mapping saved earlier.

5 Results

The performance on both the splits (ADD JUMP, AROUND RIGHT) is significantly better than the baseline transformer (1) which indicates that we have succeeded in encoding a degree of generalisation through the pseudo-constructions. Overall, they capture reusable structure absent from the flat surface strings, enabling the model to generalise compositionally.

6 Data Efficiency

Compositionality theory posits that exploiting compositional structure enables grasping abstract patterns from far fewer training examples than treating data only at the surface level (Chomsky (1957), Chomsky (1965), Fodor and Pylyshyn (1988)). We test this by training models with smaller samples of the SCAN splits.

After segmentation of the training source file, each sentence is transformed into a series of pseudo-constructions in such a way that multiple sentences might fall into the same resultant sentence type.

look opposite left twice and walk twice → `( W_1 opposite W_2 twice ) and (W_3 thrice)`

jump opposite right twice and run twice → `( W_1 opposite W_2 twice ) and (W_3 thrice)`

To assess the data efficiency of our method we constructed *sentence type–balanced training subsets*, retaining every sentence type but varying the per-type quota $k \in \{1, 3, 5, 10, 25\}$. This produces monotonic subsamples ranging from 5 % to 60 % of the original corpus while guaranteeing full coverage. (Table 1).

On the ADD JUMP split, with only $k = 10$ examples per type—approximately *39 %* of the full training data—the model attains *40.7 %* accuracy, not far from the *47.8 %* trained on entire set.

For the AROUND RIGHT at the same $k = 10$ mark the model reaches merely *9.4 %*, less than half of the *20.4 %* full-data accuracy, and increasing to $k = 25$ (60 %) yields only a marginal gain to *10.2 %*. This pronounced gap reflects the split's higher compositional complexity: mastering the nested "around DIR" construction with repetition operators may require substantially more evidence than the shallow "add jump" pattern.

Split	k / type	Acc. (%)	Size	%
Around Right	1	2.77	741	5
	3	5.98	2,042	13
	5	8.66	3,166	21
	10	9.38	5,423	36
	25	10.18	9,175	60
	Full	20.73	15,225	100
Add Jump	1	12.79	666	5
	3	20.50	1,972	13
	5	29.44	3,178	22
	10	40.69	5,660	39
	Full	47.81	14,670	100

Table 1: Accuracy as the training set is reduced to k examples per type.

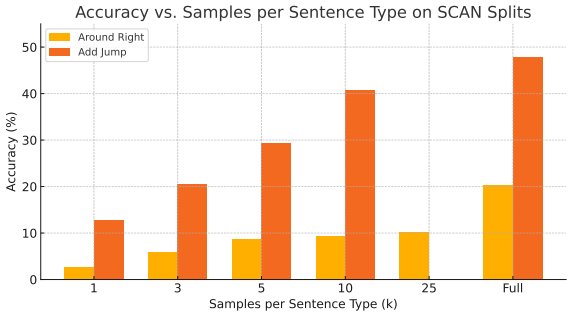

Figure 1: Accuracy versus percentage of full training data for the AROUND RIGHT and ADD JUMP SCAN splits.

The pseudo-construction bias confers strong sample-efficiency benefits on syntactically simple splits (ADD JUMP), but this may not scale to harder generalisation problems (AROUND RIGHT).

7 Conclusion

While we define pseudo-constructions operationally as automatically mined templates, they can be seen as computational approximations to Construction Grammar's notion of conventionalised form–meaning pairings. Unlike fully fossilised or community-shared constructions, pseudo-constructions are data-driven and context-specific, yet they capture structural regularities that support compositional generalisation. Thus, while our primary aim is methodological, the results also lend indirect support to the constructionist hypothesis

that access to reusable schematic patterns is crucial for systematic generalisation. We leave a fuller exploration of their linguistic plausibility and theoretical integration to future work.

