Exploiting Reversible Semantic Parsing and Text Generation for Error Correction with Pre-trained LLMs

Anonymous ACL submission

Abstract

Semantic parsing and text generation are reversible processes when working with Discourse Representation Structures (DRS). Obviously, errors can arise in both the parsing (text-to-DRS) and generation (DRS-to-text). This paper presents an approach that exploits the reversible nature of these tasks to automatically correct such errors without additional model training. We leverage pre-trained large language models (LLMs) in two pipeline setups: Pars-Gen-Pars and Gen-Pars-Gen, where the output of one model serves as the input to the next. In the Pars-Gen-Pars pipeline, input text is parsed into a DRS, then used to generate text, which is finally parsed again. Conversely, the Gen-Pars-Gen pipeline starts with a DRS, generates text, parses it, and regenerates text from the parsed DRS. Interestingly, by propagating the data through these reversible pipelines, errors from the initial parse or generation step can be mitigated, instead of being amplified. Experiments on the Parallel Meaning Bank dataset demonstrate the efficacy of our approach, with improved performance over baseline models on semantic parsing (SMATCH) and text generation (BLEU, METEOR, COMET, chrF, BERTScore) metrics. Our error analysis also sheds light on the types of mistakes addressed by each pipeline setup. The proposed method offers a simple yet effective way to enhance DRS-based natural language processing without costly model retraining.

1 Introduction

Discourse Representation Structure (DRS) provides a formal semantic representation of natural language that captures meaning beyond the literal text (Kamp and Reyle, 1993). DRS derived from Discourse Representation Theory (DRT) offers a comprehensive formal meaning representation that spans a wide range of linguistic phenomena (Kamp et al., 2010). These include anaphors, presuppositions, temporal expressions, and multisentence discourses, as well as the nuanced semantics of negation, modals, and quantification (Kamp and Reyle, 2013; Jaszczolt and Jaszczolt, 2023). Notably, DRS enables a language-neutral meaning representation, allowing a single representation to be applied across texts in different languages (Bos, 2021).

DRS has found applications in various natural language processing (NLP) tasks such as machine translation (van Noord et al., 2018), semantic parsing (mapping text to DRS) (Noord, 2019; van Noord et al., 2019), and text generation (mapping DRS to text) (Wang et al., 2021a; Amin et al., 2022; Liu et al., 2021; Amin et al., 2024). While different models have been proposed for these tasks, an interesting property is that they are reversible processes—the output of one can serve as the input of the other. In literature, semantic parsing and generation approaches have been studied separately for each language, focusing mainly on English. This approach requires building distinct models from scratch for each task and language, which is limited by the lack of available data.

In recent years, large pre-trained language models (LLMs) have significantly advanced NLP tasks. However, semantic parsing and text generation have been unable to fully leverage these advancements, as the explicit representation of meaning is not inherently integrated into the training of these models (Amin et al., 2024). Indeed, despite recent advances, both DRS semantic parsing and text generation are challenging and error-prone (Wang et al., 2023a). Parsing mistakes can lead to incorrect or incomplete meaning representations, while generation errors result in disfluent or meaningless text (Wang et al., 2021a). Traditionally, improving performance on these tasks involves costly retraining of models on larger datasets or using more complex architectures.

In this work, we propose a simple yet effective approach leveraging the reversible nature of semantic parsing and text generation to automatically correct errors without additional model training. Our method utilizes LLMs in two pipeline setups: 1) Pars-Gen-Pars, where input text is parsed, used to generate text, and then parsed again; and 2) Gen-Pars-Gen, where a DRS is used to generate text, which is parsed and then used to regenerate text. By propagating the data through these reversible pipelines, errors from the initial parsing or generation step can be mitigated in the subsequent stages.

We evaluate our approach on the Parallel Meaning Bank[1] (PMB) dataset, a benchmark for DRS-based semantic processing (Abzianidze et al., 2017). Results show that the proposed Pars-Gen-Pars and Gen-Pars-Gen pipelines improve performance over baseline models on both semantic parsing (measured by SMATCH) and text generation (measured by BLEU, METEOR, COMET, CHRF, BERT-SCORE) metrics. Furthermore, our error analysis provides insights into the types of mistakes each pipeline setup addresses.

The research questions addressed in this paper are:

- How can we leverage the reversible nature of semantic parsing and text generation with DRS to automatically correct errors?
- Can LLMs be effectively utilized in a pipeline approach to mitigate errors without additional model training?
- What are the performance improvements achieved by the proposed reversible pipelines compared to baseline models?
- Which types of errors are more effectively addressed by the Pars-Gen-Pars and Gen-Pars-Gen pipeline?
- What are the capabilities and limitations of the reversible pipeline approaches in correcting different error categories?

The key contributions of this paper are: (1) proposing a novel method for error correction in DRS-based NLP tasks by exploiting reversibility, (2) demonstrating the effectiveness of this approach using LLMs without costly retraining, and (3) analyzing the capabilities and limitations of the proposed pipelines through rigorous error analysis[2].

[1]The PMB is developed at the University of Groningen as part of the NWO-VICI project "Lost in Translation – Found in Meaning" (Project number 277-89-003), led by Johan Bos.

[2]Code can be provided on acceptance.

