Exploring the Compositional Generalization in Context Dependent Text-to-SQL Parsing

Aiwei Liu*, **Wei Liu***, **Xuming Hu, Shu'ang Li, Fukun Ma,**
Yawen Yang, Lijie Wen†
Tsinghua University
{liuaw20,liu-w21,hxm19,lisa18, mafk19, yyw19}@mails.tsinghua.edu.cn
wenlj@tsinghua.edu.cn

Abstract

In the context-dependent Text-to-SQL task, the generated SQL statements are refined iteratively based on the user input utterance from each interaction. The input text from each interaction can be viewed as component modifications to the previous SQL statements, which could be further extracted as the modification patterns. Since these modification patterns could also be combined with other SQL statements, the models are supposed to have the compositional generalization to these novel combinations. This work is the first exploration of compositional generalization in context-dependent Text-to-SQL scenarios. To facilitate related studies, we constructed two challenging benchmarks named CoSQL-CG and SPARC-CG by recombining the modification patterns and existing SQL statements. The following experiments show that all current models struggle on our proposed benchmarks. Furthermore, we found that better aligning the previous SQL statements with the input utterance could give models better compositional generalization ability. Based on these observations, we propose a method named `p-align` to improve the compositional generalization of Text-to-SQL models. Further experiments validate the effectiveness of our method. Source code and data are available [1]

1 Introduction

Recently, the poor generalization of semantic parsing models to out-of-distribution samples is under increasing attention (Keysers et al., 2020; **?**). These examples are usually obtained by recombining existing structures. For example, in the SCAN dataset (Lake and Baroni, 2018a), models may fail to parse "jump twice and walk" even though "jump twice" and "walk" could be parsed successfully. The ability to generalize to novel combinations is also known as compositional generalization. Text-to-SQL (Yu et al., 2018) allows non-expert users to access the information from the database by converting the user input text into SQL statements executed in the database. As a typical semantic parsing task, the study of its compositional generalization is of great importance.

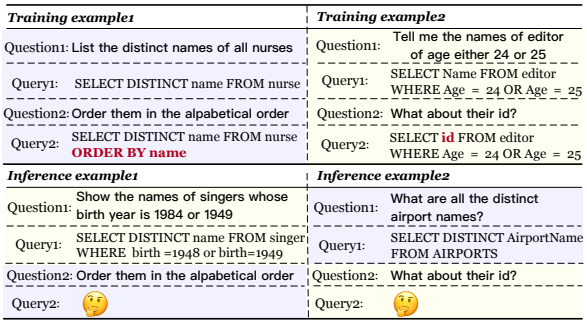


Figure 1: During the inference phase, the base queries and their modifications could be re-combined. Models with compositional generalization ability should successfully parse these novel combinations.

Existing works explore the compositional generalization of Text-to-SQL only in the scenario that precisely maps stand-alone utterances to SQL queries. Shaw et al. (2021) define the atom and compound for SQL statements and propose the TMCD split to repartition the dataset. Gan et al. (2022) annotate the alignment of sub-sentence and sub-SQL in the spider dataset (Yu et al., 2018) and then recombine these sub-SQLs and sub-sentences. In these settings, the SQL statements and user questions in the constructed test split tend to be much more complex. However, it is difficult for users to express complex queries in a stand-alone sentence. In real scenarios, users often start with a simple query and continuously combine additional query conditions with subsequent questions.

In this work, we focus on the study of compositional generalization in context-dependent Text-to-SQL tasks, which is more natural and applica-

[1] https://github.com/THU-BPM/CD-Text2SQL-CG
*Equally Contributed.
† Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2023, pages 688–700
July 9-14, 2023

ble. In the context-dependent Text-to-SQL task (Yu et al., 2019b), the generated SQL statements are refined based on the user input text during each interaction. The input text from each interaction can be viewed as component modifications to the previous SQL statement, which could be further extracted as the modification patterns. Since these modification patterns could also be combined with other SQL statements, the models are supposed to have the compositional generalization to these novel combinations. For example, in Figure 1, the modifications and the queries of the first turn in the training phrase could be re-combined in the inference phrase. Applicable models are supposed to successfully parse these novel combinations.

To better investigate compositional generalization in the context-dependent Text-to-SQL, we first construct compositional generalization benchmarks based on the existing datasets. First, we extract the modification patterns from the training dataset and then recombine them with the existing SQL statements in the development set. Note that in the compositional generalization setting, only the recombination results not existing in the training set are kept. To generate the corresponding utterances, we use a semi-automatic approach. The utterances are initially generated by a pre-trained model fine-tuned on the training data, and then reviewed and verified by human experts. As a result, we create two benchmarks, CoSQL-CG and SPARC-CG, specifically for the datasets CoSQL(Yu et al., 2019a) and SPARC(Yu et al., 2019b). Our experiments reveal that current state-of-the-art models perform poorly on these benchmarks, emphasizing the significance of enhancing compositional generalization capabilities.

We further explore how to improve the compositional generalization in context-dependent Text-to-SQL tasks. Inspired by the previous works to improve compositional generalization by fine-grained alignment of inputs and outputs (Zheng and Lapata, 2022; Akyürek and Andreas, 2021), we propose a method to better align the current text with the previous SQL statements. We follow the common practice of most competitive Text-to-SQL models which take the concatenation of all utterances as input. Specifically, our proposed `p-align` method extracts the embedding of the text from each interaction after the encoding process and then decodes them into the corresponding SQL statements separately. Further experiment results show that our `p-align` method could effectively improve the compositional generalization of current models, which also demonstrates that better alignment of text and SQL statements and the introduction of previous SQL statements are of great importance.

To summarize, the main contributions of our paper are as follows:

- To the best of our knowledge, we are the first to explore compositional generalization in context-dependent Text-to-SQL.
- We construct two benchmarks named CoSQL-CG and SPARC-CG to better facilitate the relevant research.
- We propose a simple and effective method named `p-align` to improve the compositional generalization ability of models.

2 Related Work

2.1 Context dependent Text-to-SQL

Most current research on Text-to-SQL is conducted under the context-independent setting, with many recent methods achieving excellent results on the Spider dataset (Yu et al., 2018), including graph-based methods such as LGESQL(Cao et al., 2021a), RAT-SQL (Wang et al., 2020) and ISESL-SQL (Liu et al., 2022a), as well as sequence-to-sequence-based methods like PICARD (Scholak et al., 2021). Recently, with the presentation of two datasets CoSQL(Yu et al., 2019a) and SPARC(Yu et al., 2019b), the Text-to-SQL parsing under the context-dependent setting has attracted much attention, which is more realistic and applicable. Subsequently, various methods have been proposed. Among them, SCORE(Yu et al., 2021) and STAR(Cai et al., 2022) aim to train better pre-trained models to improve the parsing ability of models. Also, many sequence-to-sequence methods based on T5 pre-trained model like PICARD (Scholak et al., 2021) and RASAT (Qi et al., 2022) have achieved great success. Meanwhile, more methods pay more attention to contextual information or conversation history during encoding, including IGSQL(Cai and Wan, 2020), HIE-SQL(Zheng et al., 2022), and IST-SQL(Wang et al., 2021). Meanwhile, other rewriting-based methods like DELTA(Chen et al., 2021) and CQR-SQL(Xiao et al., 2022) reformulate the current and the historical texts into an individual sentence. Different from the previous works, we mainly focus

on exploring compositional generalization under context-dependent text-to-SQL settings.

2.2 Compositional Generalization

Compositional Generalization is an important metric for evaluating the robustness of the model (Liu et al., 2022b) in the field of natural language processing. For semantic parsing tasks, the ability to generalize to structures generated by systematically combining known atomic components is of vital importance. Lake and Baroni (2018b) propose the SCAN dataset, which maps word sequences into navigation command sequences (e.g., jump twice → JUMP JUMP). Their training/evaluation split are constructed in a compositional generalization way. Keysers et al. (2020), introduce CFQ dataset and propose distribution-based compositionality assessment to measure compositional generalization. Hupkes et al. (2020) summerize five different compositionally generalization splits and combine them to generate PCFG SET. Many works focus on improving the compositional generalization of models. This is usually achieved by introducing more detailed lexicon or lexicon-style alignments (Zheng and Lapata, 2022; Akyürek and Andreas, 2021) or adopting a grammar-based decoder (Herzig and Berant, 2021; Qiu et al., 2022b; Guo et al., 2020). Another line of work attempts to synthesize examples utilizing grammar and generative models for data augmentation (Qiu et al., 2022a; Andreas, 2020; Jia and Liang, 2016).

Recently, the compositional generalization of Text-to-SQL parsing has gained more and more interest. Shaw et al. (2021) define the atom and compound for SQL statements and propose the TMCD split to repartition the dataset. Gan et al. (2022) annotate the alignment of sub-sentence and sub-SQL in the spider dataset (Yu et al., 2018) and then recombine these sub-SQLs and sub-sentences. The above works only focus on the Text-to-SQL parsing in the context-independent setting, which precisely maps stand-alone utterances to SQL queries. However, it is difficult for users to express complex queries in a stand-alone sentence. In this work, we first explore the compositional generalization for context-dependent Text-to-SQL Parsing.

3 Compositional Generalization in Context-dependent Text-to-SQL

To facilitate the understanding of the following sections, we provide a more detailed explanation of compositional generalization in context-dependent Text-to-SQL parsing in this section.

The template split is a typical compositional generalization setting, where the structure templates in the training and test set are completely separated. Our compositional generalization scenario can be viewed as an extension of the template split, where the combination of basic SQL templates and modification templates in the training and test set are separated. Note that basic SQL and modification templates in the test set all appear in the training set individually. For instance, in figure 1, in the inference phrase, although all the templates are seen during training, their combinations are novel.

From another point of view, our compositional generalization scenario could also be viewed as a special case of TMCD split (Shaw et al., 2021), where the SQL templates and modification templates could be seen as atoms and their combination results are the compounds. Note the utterance to the SQL templates (first atom) are provided during training, which could be further utilized to improve the compositional generalization (Section 5).

4 Benchmark construction

Since there are few data satisfying the compositional generalization setting in the origin SPARC and COSQL development set. We first construct new benchmarks to facilitate the related research.

As illustrated in Figure 2, the benchmark construction process can be divided into four steps. The first step is to filter out context-independent examples; next, modification patterns are extracted from the remaining examples; after that, these modification patterns are combined with other SQL statements, and finally, corresponding utterances are generated.

4.1 Filter out context-independent examples

It is observed that a significant number of examples in the SPARC or COSQL datasets are context-independent, meaning that no context information is needed to generate the current queries. In this work, we propose a schema-linking-based method to filter out these context-independent examples.

Schema linking is a common technique in Text-to-SQL which links the exact or partial occurrences of the column/table names in the question such as the `ARILINES` and `Abbreviation` in Figure 2(a). Our main motivation is that if current data is context-dependent, there are some column/table

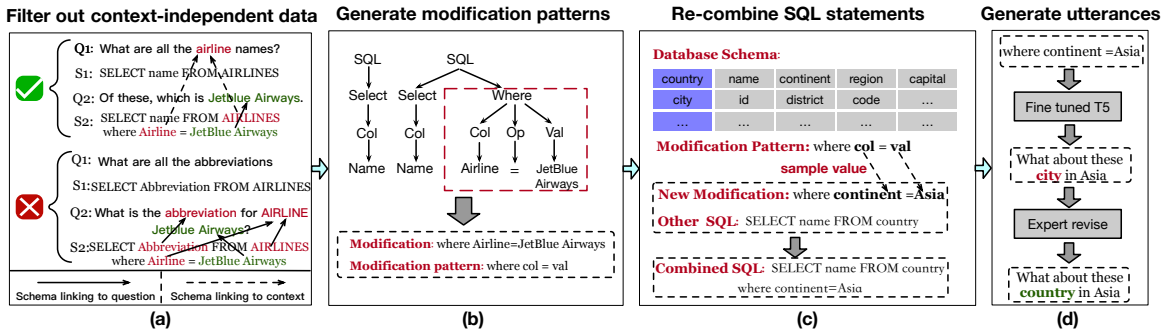


Figure 2: The benchmark construction process can be divided into four steps. The first step is to filter the context-independent data; then the next step is to generate modification patterns from the remained examples; after that, the modification patterns are recombined with other queries and the last step is to generate the corresponding utterances.

names not linked to the current question but linked to history questions (context), such as the first example in Figure 2(a). Specifically, the schemas in the target query are represented as S. We use the n-gram matching method to find occurrences S in the current question, where the matched schemas could be represented as S_c. Similarly, the matched schemas in the history questions are represented as S_p. The current example is context-dependent only if $S_p - S_c \neq \emptyset$. Finally, we keep 4270 and 2347 context-dependent examples in SPARC and COSQL training set respectively.

4.2 Generate Modification Pattern

After filtering out context-independent data, the next step is to generate modification patterns from the remaining context-dependent examples.

As shown in Figure 2(b), we first parse current and previous SQL statements into abstract syntax trees and then compare the tree structures to get the modified components. Specifically, a top-down traversal algorithm is adopted to find the different nodes. The nodes along with their children constitute the modified component. Then the generated modification component is anonymized to generate the modification template. Finally, we generate 409 and 191 modification templates for SPARC and COSQL respectively.

4.3 Re-combine SQL statements

With the generated modification patterns, the next step is to re-combine these patterns with other SQL statements to generate new SQL statements.

First, modification patterns are filled with new table/column names sampled from target database schemas to generate new modifications. Then the modifications are directly combined with the other SQL statements. Note that in the previous modification pattern generation process, the relationship of the schema is kept (e.g. primary key and foreign key relationships) and the table/column name sampling results must conform to the above relationship constraints. As mentioned in Section 3, the combination process requires that the base SQL templates and modification templates are all shown in the training set but their combinations are novel. Finally, we generate 5958 and 2594 combination results in SparC and CoSQL respectively.

4.4 Utterance generation

The final step of our benchmark construction is to generate the context-dependent utterance for the generated SQL statements. Since pre-trained language models have shown great ability in text generation, we first utilize a fine-tuned T5 model (Raffel et al., 2020) to generate the context-dependent utterance. More specifically, the input to the T5 model is the concatenation of the modification, previous SQL statement, and previous utterance.

For the utterance generated by the T5 model may be noisy, we further invite human experts to filter and revise the generated data. The first task of human experts is to remove SQL statements that don't fit realistic scenarios. For example, the statement `SELECT Count(loser_entry) FROM matches ORDER BY matches.winner_age` is invalid because the function `Count()` and the clause `ORDER BY` usually do not appear together. The second task of the human experts is to revise the utterances generated by the T5 model as shown in Figure 2(d). To ensure annotation consistency, we introduce two experts to double-check the annotated results. Finally, after the filtering and revising process, we get 372 and 267 questions for SPARC

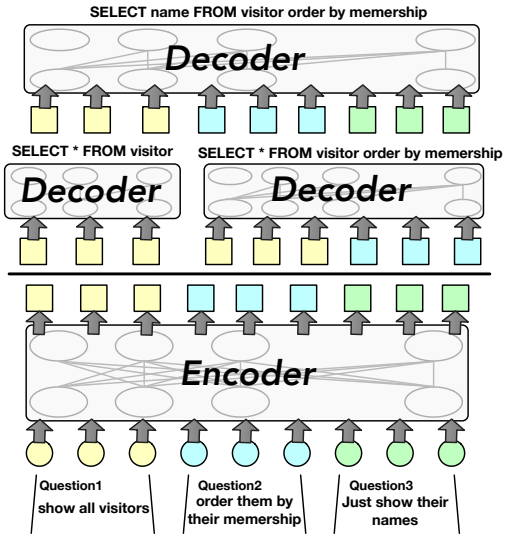


Figure 3: The whole process of our p-align method. The input to the encoding process is the concatenation of the utterance from all interactions. In the decoding process, the utterance embeddings of each interaction are extracted to decode the corresponding SQL.

and CoSQL datasets respectively, which further construct our SPARC-CG and CoSQL-CG benchmarks. More detailed statistics of the benchmarks will be described in the experiment section.

5 Methods

After constructing the SPARC-CG and CoSQL-CG, we further explore how to improve the compositional generalization in context-dependent Text-to-SQL parsing. According to the previous works (Zheng and Lapata, 2022; Akyürek and Andreas, 2021), the key to improving the compositional generalization is to construct better component alignment between inputs and outputs. In the context-dependent Text-to-SQL settings, the utterance-query pair of previous interactions could be utilized to align input utterances and output queries. Based on this motivation, we propose p-align to improve the compositional generalization of existing Text-to-SQL models. Note that our method follows the common practice of most competitive Text-to-SQL models which take the concatenation of all utterances as input.

Specifically, given the input utterances $X = [X_1, X_2, ..., X_n]$ at the n-th interaction, where $X_n = [x_1, ...x_j]$ is an utterance with j words, the encoder aims to generate embeddings for each word such that $\mathbf{X} = \mathbf{H}(X)$. In the origin decoding process, the result query y could be represented as an action sequence $[a_1, ...a_t]$ and the whole decoding process could be represented as the product of probabilities for each generation step as follows:

$$\prod_{t=1}^{T} p\left(a_t \mid \{a_1, \ldots, a_{t-1}\}, \mathbf{X}\right). \quad (1)$$

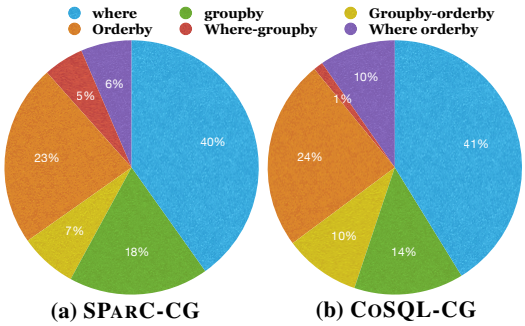
In our p-align method, the utterance embeddings of each interaction are extracted to decode the corresponding SQL statements. As shown in Figure 3, the decoder process of our p-align could be represented as:

$$\sum_{i=1}^{n} \prod_{t=1}^{T_i} p\left(a_t^i \mid \{a_1^i, \ldots, a_{t-1}^i\}, \mathbf{X}_{\leq i}\right). \quad (2)$$

In this way, our p-align method aligns corresponding parts of the input utterance to the previous queries and thus improves the compositional generalization ability of models.

	# Questions	# Non-CG Questions	# CG Questions
SPARC	1625	491	31
SPARC-CG	921	491	**372**
COSQL	1300	207	14
COSQL-CG	471	207	**167**

Table 1: The detailed statistics of SPARC-CG and COSQL-CG benchmark.



(a) SPARC-CG (b) COSQL-CG

Figure 4: Distributions of different modification patterns in SPARC-CG and COSQL-CG benchmark.

6 Experiment

In this section, we first perform more detailed statistics on our constructed SPARC-CG and COSQL-CG. Then we further analyze our benchmarks with current competitive Text-to-SQL models. Finally, several experiments are conducted to verify the effectiveness of our p-align method.

6.1 Benchmark statistics

The detailed statistics of SPARC-CG and COSQL-CG are shown in Table 1. We mainly count three metrics here: # Question, # Non-CG Questions, and # CG Questions, where # Question is the total

Methods / Datasets	SPARC			COSQL		
	Dev	**Non-CG**	**CG**	**Dev**	**Non-CG**	**CG**
SPIC (CONCAT) + BERT-Large (Liu et al., 2020)	55.3	63.4	18.9(36.4↓)	45.2	52.3	13.3(31.9↓)
SPIC (TURN) + BERT-Large (Liu et al., 2020)	54.6	62.1	18.2(36.4↓)	44.8	51.3	12.2(32.6↓)
SPIC (GATE) + BERT-Large (Liu et al., 2020)	54.3	62.4	17.3(37.0↓)	44.2	51.8	12.4(31.8↓)
RAT-SQL + SCORE (Yu et al., 2021)	60.4	69.6	22.4(38.0↓)	52.1	55.6	20.4(31.7↓)
LGESQL + ELECTRA-Large (Cao et al., 2021b)	65.0	73.4	25.3(39.7↓)	54.4	62.4	21.0(33.4↓)
LGESQL + STAR (Cai et al., 2022)	66.9	75.4	25.8(41.1↓)	59.7	68.4	26.3(33.4↓)
PICARD + T5-3B (Scholak et al., 2021)	-	-	-	56.9	58.1	21.5(35.4↓)
RASAT + T5-3B (Qi et al., 2022)	66.7	75.8	22.0(44.7↓)	58.8	67.9	20.4(38.4↓)

Table 2: Question match accuracy of current competitive models on three different benchmarks: Dev, Non-CG, and CG. For all the models, we adopt the given parameters.

number of questions, # CG Questions is the number of questions that meet the definition of compositional generalization in Section 3 and # Non-CG Questions is the number of in-domain questions (the templates and combination of templates are both seen in training). The Non-CG questions in SPARC-CG and COSQL-CG are obtained directly from the SPARC and COSQL datasets. The number of CG questions in our benchmarks is far more than in that SPARC and COSQL. Note that a large portion of the data in the SPARC and COSQL datasets is context-independent or has no context, which makes the sum of # Non-CG Questions and # CG Questions relatively small.

We present the components distributions of modification patterns of SPARC-CG and COSQL-CG in Figure 4. The most common component in modification patterns is *where*. *Orderby* and *groupby* also take a large proportion. There are also many modification patterns that include multiple components, such as *where-groupby* and *where-orderby*. Finally, the distributions of modification patterns in SPARC-CG and COSQL-CG are similar, which illustrates our benchmark construction's consistency. Note that the *select* components are not counted, as they are included in almost all modifications.

6.2 Experiment Setup

Models. We adopt many current competitive Text-to-SQL models to explore the impact of compositional generalization. SPIC (Liu et al., 2020) is a simple model which explores different methods to incorporate context questions, where SPIC (CONCAT) concatenates context questions with current questions, SPIC (TURN) employs a turn-level encoder to capture the inter-dependencies among questions in different turns and SPIC (GATE) uses a gate mechanism to compute the importance of each question. SCORE and STAR (Cai et al., 2022) are two specialized pre-trained models for RAT-SQL and LGESQL(Cao et al., 2021b) respectively. PICARD (Scholak et al., 2021) and RASAT (Qi et al., 2022) are two seq2seq based models based on pre-trained T5 model (Raffel et al., 2020).

Evaluation Metric. We mainly use the *question match* (QM) (Yu et al., 2019b) as our evaluation metric, which is the exact set matching score (Yu et al., 2018) over all questions. The exact set matching score decomposes predicted queries into SQL components such as SELECT and WHERE and then computes scores for each component. For each model, we report the QM on the origin SPARC/COSQL development set as well as the Non-CG and CG benchmarks. Note that the *interaction match* (Yu et al., 2019b) is not reported in our paper because we are only interested in the scores of the model on questions satisfying the compositional generalization condition.

6.3 Evaluation on SPARC-CG/COSQL-CG

We report the question match accuracy on SPARC and COSQL datasets under three benchmarks: Dev, Non-CG, and CG in Table 2.

Based on the above results, we summarize the following observations. (1) The accuracy of all models significantly decreases under the compositional generalization setting. Specifically, the QM on SPARC-CG and COSQL-CG decreases 39.3 and 33.6 on average compared to the origin development set, which indicates current models lack the compositional generalization ability. (2) The models perform better on the Non-CG benchmarks than the origin development set (8.4 and 6.5 on average for SPARC and COSQL respectively), which demonstrates that in-domain data are easily generalized. (3) CONCAT could better incorporate context

SQL Components	DEV	Non-CG	CG
SELECT	84.6	88.2	60.2
SELECT (no AGG)	86.3	89.3	62.9
WHERE	80.6	91.8	62.5
WHERE(no OP)	85.1	95.3	69.2
GROUP BY (no HAVING)	81.1	85.7	66.4
GROUP BY	76.9	81.6	54.5
ORDER BY	78.2	82.0	58.3
AND/OR	99.0	99.3	91.2
KEYWORDS	86.3	92.8	67.1

Table 3: Accuracy on the different SQL components. The reported results are the average results over STAR and RASAT on three benchmarks of SPARC.

questions than TURN and GATE. Therefore, our `p-align` is only designed for the CONCAT method. (4) The grammar tree-based decoder (LGESQL) and the larger language model (T5-3B) could help improve the compositional generalization ability.

6.4 Detailed Evaluation

Evaluation ay Different Levels of Difficulty. The SQL queries could be divided into four difficulty levels based on the complexity of SQL statements: easy, medium, hard and extra hard. To better demonstrate the performance in the compositional generalization setting, we conduct further evaluations on different levels of difficulties. As shown in Figure 5a-b, the STAR model performs worse on the CG benchmark than on the original development set at all difficulties, which further indicates the model's compositional generalization ability requires further improvement. Meanwhile, there is an obvious improvement in the Non-CG benchmark compared to the original development set.

Evaluation at Different Turns. We further illustrate the question match accuracy on three benchmarks with the increase of conversation turns in Figure 5c-d. The accuracy decreases sharply on the CG benchmark and the origin development set while staying stable on the non-CG benchmark. This suggests that the compositional generalization ability of models decreases with the increase of conversation turns.

Evaluation on different components. To better investigate the poor performance of the current competitive models under the compositional generalization setting, we further report the question match accuracy on different detailed SQL components in Table 3. The reported results are the average results over STAR and RASAT on three benchmarks of SPARC. As demonstrated in the table, nearly all components' accuracy significantly decreases under the compositional generalization setting, which illustrates the impact of compositional generalization on the models is balanced.

Methods	DEV	Non-CG	CG
SPARC			
SPIC (CONCAT) + BERT-Base	47.6	53.5	8.9
w. `p-align`	50.6	54.1	16.4(7.5↑)
SPIC (CONCAT) + BERT-Large	55.3	63.4	19.5
w. `p-align`	56.1	63.8	20.6(1.1↑)
LGESQL + ELECTRA-Large	65.0	73.4	25.3
w. `p-align`	64.8	73.0	26.2(0.9↑)
COSQL			
SPIC (CONCAT) + BERT-Base	39.2	35.0	5.2
w. `p-align`	40.5	36.2	9.6(4.4↑)
SPIC (CONCAT) + BERT-Large	45.2	52.3	12.2
w. `p-align`	45.5	52.7	14.4(2.2↑)
LGESQL + ELECTRA-Large	54.4	62.4	21.0
w. `p-align`	53.8	62.3	21.2(0.2↑)

Table 4: The results of different models w. & w/o `p-align` on three benchmarks of SPARC and COSQL.

Error component	STAR	RASAT	LGESQL
Context Info	24	15	25
Modification Info	149	136	139
Context & Modification Info	112	128	127

Table 5: Statistical analysis of different error types on SPARC-CG benchmark.

6.5 Evaluation of `p-align` method

Table 4 shows the results of different models with & without `p-align` on three benchmarks of SPARC and COSQL. We choose SPIC (CONCAT) + BERT-Base, SPIC (CONCAT) + BERT-large and LGESQL+ELECTRA-Large as our base models because other models are either customized pretrained models (STAR and SCORE) or with a too large model(T5-3B). All the hyperparameters are the same as the original model.

Overall, our `p-align` method significantly improves the performance of the model on the CG benchmarks, with an average improvement of 3.2 and 2.3 on the SParC-CG and CoSQL-CG benchmarks respectively. While the improvement on DEV and Non-CG benchmarks is relatively small, at 0.77 and 0.35 on average respectively, this suggests that our method is particularly effective in compositional generalization settings. These results support our hypothesis that improving alignment between utterances and queries can enhance the model's compositional generalization abilities, and should be considered as a potential direction for future research.

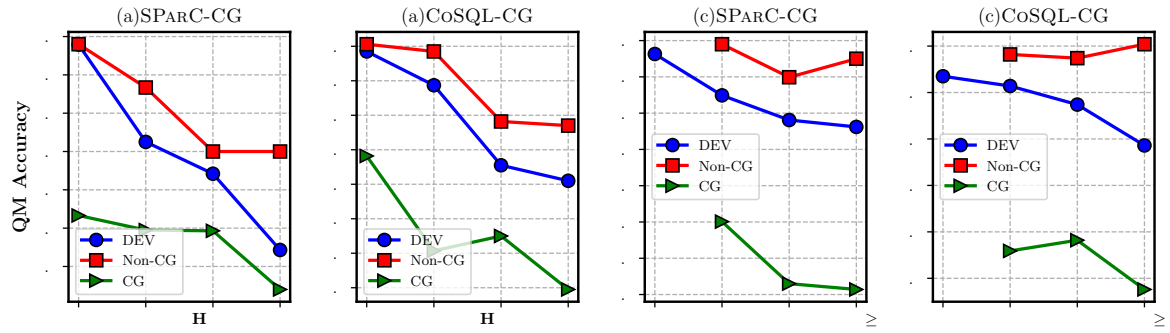


Figure 5: The results on different benchmarks by varying the difficulty levels of the data (a-b) and by varying the conversation turns(c-d). We use the STAR model here as an example.

Case #1 Error at the modification info	
Context :	Show all the owner information. | What about the dogs?
Question:	How many abandoned are in HUS?
Gold:	SELECT **Sum(abandoned_yn)** FROM Dogs WHERE breed_code = "HUS"
STaR:	SELECT **COUNT(*)** FROM Dogs WHERE Dogs.breed_code = "1" ✗
RASAT:	SELECT **COUNT(*)** FROM Dogs WHERE breed_code = "HUS" ✗

Case #2Error at the modification info	
Context :	What are all the nations| Which of those have a government that is a US territory?
Question:	What are the surface areas of those countries with independent year 1975 or 1278?
Gold:	SELECT SurfaceArea FROM country WHERE country.GovernmentForm = "US Territory" AND IndepYear = 1975 **Or IndepYear = 1278**
STaR:	SELECT country.SurfaceArea FROM country WHERE country.GovernmentForm = "1" OR country.IndepYear = 1975 ✗ Missing **Indepyear=1278**
RASAT:	SELECT SurfaceArea FROM country WHERE country.GovernmentForm = "US Territory" OR country.IndepYear = 1975 ✗ Missing **Indepyear=1278**

Case #3 Ignore the context info	
Context :	Which continents have an average life expectancy less than age 72?
Question:	Which of them have an average surface area smaller than 65209?
Gold:	SELECT Continent FROM country GROUP BY Continent HAVING **Avg(LifeExpectancy) < 72.0** AND Avg(SurfaceArea) < 652090.0
STaR:	SELECT Continent FROM country GROUP BY Continent HAVING AVG(SurfaceArea) < "1" ✗ Missing **Avg(LifeExpectancy)<72**
RASAT:	SELECT continent FROM country GROUP BY continent HAVING AVG(surfacearea) < 65209" ✗ Missing **Avg(LifeExpectancy)<72**

Case #4 Ignore the context & Error atmodification info	
Context :	How many dogs for each breed code?
Question:	Which ones have been abandoned, and in a breed code that contains many dogs.
Gold:	SELECT breed_code, **Count(*)** FROM Dogs **WHERE abandoned_yn = "1"** GROUP BY Dogs.breed_code HAVING Count(*) >= 1
STaR:	SELECT Dogs.breed_code FROM Dogs GROUP BY Dogs.breed_code HAVING COUNT(*) >= "1" ✗ Missing **both context and modification**
RASAT:	SELECT **abandoned_yn** FROM dogs GROUP BY Dogs.breed_code **HVAING count(*) >= 1** ✗ Missing **both context and modification**

Figure 6: Four examples from SPARC-CG benchmark and the corresponding wrong prediction results of STAR and RASAT. These examples are categorized according to the different errors.

6.6 Error analysis

To evaluate the compositional generalization ability of current models, we selected four incorrect prediction results from the SPARC-CG benchmark. For each example, we provided the context, the current question, the correct query, and the prediction results from STAR and RASAT.

As illustrated in Figure 6, in the first two scenarios, the models struggle to accurately interpret the changes brought about by current questions, despite maintaining a grasp of the context information. Conversely, in the third case, the models are able to interpret the modifications of the current question, but fail to take into account the context information. The fourth case represents the worst-case scenario, with the models unable to correctly parse either the modifications or the context information. Note that the incorrect results predicted by both models in the first three cases are similar, indicating that the failure of the current models to perform well in a compositional generalization setting is a widespread issue, not an isolated incident.

The presented case study categorizes three scenarios where current models make incorrect predictions, which include: failing to consider contextual information, inability to interpret modifications, and failing to understand both modifications and context. We further conduct statistical analysis on the SParC-CG benchmark in Table 5 and found that the majority of errors occur when models cannot interpret modifications. Additionally, when models neglect context, they also tend to misinterpret modifications. Interestingly, the proportion of errors for the different models evaluated in the study is quite similar, indicating that the compositional generalization challenges faced by these models are consistent across them.

7 Conclusion

In this study, we conduct the first exploration of compositional generalization in context-dependent Text-to-SQL scenarios. To support further research in this area, we construct two benchmarks named SPARC-CG and COSQL-CG composed of out-of-

distribution examples. Additionally, we introduce the `p-align` method to enhance the compositional generalization capabilities of existing models. Further experiments show that current models perform poorly on our constructed benchmarks and demonstrate the effectiveness of our `p-align` method. Also, with the recent advancements in generative language models, such as GPT3.5 and GPT4 (OpenAI, 2023), explorations into these models (Liu et al., 2023) should also constitute a significant part of future work.

Acknowledgement

The work was supported by the National Key Research and Development Program of China (No. 2019YFB1704003), the National Nature Science Foundation of China (No. 62021002), Tsinghua BNRist and Beijing Key Laboratory of Industrial Bigdata System and Application.

8 Limitations

In this paper, the approach to improve the compositional generalization under the context-dependent setting is insufficient. We only construct a better component alignment between inputs and outputs for models taking the concatenation of all utterances as input. However, it is important to note that other methods, such as using a turn-level encoder or implementing a gate mechanism, should also be considered. Additionally, other types of methods are also ignored. Future research could explore data augmentation techniques (Hu et al., 2022) and enhanced training objectives, such as meta-learning (Hu et al., 2021) and contrastive learning(Liu et al., 2022c; Li et al., 2023; Hu et al., 2020), as potential avenues for improvement.

References

Ekin Akyürek and Jacob Andreas. 2021. Lexicon learning for few-shot neural sequence modeling. *CoRR*, abs/2106.03993.

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.

Yitao Cai and Xiaojun Wan. 2020. IGSQL: Database schema interaction graph based neural model for context-dependent text-to-SQL generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6903–6912, Online. Association for Computational Linguistics.

Zefeng Cai, Xiangyu Li, Binyuan Hui, Min Yang, Bowen Li, Binhua Li, Zheng Cao, Weijie Li, Fei Huang, Luo Si, et al. 2022. Star: Sql guided pre-training for context-dependent text-to-sql parsing. *arXiv preprint arXiv:2210.11888*.

Ruisheng Cao, Lu Chen, Zhi Chen, Yanbin Zhao, Su Zhu, and Kai Yu. 2021a. LGESQL: Line graph enhanced text-to-SQL model with mixed local and non-local relations. 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 2541–2555, Online. Association for Computational Linguistics.

Ruisheng Cao, Lu Chen, Zhi Chen, Yanbin Zhao, Su Zhu, and Kai Yu. 2021b. LGESQL: line graph enhanced text-to-sql model with mixed local and non-local relations. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 2541–2555. Association for Computational Linguistics.

Zhi Chen, Lu Chen, Hanqi Li, Ruisheng Cao, Da Ma, Mengyue Wu, and Kai Yu. 2021. Decoupled dialogue modeling and semantic parsing for multi-turn text-to-SQL. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3063–3074, Online. Association for Computational Linguistics.

Yujian Gan, Xinyun Chen, Qiuping Huang, and Matthew Purver. 2022. Measuring and improving compositional generalization in text-to-SQL via component alignment. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 831–843, Seattle, United States. Association for Computational Linguistics.

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 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

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, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 908–921. Association for Computational Linguistics.

Xuming Hu, Yong Jiang, Aiwei Liu, Zhongqiang Huang, Pengjun Xie, Fei Huang, Lijie Wen, and Philip S Yu.

2022. Entda: Entity-to-text based data augmentation approach for named entity recognition tasks. *arXiv preprint arXiv:2210.10343*.

Xuming Hu, Lijie Wen, Yusong Xu, Chenwei Zhang, and Philip S. Yu. 2020. Selfore: Self-supervised relational feature learning for open relation extraction. In *Proc. of EMNLP*, pages 3673–3682.

Xuming Hu, Chenwei Zhang, Fukun Ma, Chenyao Liu, Lijie Wen, and Philip S. Yu. 2021. Semi-supervised relation extraction via incremental meta self-training. In *Findings of EMNLP*, pages 487–496.

Dieuwke Hupkes, Verna Dankers, Mathijs Mul, and Elia Bruni. 2020. Compositionality decomposed: How do neural networks generalise? *J. Artif. Intell. Res.*, 67:757–795.

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.

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 *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Brenden Lake and Marco Baroni. 2018a. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *International conference on machine learning*, pages 2873–2882. PMLR.

Brenden M. Lake and Marco Baroni. 2018b. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 2879–2888. PMLR.

Shu'ang Li, Xuming Hu, Li Lin, Aiwei Liu, Lijie Wen, and Philip S. Yu. 2023. A multi-level supervised contrastive learning framework for low-resource natural language inference. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:1771–1783.

Aiwei Liu, Xuming Hu, Li Lin, and Lijie Wen. 2022a. Semantic enhanced text-to-sql parsing via iteratively learning schema linking graph. In *Proc. of KDD*, pages 1021–1030.

Aiwei Liu, Xuming Hu, Lijie Wen, and Philip S Yu. 2023. A comprehensive evaluation of chatgpt's zero-shot text-to-sql capability. *arXiv preprint arXiv:2303.13547*.

Aiwei Liu, Honghai Yu, Xuming Hu, Shu'ang Li, Li Lin, Fukun Ma, Yawen Yang, and Lijie Wen. 2022b. Character-level white-box adversarial attacks against transformers via attachable subwords substitution. In *Proc. of EMNLP*.

Qian Liu, Bei Chen, Jiaqi Guo, Jian-Guang Lou, Bin Zhou, and Dongmei Zhang. 2020. How far are we from effective context modeling? an exploratory study on semantic parsing in context. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 3580–3586. ijcai.org.

Shuliang Liu, Xuming Hu, Chenwei Zhang, Shu'ang Li, Lijie Wen, and Philip S. Yu. 2022c. Hiure: Hierarchical exemplar contrastive learning for unsupervised relation extraction. In *Proc. of NAACL-HLT*, pages 5970–5980.

OpenAI. 2023. Gpt-4 technical report. *ArXiv*, abs/2303.08774.

Jiexing Qi, Jingyao Tang, Ziwei He, Xiangpeng Wan, Chenghu Zhou, Xinbing Wang, Quanshi Zhang, and Zhouhan Lin. 2022. RASAT: integrating relational structures into pretrained seq2seq model for text-to-sql. *CoRR*, abs/2205.06983.

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, Pawel Krzysztof Nowak, Tal Linzen, Fei Sha, and Kristina Toutanova. 2022b. 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, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 4341–4362. 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. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Torsten Scholak, Nathan Schucher, and Dzmitry Bahdanau. 2021. PICARD: Parsing incrementally for constrained auto-regressive decoding from language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9895–9901, Online and Punta Cana, Dominican Republic. 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.

Bailin Wang, Richard Shin, Xiaodong Liu, Oleksandr Polozov, and Matthew Richardson. 2020. RAT-SQL: Relation-aware schema encoding and linking for text-to-SQL parsers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7567–7578, Online. Association for Computational Linguistics.

Runze Wang, Zhen-Hua Ling, Jingbo Zhou, and Yu Hu. 2021. Tracking interaction states for multi-turn text-to-sql semantic parsing. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 13979–13987. AAAI Press.

Dongling Xiao, Linzheng Chai, Qian-Wen Zhang, Zhao Yan, Zhoujun Li, and Yunbo Cao. 2022. CQR-SQL: conversational question reformulation enhanced context-dependent text-to-sql parsers. *CoRR*, abs/2205.07686.

Tao Yu, Rui Zhang, Heyang Er, Suyi Li, Eric Xue, Bo Pang, Xi Victoria Lin, Yi Chern Tan, Tianze Shi, Zihan Li, Youxuan Jiang, Michihiro Yasunaga, Sungrok Shim, Tao Chen, Alexander Fabbri, Zifan Li, Luyao Chen, Yuwen Zhang, Shreya Dixit, Vincent Zhang, Caiming Xiong, Richard Socher, Walter Lasecki, and Dragomir Radev. 2019a. CoSQL: A conversational text-to-SQL challenge towards cross-domain natural language interfaces to databases. 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 1962–1979, Hong Kong, China. Association for Computational Linguistics.

Tao Yu, Rui Zhang, Alex Polozov, Christopher Meek, and Ahmed Hassan Awadallah. 2021. Score: Pre-training for context representation in conversational semantic parsing. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir Radev. 2018. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-SQL task. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3911–3921, Brussels, Belgium. Association for Computational Linguistics.

Tao Yu, Rui Zhang, Michihiro Yasunaga, Yi Chern Tan, Xi Victoria Lin, Suyi Li, Heyang Er, Irene Li, Bo Pang, Tao Chen, Emily Ji, Shreya Dixit, David Proctor, Sungrok Shim, Jonathan Kraft, Vincent Zhang, Caiming Xiong, Richard Socher, and Dragomir Radev. 2019b. SParC: Cross-domain semantic parsing in context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4511–4523, Florence, Italy. Association for Computational Linguistics.

Hao Zheng and Mirella Lapata. 2022. Disentangled sequence to sequence learning for compositional generalization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 4256–4268. Association for Computational Linguistics.

Yanzhao Zheng, Haibin Wang, Baohua Dong, Xingjun Wang, and Changshan Li. 2022. HIE-SQL: History information enhanced network for context-dependent text-to-SQL semantic parsing. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2997–3007, Dublin, Ireland. Association for Computational Linguistics.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section 8

☑ A2. Did you discuss any potential risks of your work?
Section 8

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Abstract, Section 1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Section 5, Section 6

☑ B1. Did you cite the creators of artifacts you used?
Section 5, Section 6

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Section 5, Section 6

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Section 5, Section 6

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
No response.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
No response.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Section 6

C ☑ Did you run computational experiments?

Section 6

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section 6

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 6

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 6

☐ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Not applicable. Left blank.

D ☑ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Section 4,6

☑ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Section 4

☒ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
The authors annotate the data.

☑ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Section 4, 6

☑ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Section 4, 6

☑ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Section 4, 6