A deeper look into errors showed us that our method for finding pseudo-constructions can make several mistakes. For instance, while at first sight "turn around right" and "walk around right" seem to follow the same pattern, their corresponding outputs can vary significantly - this can lead to confusion and failure if the word "turn" is masked away.

We call for more robust approaches into finding constructions in text and for future work into deeper integration of construction processing into neural models.

References

Emre Alp Akyürek and Jacob Andreas. 2022. Lexsym: Discovering and exploiting lexical symmetries for compositional generalization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Jacob Andreas. 2020. Good-enough compositional data augmentation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. 2017. Neural module networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Joost Baan, Dieuwke Hupkes, and Willem Zuidema. 2019. Inspecting the inductive biases of rnns with attentive guidance. In *Proceedings of the 2019 Workshop on Analyzing and Interpreting Neural Networks for NLP (BlackboxNLP)*.

Noam Chomsky. 1957. *Syntactic Structures*. Mouton, The Hague.

Noam Chomsky. 1965. *Aspects of the Theory of Syntax*. MIT Press, Cambridge, MA.

William Croft. 2001. *Radical Construction Grammar: Syntactic Theory in Typological Perspective*. Oxford University Press, Oxford, UK.

Róbert Csordás, Toma Gruber, and Marc Henniges. 2022. Neural data routing: Enforcing compositionality with geometric attention. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*.

Honghua Dong, Jiayuan Mao, Jiajun Lin, Chuang Wang, Lihong Li, Dengyong Zhou, and Yuncheng Song. 2019. Neural logic machines. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*.

Jeffrey L. Elman. 1990. Finding structure in time. *Cognitive Science*, 14(2):179–211.

Charles J. Fillmore, Paul Kay, and Mary Catherine O'Connor. 1988. Regularity and idiomaticity in grammatical constructions: The case of let alone. In Timothy Shopen, editor, *Language Typology and Syntactic Description, Vol. 1*, pages 501–538. Cambridge University Press.

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

Adele E. Goldberg. 1995. *Constructions: A Construction Grammar Approach to Argument Structure*. University of Chicago Press, Chicago, IL.

Adele E. Goldberg. 2006. *Constructions at Work: The Nature of Generalization in Language*. Oxford University Press, Oxford, UK.

Jonathan Herzig, Peter Shaw, Ming-Wei Chang, Kelvin Guu, Panupong Pasupat, and Yuan Zhang. 2021. Unlocking compositional generalization in pre-trained models using intermediate representations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Zihan Huang, Yao Wang, Qian Wu, and Maosong Sun. 2024. Cat: A compositionally aware transformer with multi-primitive decomposition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*.

Drew A. Hudson and Christopher D. Manning. 2018. Compositional attention networks for machine reasoning. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Dieuwke Hupkes, Verna Dankers, Mathijs Mul, and Elia Bruni. 2020. Compositionality decomposed: How do neural networks generalise? *Journal of Artificial Intelligence Research*, 67:757–795. Dataset introduced: *PCFG-SET*.

Dieuwke Hupkes, Sara Veldhoen, and Willem Zuidema. 2019. Compositional generalization for neural sequence learning via attentive guidance. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Zhengxuan Jiang and Mohit Bansal. 2021. CGPS-transformer: Compositional generalization by auxiliary sequence prediction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

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 (ICLR)*. Dataset introduced: *CFQ*.

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.

Youngjin Kim, Daniel Keysers, Nathanael Schärli, Daniel Furrer, and Olivier Bousquet. 2021. Structural guidance for transformer self-attention: Hard and soft masking for compositional generalization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Julia Kreutzer, Jasmijn Bastings, and Stefan Riezler. 2019. Joey NMT: A minimalist NMT toolkit for novices. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 109–114, Hong Kong, China. Association for Computational Linguistics.

Brenden M. Lake. 2019. Compositional generalization through meta sequence-to-sequence learning. In *Advances in Neural Information Processing Systems (NeurIPS) 32*.