The remaining paper is structured as follows: Section 2 describes DRS and reviews related work in semantic parsing and text generation; Section 3 describes our methodology, pipeline configurations, and experimental results in detail; Section 4 presents a detailed error analysis with the discussion regarding the mitigation of errors; finally Section 5 concludes the paper, highlights limitations, and suggests directions for future research.

2 Background and Related Work

This section provides an overview of DRS, the formal meaning representation tool employed in our approach, and reviews the pertinent background and related research in the domains of semantic parsing and text generation. In Section 2.1, we provide a basic background on DRS formalis, and in Sections 2.2 and 2.3 we report the most important reference for parsing to and generating from DRS respectively.

2.1 Discourse Representation Structures

As a thorough formal meaning representation, DRS captures the main idea of the text and deals with a number of linguistic occurrences, such as temporal expressions and anaphoras (Bos, 2023). Unlike other formalisms used in large-scale semantic annotation initiatives, like Abstract Meaning Representation (AMR) (Banarescu et al., 2013), DRS is distinguished by its capacity to handle logical negation, quantification, and discourse relations, in addition to offering complete word sense disambiguation and a language-neutral meaning representation.

Figure 1 illustrates the different formats that can be used to express DRS. Using boxes to hold discourse referents and conditions is one frequent notation. Discourse referents, like $x1$, serve as stand-ins for newly presented entities. Using roles or comparison operators, conditions describe these referents' attributes, including the concepts to which they belong and their relationships with other referents. Concepts are based on WordNet synsets (Fellbaum, 1998), such as $male.n.02$. VerbNet (Bonial et al., 2011) is a resource used to generate thematic roles; examples include Agent. Operators like $<$, $>$, $\neq$, and $\neg$ are used to create negations and comparisons between entities. Furthermore, conditions might be complex, representing rhetorical linkages between many sets of conditions or logical relations (negation, $\neg$).

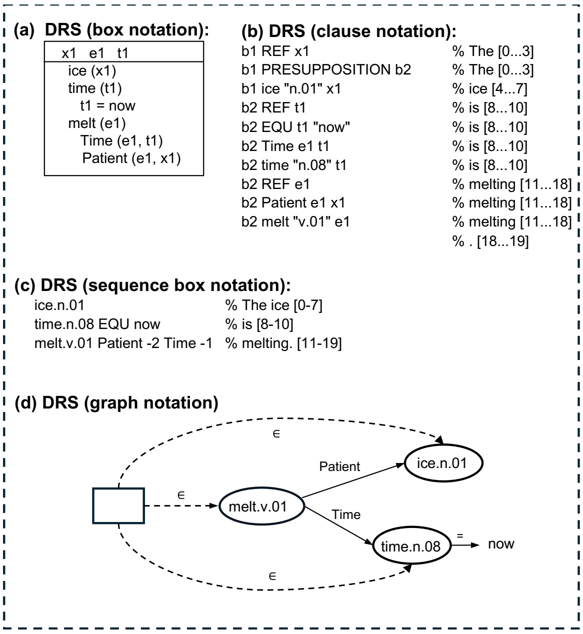

Figure 1: Different graphical representations of DRS for the text "The ice is melting.".

In order to make integration with machine learning models easier, the box notation (Figure 1(a)) is converted into clause notation (Figure 1(b)) (van Noord et al., 2018). This conversion entails rearranging the structure so that the discourse referents and conditions are positioned before the label of the box.

Sequence Box Notation (SBN) (Figure 1(c)) is a simplified version of DRS that emphasizes the sequential arrangement of logical entities (Bos, 2023). Each word's meaning is organized according to an entity-role-index format in SBN, where indices connect entities and roles and decorate the connections. Discourse relations, like NEGATION and ELABORATION, are slightly modified to signal the beginning of a new context. Subsequent indices, marked with comparison symbols (<,>), establish links between the newly formed context and another context. SBN can be visually represented as a directed acyclic graph, as seen in Figure 1(d).

2.2 Text-to-DRS Parsing

Rule-based and neural network-based techniques are the two main categories into which traditional DRS parsing techniques can be divided. The Boxer system is a well-known paradigm among rule-based approaches that blend statistical methodologies with rules (Bos, 2008). In order to achieve performance that is on par with or even better than BERT-based models, (Poelman et al., 2022a) has more recently built a multilingual DRS parser that makes use of already-existing Universal Dependency parsers. In this sector, neural models have emerged as the main method because of their persistent high performance (van Noord et al., 2018; Wang et al., 2023a; Amin et al., 2024). In addition to sequence-to-sequence models, two separate research streams concentrate on tree-based (Liu et al., 2021) and graph-based (Fancellu et al., 2019; Fu et al., 2020) techniques, with (Fu et al., 2020) representing the initial attempt at multilingual DRS parsing.

2.3 DRS-to-Text Generation

Unlike the well-established tenacity of DRS parsing, NLP researchers have only recently turned their attention to the task of generating text from DRS (Basile and Bos, 2011; Wang et al., 2021a; Amin et al., 2022; Wang et al., 2023a; Amin et al., 2024). Like DRS parsing, rule-based methods (Basile and Bos, 2011) and neural network-based methods (Wang et al., 2021a; