Translate First Reorder Later: Leveraging Monotonicity in Semantic Parsing

Francesco Cazzaro[*] **Davide Locatelli**[*] **Ariadna Quattoni**
Universitat Politècnica de Catalunya
name.lastname@upc.edu

Xavier Carreras
dMetrics
xavier.carreras@dmetrics.com

Abstract

Prior work in semantic parsing has shown that conventional seq2seq models fail at compositional generalization tasks. This limitation led to a resurgence of methods that model alignments between sentences and their corresponding meaning representations, either implicitly through latent variables or explicitly by taking advantage of alignment annotations. We take the second direction and propose TPOL, a two-step approach that first translates input sentences monotonically and then reorders them to obtain the correct output. This is achieved with a modular framework comprising a *Translator* and a *Reorderer* component. We test our approach on two popular semantic parsing datasets. Our experiments show that by means of the monotonic translations, TPOL can learn reliable lexico-logical patterns from aligned data, significantly improving compositional generalization both over conventional seq2seq models, as well as over other approaches that exploit gold alignments. Our code is publicly available at https://github.com/interact-erc/TPol.git

1 Introduction

The goal of a semantic parser is to map natural language sentences (NLs) into meaning representations (MRs). Most current semantic parsers are based on deep sequence-to-sequence (seq2seq) approaches and presume that it is unnecessary to model token alignments between NLs and MRs because the attention mechanism can automatically learn the correspondences (Dong and Lapata, 2016; Jia and Liang, 2016). However, recent work has shown that such seq2seq models find compositional generalization challenging, i.e., they struggle to predict unseen structures made up of components observed at training (Lake and Baroni, 2018; Finegan-Dollak et al., 2018).

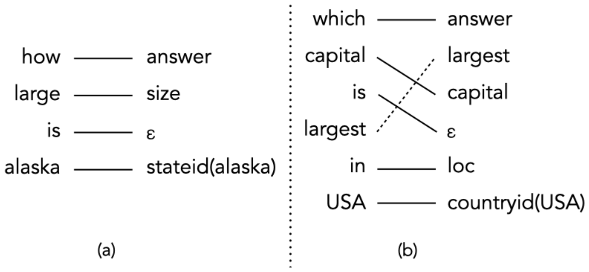

Figure 1: Examples from the GEOALIGNED dataset. (a) is a monotonic alignment, (b) is non-monotonic.

This limitation motivated the resurgence of approaches that model alignments between NL sentences and their corresponding MRs more similarly to classical grammar and translation-based parsers (Herzig and Berant, 2021). Alignments can be modeled either implicitly through latent variables (Wang et al., 2021), or explicitly by leveraging gold alignment annotations (Shi et al., 2020; Liu et al., 2021a). We take the second direction and exploit a recently released multilingual dataset for semantic parsing annotated with word alignments: GEOALIGNED (Locatelli and Quattoni, 2022), which augments the popular GEO benchmark (Zelle and Mooney, 1996).

Figure 1 shows some examples of the annotations provided. One key observation is that a significant percentage of the alignments are monotonic, i.e., they require no reordering of the target MR (Figure 1a), as opposed to non-monotonic alignments (Figure 1b). This suggests that learning reliable lexico-logical translation patterns from aligned data should be possible. If there are simple patterns, shouldn't an ideal model be able to exploit them?

With this in mind, we propose TPOL, a Two-step Parsing approach that leverages monotonic translations. TPOL introduces a modular framework with two components: a *Monotonic Translator* and a *Reorderer*. The Translator is trained from pairs of NLs and MRs, where the MRs have been permuted to be monotonically aligned. Hence,

[*] Equal contribution

Findings of the Association for Computational Linguistics: EACL 2023, pages 227–238
May 2-6, 2023 ©2023 Association for Computational Linguistics

the Translator's output will be an MR whose order might not correspond to that of the gold truth. For this reason, the Reorderer is trained to restore the correct order of the original MR.

Our experiments on GEOALIGNED demonstrate that compared to a multilingual BART model (Liu et al., 2020), TPOL achieves similar performance on the random test split but significantly outperforms on the compositional split across all languages. For example, on the query split in English, mBART obtains 69.4% in exact-match accuracy and TPOL obtains 87.8%. This result also improves on the 74.6% obtained by SPANBASED (Herzig and Berant, 2021), another approach that leverages alignment annotations.

Because most semantic parsing datasets do not contain alignment information, we experiment with alignments generated automatically. On GEO, TPOL trained with automatic alignments still outperforms mBART, and in particular on the English query split it improves by almost 10 points. Furthermore, we show competitive results on the popular SCAN dataset (Lake and Baroni, 2018).

In summary, the main contributions of this paper are:

1. We propose TPOL, a modular two-step approach for semantic parsing which explicitly leverages monotonic alignments;

2. Our experiments show that TPOL improves compositional generalization without compromising overall performance;

3. We show that even without gold alignments TPOL can achieve competitive results.

2 Related Work

Recently, the semantic parsing community has raised the question of whether current models can generalize compositionally, along with an effort to test for it (Lake and Baroni, 2018; Finegan-Dollak et al., 2018; Kim and Linzen, 2020). The consensus is that conventional seq2seq models struggle to generalize compositionally (Loula et al., 2018; Keysers et al., 2020). Moreover, large pre-trained language models have been shown not to improve compositional generalization (Oren et al., 2020; Qiu et al., 2022b). This has prompted the community to realize that parsers should be designed intentionally with compositionality in mind (Lake, 2019; Gordon et al., 2020; Weißenhorn et al., 2022). It has also been pointed out that compositional architectures are often designed for synthetic datasets and that compositionality on non-synthetic data is under-tested (Shaw et al., 2021).

Data augmentation techniques have been proposed to improve compositional generalization (Andreas, 2020; Yang et al., 2022; Qiu et al., 2022a). Another strategy is to exploit some level of word alignments. In general, there has been a resurgent interest in alignments as it has been shown that they can be beneficial to neural models (Shi et al., 2020). It has also been conjectured that the lack of alignment information might hamper progress in semantic parsing (Zhang et al., 2019). As a result, the field has seen some annotation efforts in this regard (Shi et al., 2020; Herzig and Berant, 2021; Locatelli and Quattoni, 2022).

Alignments have been modeled implicitly: Wang et al. (2021) treat alignments as discrete structured latent variables within a neural seq2seq model, employing a framework that first reorders the NL and then decodes the MR. Explicit use of alignment information has also been explored: Herzig and Berant (2021) use alignments and predict a span tree over the NL. Sun et al. (2022) recently proposed an approach to data augmentation via sub-tree substitutions. In text-to-SQL, attention-based models that try to capture alignments have been proposed (Lei et al., 2020; Liu et al., 2021b), as well as attempts that try to leverage them directly (Sun et al., 2022).

Our two-step approach resembles statistical machine translation, which decomposes the translation task into lexical translation and reordering (Chang et al., 2022). Machine translation techniques have previously been applied to semantic parsing. The first attempt was by Wong and Mooney (2006), who argued that a parsing model can be viewed as a syntax-based translation model and used a statistical word alignment algorithm. Later a machine translation approach was used on the GEO dataset, obtaining what was at the time state-of-the-art results (Andreas et al., 2013). More recently, Agarwal et al. (2020) employed machine translation to aid semantic parsing.

3 Preliminaries: Word Alignments

This section briefly explains word alignments, showing the difference between monotonic and non-monotonic alignments, and illustrates the notion of monotonic translations.

Assume that we have a pair of sequences $\mathbf{x} = x_1, \ldots, x_n$ and $\mathbf{y} = y_1, \ldots, y_m$, where n and m are the respective sequence lengths. A bi-sequence is defined as the tuple $(\mathbf{x}, \mathbf{y})$. In our application, $\mathbf{x}$ is a NL sentence, and $\mathbf{y}$ is its corresponding MR. For example:

$\mathbf{x} =$ which city has the highest population density?
$\mathbf{y} =$ `answer(largest(density(city(all))))`

A word alignment is a set of bi-symbols $\mathcal{A}$, where each bi-symbol defines an alignment from a token in the NL to a token in the MR. For instance, the bi-symbol (x_i, y_j) aligns token x_i to token y_j. In our example, the tokens "which" and "answer" could be paired by a bi-symbol (which, answer).

If a token x_i does not align to anything in $\mathbf{y}$, an ε is introduced in $\mathbf{y}$: the resulting bi-symbol (x_i, ε) corresponds to a deletion. In our example, the token "has" in the NL can be deleted with a bi-symbol (has, ε). Similarly, if a token y_j is not aligned to a token in $\mathbf{x}$, an ε is introduced in $\mathbf{x}$: (ε, y_j) is an insertion. In our example, the token "all" in the MR is inserted with bi-symbol (ε, all).

The bi-symbols in $\mathcal{A}$ are all one-to-one. Hence, to map a single token to a phrase, i.e., to multiple tokens, it is necessary to choose a head token in the phrase, while the remaining tokens require insertion or deletion. In our example, the token "density" in the MR corresponds to "population density" in the NL, and, if "density" is chosen as the head token in the NL, "population" needs a deletion: the alignment will be given by the bi-symbols (population, ε) and (density, density).[1] Following this strategy, this notation can account for one-to-many and many-to-one alignments with deletion and insertion operations.

Figure 2a shows a possible bi-sequence word alignment for the aforementioned example. Each bi-symbol is conveniently represented by a horizontal line connecting the tokens it aligns.

Alignments can be monotonic or non-monotonic. An alignment is monotonic if it does not involve any *crossing*, i.e., a mapping that does not require reordering tokens. In our example, the alignment is non-monotonic because the bi-symbol (city,city) crosses over others. By permuting the MR, we can obtain a monotonic translation of the NL: Figure 2b shows such permutation. The next section illustrates how TPOL can leverage these translations.

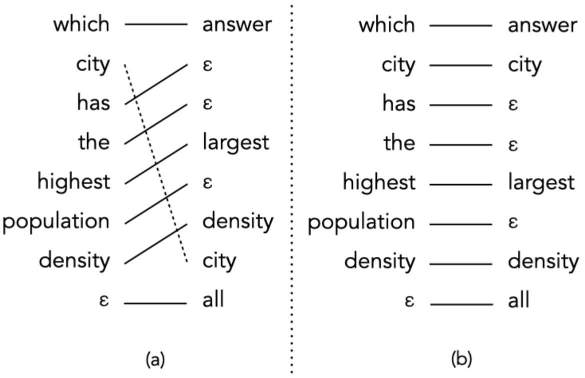

Figure 2: (a) A possible alignment for an NL-MR pair. (b) The corresponding monotonic translation. For simplicity, we removed the brackets and question mark.

4 Translate First Reorder Later

We propose TPOL, a two-step parsing approach with a modular framework made up of two components: a *Monotonic Translator* and a *Reorderer*. Figure 3 shows how our semantic parser takes an input sentence $\mathbf{x}$ and predicts the corresponding MR $\mathbf{y}$. In the first step, $\mathbf{x}$ is fed to the Translator, which outputs a monotonic translation $\mathbf{z}$. In other words, $\mathbf{z}$ is the target MR that has been permuted so that it aligns monotonically to the input NL. Then, in a second step, $\mathbf{z}$ is fed to the Reorderer, which is trained to place the MR tokens back into the correct order to produce the final prediction $\mathbf{y}$.

The main idea behind TPOL is decomposing the task into lexical translation and reordering, to learn more reliable translation patterns. We purport that modeling monotonic alignments eases the learning of novel pattern combinations of seen structures, improving compositional generalization.

An alternative approach would be to permute the NL inputs rather than the MRs monotonically. We do not follow this direction due to the observation that in semantic parsing, multiple NLs can map to the same MR. In other words, the NL domain is larger than that of the MRs, and thus we believe that learning to reorder the MRs is more feasible.

4.1 Monotonic Translator

The Monotonic Translator is responsible for making an initial prediction of the MR sequence, which will contain the correct tokens in monotonic order. To create the training bi-sequences, we use alignment information and permute the gold MR

[1] Locatelli and Quattoni (2022) showed that annotators are consistent in the way they pick head-tokens, and reported high inter-annotator agreement scores on GEOALIGNED.

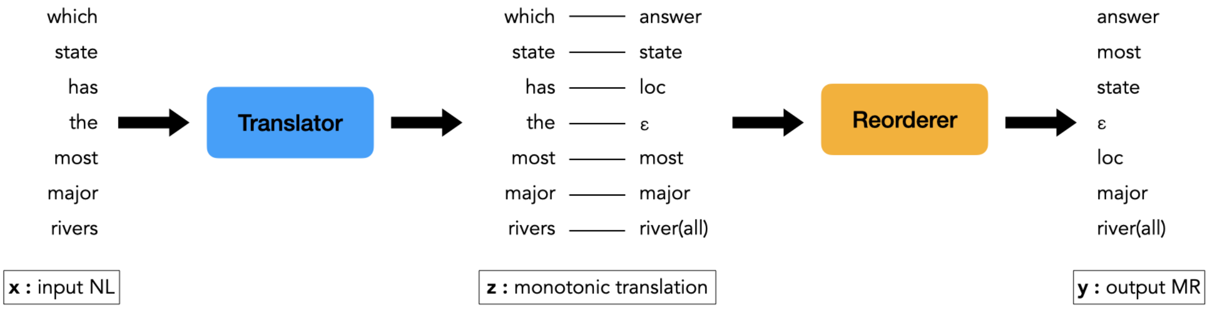

Figure 3: The TPoL parsing approach. An input sentence $\mathbf{x}$ is fed to the *Monotonic Translator* that predicts an intermediate monotonic MR $\mathbf{z}$. This is in turn fed to the *Reorderer*, which outputs the final prediction $\mathbf{y}$.

sequences to obtain a monotonic mapping with the NL. As a concrete example, consider the non-monotonic alignment in Figure 2a, and its monotonic translation in Figure 2b.

The translation task can be formulated in various ways. In our implementation, we work with two alternative approaches: a seq2seq Translator, and a tagger Translator. In the seq2seq formulation, $\mathbf{x}$ is fed into an encoder network, which produces a hidden vector. The hidden vector is fed to a decoder network which produces the output $\mathbf{z}$, i.e., the monotonically aligned MR. This can be implemented, for example, with a BART model (Lewis et al., 2020), which uses a bidirectional encoder and a left-to-right decoder. In our experiments, we use the multilingual version of BART (Liu et al., 2020). In the tagging formulation, the Translator assigns an MR token to each token in $\mathbf{x}$, obtaining the monotonic translation $\mathbf{z}$ by explicitly aligning in a token-by-token fashion. We implement this with a BERT model (Devlin et al., 2019) and we use its classification head as the tagger.

A crucial difference between the seq2seq and the tagger Translator is that the latter needs $\mathbf{x}$ and $\mathbf{z}$ to be the same length. The seq2seq Translator can learn to perform deletion operations from the raw NL, without needing epsilons in the input to perform insertions. By contrast, the tagger Translator needs insertions to be performed on $\mathbf{x}$ before predicting $\mathbf{z}$. In general, NL sequences are significantly longer than the MR sequences, i.e., most epsilons are in the MR sequence. In other words, deletions are more frequent than insertions.

However, for some datasets, some alignments contain epsilons in the NL sequence: at prediction time, we will not know where insertions might occur, and thus we need a way to predict them. For this purpose, for every token followed by an epsilon in the train split, we add an epsilon after it at test time. We saw that this strategy was sufficient in our experiments. Alternatively, this step could be done by a trained model or with a rule-based system similar to Ribeiro et al. (2018).

4.2 Reorderer

The Reorderer module is responsible for taking the monotonic predictions of the Translator and putting them back into the correct order to obtain the final prediction. This model is trained from pairs of MR sequences $(\mathbf{z}, \mathbf{y})$ where the input $\mathbf{z}$ is a monotonically permuted MR and the output $\mathbf{y}$ is the target MR in its correct order. These training pairs can be generated from the alignment annotations.

Similarly to the Translator module, the Reorderer can be implemented both as a seq2seq model and as a tagger. We use mBART in the seq2seq formulation and BERT as a tagger in our experiments. Note that we do not enforce the output to be a permutation of the input.

5 Experiments

5.1 Datasets

We test TPoL on two semantic parsing datasets, training with gold and automatically generated alignments in multiple languages, on standard IID partitions and the more challenging compositional ones.

5.1.1 GeoAligned

GeoAligned (Locatelli and Quattoni, 2022) augments the popular Geo semantic parsing benchmark (Zelle and Mooney, 1996) with token alignment annotations. The dataset contains questions about US geography and corresponding meaning representation using the FunQL formalism (Kate et al., 2005). In total, there are 880 examples, all annotated with token alignments. We evaluate on three partitions: question (?), query (Q) and length

(LEN). The question partition is a standard IID split where test and train are sampled from the same distribution. The query partition, introduced by Finegan-Dollak et al. (2018), is designed to be compositional by ensuring that the templates of the MRs in the test set are never seen during training. The length partition, introduced by Herzig and Berant (2021), assigns the longest sequences to the test.

The dataset comes in English, Italian and German: in this way we can test our approach across different languages. In our experiments, we do not anonymize constants: in other words, we keep the original NL and MR sequences which include names of cities, states, etc. We follow Wang et al. (2021) in removing brackets.

5.1.2 SCANSP

SCANSP (Herzig and Berant, 2021) is a set of navigational commands presented in natural language paired with action sequences. It is based on the SCAN dataset by Lake and Baroni (2018), which does not contain program MRs. Herzig and Berant (2021) translated the sequences into programs to obtain a semantic parsing version of the dataset. Besides the IID split, we test on the compositional partitions based on the "right" (RX) and "around right" (ARX) primitives from Loula et al. (2018). SCANSP has 20,910 commands distributed roughly as 12,000 train, 3,000 validation, and 4,000 test examples.

The SCANSP dataset does not come with alignments. Therefore we employ the IBM models (Brown et al., 1993) to generate them automatically using the GIZA++ toolkit (Och and Ney, 2003). We also do this for GEO to compare the performance of TPOL when trained with gold and automatic alignment annotations.

5.2 Models for comparisons

We compare with competitive baselines and state-of-the-art models that do not leverage alignments and competing models that do.

- **LSTM**: a standard seq2seq model with a bidirectional LSTM encoder and an LSTM decoder with attention (Bahdanau et al., 2015). We use pre-trained GloVe embeddings for the three languages: English (Pennington et al., 2014), Italian and German (Ferreira et al., 2016).
- **mBART** (Liu et al., 2020): a multilingual version of BART (Lewis et al., 2020), a pre-trained Transformer-based seq2seq model that has been successfully applied to parsing (Bevilacqua et al., 2021).
- **mT5** (Xue et al., 2021): a multilingual version of T5 (Raffel et al., 2020), pre-trained on the mC4 dataset (Xue et al., 2021).
- **SPANBASED** (Herzig and Berant, 2021): a semantic parser that predicts a span tree over an input utterance trained with gold alignment trees. The authors provided annotations for the English version of GEO and SCANSP. For the other languages of GEO we train without gold alignments. We use their model without the lexicon, as that would be unfair with respect to the other models.
- **LEAR** (Liu et al., 2021a): a model that learns to recombine structures recursively by predicting a latent syntax tree and assigning semantic operations to non-terminal nodes. LEAR explicitly uses alignments using a phrase table.
- **REMOTO** (Wang et al., 2021): a model that first reorders the tokens in the NL and then predicts the MR. REMOTO is not trained with gold alignments.[2]

5.3 Evaluation metric

We follow the standard practice of using exact-match accuracy for evaluation: the predicted MR is correct only if it is the same as the gold.

5.4 Main Results

We report the results of our experiments in Table 1. For TPOL, the choice of the modules' architecture is validated on the development set, and we report a performance breakdown in Section 7.

We first consider the results of TPOL trained with gold alignments. On the GEO dataset, the LSTM and mT5 achieve the lowest performance in all the partitions. Looking at the question partition (?), the models show similar performance to mBART and SPANBASED, which is not surprising as the test split does not require compositional generalization. On the query partition (Q), designed to test for compositional generalization, TPOL shows significant improvements over all the other models across all languages. In English it obtains 87.3% outperforming mBART (69.4%), SPANBASED (74.6%) and LEAR (84.1%). In Italian and German, it obtains 81.6% and 69.4% respectively, while mBART 67.4% and 56.3%. On

[2] For LEAR and REMOTO, we report results directly from the respective papers, noting that in their setting constants, such as names of states, cities, and so on, are anonymized.

Model	**GEO**											
	EN			**IT**			**DE**			**SCANSP**		
	?	**Q**	**LEN**	**?**	**Q**	**LEN**	**?**	**Q**	**LEN**	**IID**	**RX**	**ARX**
LSTM	52.9	24.9	5.0	46.4	18.1	4.3	42.9	17.6	3.2	100	24.4	1.1
mT5	80.0	60.0	19.3	73.2	44.9	20.4	68.2	47.8	18.6	100	41.2	99.8
mBART	87.5	69.4	27.5	**86.6**	76.4	23.3	**75.5**	56.3	18.2	100	99.4	100
LEAR	-	84.1	-	-	-	-	-	-	-	-	-	-
SPANBASED	**87.7**	74.6	**55.0**	-	-	-	-	-	-	100	100	100
- gold	66.4	51.8	24.6	49.6	37.3	10.4	40.4	21.4	5.0	100	100	100
REMOTO	75.2	43.2	23.2	-	-	-	55.6	22.3	16.6	100	-	-
TPOL	87.3	**87.8**	41.9	85.9	**81.6**	**31.3**	73.3	**69.4**	**22.9**	-	-	-
- gold	85.8	79.0	35.6	83.6	75.1	20.2	73.8	60.7	17.5	100	99.4	100

Table 1: Exact-match accuracy of all models on GEOALIGNED and SCANSP datasets. ? stands for question, Q for query and LEN for the length partition. RX stands for right and ARX for around right partitions. LEAR and REMOTO both anonymize constants in GEO, and the results are taken directly from the respective papers.

the length partition (LEN), TPOL does better than all the baselines across all languages, except for SPANBASED, which fares better on LEN(English). This is only the case, however, when gold alignments are provided.

Looking at the results obtained without gold alignments, TPOL shows considerable improvements over REMOTO and SPANBASED. In particular, it improves on the English query partition obtaining 79% against 43.2% and 51.8%, respectively. Furthermore, the accuracy does not drop significantly compared to TPOL trained with gold alignments. We tested using automatic alignments from IBM models 3, 4, and 5 and picked the best out of the three. In general, all lead TPOL to achieve similar performance.

Finally, looking at SCANSP, as expected, the models designed for compositional generalization achieve perfect performance on the dataset. What is surprising is that also mBART can do so, contrary to other deep models. With some internal testing, we have seen that this is not the case for English BART, as opposed to the multilingual version. We hypothesize that model size and pre-training might be a factor of success for mBART.

6 Error Analysis

Table 3 shows a breakdown of performance of TPOL on the English version of GEOALIGNED. The results indicate the exact-match accuracy achieved by the two modules: we can check whether the model struggles more with the translation or reordering step. To analyze the Translator's performance, we regard the monotonically aligned MRs as the gold truth. For the Reorderer, we provide it with the monotonically aligned MRs in input. In other words, the evaluation of the Reorderer assumes that the Translator makes a correct prediction.

The performance of the two modules is fairly similar, and, by comparing these results with Table 1, we see that the accuracy of each component is not much higher than the overall accuracy, suggesting that neither component is hampering performance more than the other. The only exception seems to be in the length partition, where the Reorderer does considerably better than the Translator.

Table 2 shows the breakdown of the performance over monotonically and non-monotonically aligned MRs. We can observe that, compared to SPANBASED, TPOL generally presents smaller drops in performance over the non-monotonic sequences. For instance, in the question partition, SPANBASED drops from 94.7% over the monotonic examples to 73.7% over the non-monotonic, while TPOL drops from 89.9% to 81.4%. When we look at the query partition, we see that for both of these models, the drop is much higher than the one on the question partition: SPANBASED goes from 89.6% to 39.8%, while TPOL drops from 93.7% to 69.9%. In other words, both models struggle more with non-monotonic examples when compositional generalization is required. TPOL, however, still performs significantly better.

Model	**GeoEN**					
	?		**Q**		**LEN**	
	MN	**NMN**	**MN**	**NMN**	**MN**	**NMN**
mBART	90.4	81.0	67.1	76.5	29.2	25.1
SpanBased	94.7	73.7	89.6	39.8	68.1	35.9
TPol	89.9	81.4	93.7	69.9	55.5	23.2

Table 2: Performance breakdown of TPol over monotonic (MN) and non-monotonic (NMN) sequences in GeoAligned English. In Appendix B we report the number of MN and NMN examples.

Module	**GeoEN**		
	?	**Q**	**LEN**
Translator	86.1	86.2	42.5
Reorderer	87.6	87.3	57.1

Table 3: Performance breakdown of TPol modules.

Surprisingly, mBART shows the opposite trend on the query partition, with the non-monotonic accuracy being higher than the monotonic one. By contrast, most of TPol's improved performance comes from better modeling of the monotonic sequences (67.1% → 93.7%). TPol's results suggest that regular patterns in the non-monotonic sequences can be learned. Its generalization problems can be attributed to the difficulty of learning the more challenging non-regular patterns in a small dataset. On the other hand, mBART appears to have the capacity to model these challenging reorderings better. Still, this comes at the cost of failing on the regular monotonic ones, leading to a lower performance overall.

Interestingly, the SpanBased approach shows a similar improvement on monotonic sequences compared to mBART (67.1% → 89.6%). This suggests that exploiting lexico-logical alignments allows models to capture the simpler patterns that mBART fails to learn. Most of TPol's gains over SpanBased come from better modeling the non-monotonic examples in the query partition (39.8% → 69.9%). This shows that the two-step approach offers the best of both worlds: it can capture the simple monotonic patterns while maintaining reasonable performance over the more complex alignments on which SpanBased fails.

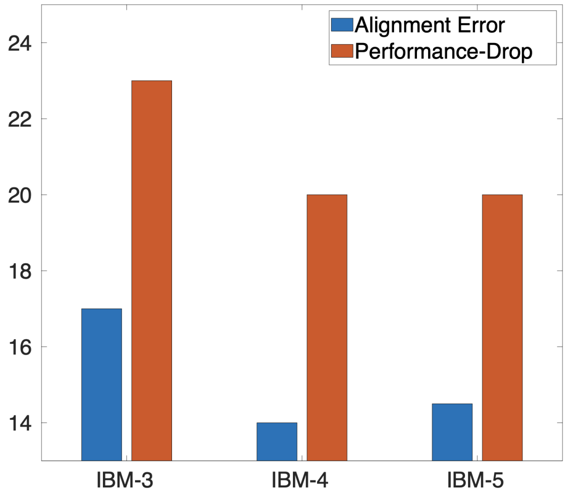

Figure 4: Average error of IBM alignment models over all partitions and languages on GeoAligned. We also plot the average drop in performance for each TPol model trained with IBM alignments with respect to the corresponding one trained with gold alignments.

Figure 4 shows the average drop in performance of the different models trained with automatic IBM alignments compared to the same model trained with gold alignments. We also report the corresponding alignment error, calculated as the percentage of bi-symbols that differ from the gold annotations from GeoAligned. We observe that, in general, higher alignment error is associated with a higher drop in performance. This validates the importance of the alignment information and points to improving the unsupervised alignment algorithm as a natural line of future work. We believe that one possible reason for the drop in performance when training with IBM alignments might be because the GEO dataset is of relatively small size, and the IBM models might have difficulty learning good alignments. That would explain why in contrast to GEO, the performance over SCAN is not affected by automatic alignments since SCAN is a much larger dataset.

Model	GEO								
	EN			IT			DE		
	?	Q	LEN	?	Q	LEN	?	Q	LEN
BERT2BERT	74.9	84.6	**41.9**	74.0	74.2	**31.3**	62.5	67.5	**22.9**
BERT2mBART	82.1	**87.8**	36.4	76.9	**81.6**	27.5	63.2	68.9	20.0
BERT2mBART SILVER	82.5	87.0	34.6	76.6	81.1	27.0	65.4	**69.4**	19.9
mBART2BERT	74.5	70.7	24.5	77.7	76.1	24.2	65.7	56.1	17.5
mBART2mBART	**87.3**	72.2	25.2	85.7	76.3	23.1	**73.3**	59.2	16.3
mBART2mBART SILVER	86.4	71.5	24.9	**85.9**	75.8	20.0	73.2	59.5	17.0

Table 4: Performance breakdown of TPOL for different module architectures on GEOALIGNED.

7 Architecture study

We emphasize here that our approach is an abstract, high-level methodology and does not place any constraint on the underlying architectures of the two components. We believe that different architectures, particularly specialized ones for each module, could be beneficial for parsing performance. We encourage further work to be carried out in this regard. To this purpose, we present some architectural studies using BERT (Devlin et al., 2019) and mBART (Liu et al., 2020) as components. We employ them as both Translator and Reorderer, examining all possible combinations. As explained in Section 4.1, mBART is used as a standard seq2seq model, and BERT is employed with a classification head to function as a tagger for every input token.

Additionally, when mBART is used as a Reorderer, we introduce a silver training setting. In the normal setting, the Reorderer is trained by taking the gold alignment annotations and outputting the meaning representation. In the silver setting, we use the predictions of the Translator model as training input. By doing so, the Reorderer trains on inputs that mimic more closely what it will actually receive at test time: this is done straightforwardly for a seq2seq model like mBART, while for our BERT tagger, every token in input needs to be aligned with a token in output, and when the input is corrupt it is not possible to achieve the same training technique.

In Table 4, we present the results for our different architectural components. To distinguish among the different model combinations, we use a [Translator]2[Reorderer] naming convention, meaning that mBART2BERT uses mBART as the Translator and BERT as the Reorderer. We observe that our two-step approach seems to be robust overall.

We can discern trends in different architecture combinations, which can be helpful when choosing an architecture for a specific task. One important observation is that the architectures that use BERT as a Translator are consistently better than the ones using mBART over the compositional partitions. We hypothesize that the BERT Translator can achieve higher compositional generalization because it can better leverage alignment information to predict unseen combinations of observed training patterns. We believe this is because a tagger's predictions can be more naturally broken into parts that can be recombined. In contrast, encoder-decoder architectures fare better on the IID partition but struggle to generalize to unseen patterns. One possible reason is that these models have a harder time inducing local patterns that can be recombined since they encode and decode complete structures all at once.

8 Conclusion

Seq2seq models have become increasingly popular in semantic parsing. However, they are limited in their abilities to generalize to unobserved structures. Here, we proposed TPOL: a two-step parsing approach that leverages alignment annotations with a modular framework composed of a Translator and a Reorderer.

We showed that TPOL improves compositional generalization over conventional seq2seq models and over competing models that also leverage alignment information. Our results also showed that our approach is robust when trained with automatically generated alignments, demonstrating competitive results on two semantic parsing datasets.

We have experimented with two possibilities for the Translator and Reorderer, but we believe that different architectural components could further improve performance. The divide-and-conquer strategy of breaking the problem into two simpler sub-tasks is designed to enable further component specialization.

9 Limitations

Regarding the limitations of our approach, our experiments used the standard FunQL meaning representation. Transitioning to a different meaning representation might need some adaptation of the framework. In particular, the alignments between NL and MRs for other meaning representations might require more insertion and deletion operations. We might also expect that other MRs might require more reordering.

A second limitation of our work is training with gold alignments. We partially address this by training TPOL with automatic alignments obtained with the IBM models. Still, we believe there is room for more work to be done so that this approach can be more easily scaled to datasets that do not have alignment annotations.

Despite TPOL's partial improvements on the length test splits, this type of partition remains challenging for all models. Here, models are required to generate predictions of greater length than what they have seen during training. This requires complex compositional productivity skills, i.e., recombining known constituents into larger structures. Further work is needed to address the limitation of the current state-of-the-art on compositional productivity benchmarks.

Acknowledgements

We would like to thank the EACL area chair and the anonymous reviewers for their feedback, as well as the other members of the INTERACT group at the Universitat Politècnica de Catalunya for fruitful discussions on an earlier draft of this work. This work is supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement No.853459). This paper reflects the authors' view only, and the funding agencies are not responsible for any use that may be made of the information it contains. The authors gratefully acknowledge the computer resources at Artemisa, funded by the European Union ERDF and Comunitat Valenciana, as well as the technical support provided by the Instituto de Fisica Corpus- cular, IFIC (CSIC-UV).

References

Oshin Agarwal, Mihir Kale, Heming Ge, Siamak Shakeri, and Rami Al-Rfou. 2020. Machine translation aided bilingual data-to-text generation and semantic parsing. In *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*, pages 125–130, Dublin, Ireland (Virtual). 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.

Jacob Andreas, Andreas Vlachos, and Stephen Clark. 2013. Semantic parsing as machine translation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 47–52, Sofia, Bulgaria. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Michele Bevilacqua, Rexhina Blloshmi, and Roberto Navigli. 2021. One SPRING to Rule Them Both: Symmetric AMR Semantic Parsing and Generation without a Complex Pipeline. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(14):12564–12573.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

Chih-Chiang Chang, Shun-Po Chuang, and Hung-yi Lee. 2022. Anticipation-free training for simultaneous machine translation. In *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*, pages 43–61, Dublin, Ireland (in-person and online). Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. 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.

Daniel C. Ferreira, André F. T. Martins, and Mariana S. C. Almeida. 2016. Jointly learning to embed and predict with multiple languages. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2019–2028, 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.

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

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.

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.

Rohit J. Kate, Yuk Wah Wong, and Raymond J. Mooney. 2005. Learning to transform natural to formal languages. In *Proceedings of the 20th National Conference on Artificial Intelligence - Volume 3*, AAAI'05, page 1062–1068. AAAI Press.

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 *International Conference on Learning Representations*.

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.

Brenden 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 2873–2882. PMLR.

Brenden M Lake. 2019. Compositional generalization through meta sequence-to-sequence learning. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Wenqiang Lei, Weixin Wang, Zhixin Ma, Tian Gan, Wei Lu, Min-Yen Kan, and Tat-Seng Chua. 2020. Re-examining the role of schema linking in text-to-SQL. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6943–6954, Online. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Chenyao Liu, Shengnan An, Zeqi Lin, Qian Liu, Bei Chen, Jian-Guang Lou, Lijie Wen, Nanning Zheng, and Dongmei Zhang. 2021a. Learning algebraic recombination for compositional generalization. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1129–1144, Online. Association for Computational Linguistics.

Qian Liu, Dejian Yang, Jiahui Zhang, Jiaqi Guo, Bin Zhou, and Jian-Guang Lou. 2021b. Awakening latent grounding from pretrained language models for semantic parsing. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1174–1189, Online. Association for Computational Linguistics.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Davide Locatelli and Ariadna Quattoni. 2022. Measuring alignment bias in neural seq2seq semantic parsers. In *Proceedings of the 11th Joint Conference on Lexical and Computational Semantics*, pages 200–207, Seattle, Washington. Association for Computational Linguistics.

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.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

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.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Linlu Qiu, Peter Shaw, Panupong Pasupat, Pawel Nowak, Tal Linzen, Fei Sha, and Kristina Toutanova. 2022a. Improving compositional generalization with latent structure and data augmentation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4341–4362, Seattle, United States. Association for Computational Linguistics.

Linlu Qiu, Peter Shaw, Panupong Pasupat, Tianze Shi, Jonathan Herzig, Emily Pitler, Fei Sha, and Kristina Toutanova. 2022b. Evaluating the impact of model scale for compositional generalization in semantic parsing.

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.

Joana Ribeiro, Shashi Narayan, Shay B. Cohen, and Xavier Carreras. 2018. Local string transduction as sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1360–1371, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Peter Shaw, Ming-Wei Chang, Panupong Pasupat, and Kristina Toutanova. 2021. Compositional generalization and natural language variation: Can a semantic parsing approach handle both? 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 922–938, Online. Association for Computational Linguistics.

Tianze Shi, Chen Zhao, Jordan Boyd-Graber, Hal Daumé III, and Lillian Lee. 2020. On the potential of lexico-logical alignments for semantic parsing to SQL queries. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1849–1864, Online. Association for Computational Linguistics.

Runxin Sun, Shizhu He, Chong Zhu, Yaohan He, Jinlong Li, Jun Zhao, and Kang Liu. 2022. Leveraging explicit lexico-logical alignments in text-to-SQL parsing. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 283–289, Dublin, Ireland. Association for Computational Linguistics.

Bailin Wang, Mirella Lapata, and Ivan Titov. 2021. Structured reordering for modeling latent alignments in sequence transduction. In *Advances in Neural Information Processing Systems*, volume 34, pages 13378–13391. Curran Associates, Inc.

Pia Weißenhorn, Lucia Donatelli, and Alexander Koller. 2022. Compositional generalization with a broad-coverage semantic parser. In *Proceedings of the 11th Joint Conference on Lexical and Computational Semantics*, pages 44–54, Seattle, Washington. Association for Computational Linguistics.

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.

Yuk Wah Wong and Raymond Mooney. 2006. Learning for semantic parsing with statistical machine translation. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 439–446, New York City, USA. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Jingfeng Yang, Le Zhang, and Diyi Yang. 2022. SUBS: Subtree substitution for compositional semantic parsing. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 169–174, Seattle, United States. Association for Computational Linguistics.

John M. Zelle and Raymond J. Mooney. 1996. Learning to parse database queries using inductive logic programming. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence - Volume 2*, AAAI'96, page 1050–1055. AAAI Press.

Sheng Zhang, Xutai Ma, Kevin Duh, and Benjamin Van Durme. 2019. AMR parsing as sequence-to-graph transduction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 80–94, Florence, Italy. Association for Computational Linguistics.

A Experimental details

For our experiments with TPOL we report the average of three runs for every result. We select the hyperparameters with grid search on the development set performance stopping when there is no more improvement. We choose the learning rate among $1e^{-4}$, $1e^{-5}$ and $1e^{-6}$ and the batch size between the bounds of 4 and 32. Usually the best performing models choose a learning rate of $1e^{-5}$ and batch size of 8. An experiment takes about 20 minutes on a single Nvidia V100 GPU. Our BERT (110M parameters) and mBART (680M parameters) implementations are taken from the transformers library (Wolf et al., 2020). We use for English *bert-base-uncased*, for Italian *dbmdz/bert-base-italian-uncased* and for German *dbmdz/bert-base-german-uncased*. For mBART we use *facebook/mbart-large-50*. For mT5 we use the *google/mt5-small* pre-trained checkpoint from the Transformers library.

B GEOALIGNED statistics

Table 5 provides statistics from the English version of GEOALIGNED (Locatelli and Quattoni, 2022). In particular, we report the number of examples that fall in the monotonic (MN) and non-monotonic (NMN) categories.

	GEO EN		
Category	**?**	**Q**	**LEN**
MN	194	154	162
NMN	86	51	118

Table 5: Number of examples that belong to the monotonic (MN) and non-monotonic (NMN) categories in GEOALIGNED English.