Brenden M. Lake and Marco Baroni. 2018. Generalization without systematicity: Strong tests for compositionality in humans and machines. In *Proceedings of the 6th International Conference on Learning Representations (ICLR)*. Dataset introduced: SCAN.

Brenden M. Lake and Marco Baroni. 2023. Meta-in-context learning induces human-like compositional generalization. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Ronald W. Langacker. 1987. *Foundations of Cognitive Grammar, Volume 1: Theoretical Prerequisites*. Stanford University Press, Stanford, CA.

Jialin Li, Shuwen Lu, Yang Liu, and Mohit Bansal. 2022. Neural-symbolic recursive machines for systematic generalization. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Ruixiang Li, Yichao Chen, Sheng Lin, and Marco Baroni. 2019. CGPS-rss: Separating primitive and functional channels improves compositional generalization. In *Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS) — Compositionality Workshop*.

X. Li and colleagues. 2021. Cognition: A dataset for testing compositional generalisation in neural machine translation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages xx–yy, Online. Association for Computational Linguistics. Dataset introduced: *COGNITION*.

Adam Liška, Germán Kruszewski, and Marco Baroni. 2018. Memorize or generalize? searching for a compositional RNN in a haystack. *arXiv preprint arXiv:1802.06467*.

Cang Liu, Pengcheng Zhou, Zhenzhong Lan, and Yang Wang. 2022. Lane: Learning analytical expressions for systematic generalization. In *Advances in Neural Information Processing Systems (NeurIPS) 35*.

João Loula, Marco Baroni, and Brenden Lake. 2018. Rearranging the familiar: Testing compositional generalization in recurrent networks. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 108–114, Brussels, Belgium. Association for Computational Linguistics.

Gary F. Marcus. 2003. *The Algebraic Mind: Integrating Connectionism and Cognitive Science*. MIT Press, Cambridge, MA.

Maxwell Nye, Armando Solar-Lezama, and Joshua B. Tenenbaum. 2020. Compositional generalization by program synthesis. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*.

Barbara H. Partee, Alice ter Meulen, and Robert E. Wall. 1990. Compositionality. In *Mathematical Methods in Linguistics*, pages 319–334. Springer.

Chen Qiu, Yutong Yu, Emre Alp Akyürek, and Jacob Andreas. 2022. Csl: Compositional structure learner for data augmentation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Jacob Russin, Jianshu Jo, and Randall C. O'Reilly. 2019. Compositional generalization in sequence-to-sequence models via syntactic attention. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Koustuv Sinha, Shagun Sodhani, Jin Dong, Joelle Pineau, and William L. Hamilton. 2019. CLUTRR: A diagnostic benchmark for inductive reasoning from text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4506–4515, Hong Kong, China. Association for Computational Linguistics.

Harish Tayyar Madabushi, Laurence Romain, Dagmar Divjak, and Petar Milin. 2020. CxGBERT: BERT meets construction grammar. In *Proceedings of the*

28th International Conference on Computational Linguistics, pages 4020–4032, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Yu-Hsiang Tseng, Cing-Fang Shih, Pin-Er Chen, Hsin-Yu Chou, Mao-Chang Ku, and Shu-Kai Hsieh. 2022. CxLM: A construction and context-aware language model. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 6361–6369.

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 (NeurIPS) 30*, pages 5998–6008.

Fabio Massimo Zanzotto, Andrea Santilli, Leonardo Ranaldi, Dario Onorati, Pierfrancesco Tommasino, and Francesca Fallucchi. 2020. KERMIT: Complementing transformer architectures with encoders of explicit syntactic interpretations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 256–267, Online. Association for Computational Linguistics.

Shuyan Zhu, Zhengxuan Jiang, Ruixiang Li, and Mohit Bansal. 2021. DUEL: Transferable compositional inductive bias via meta-learning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.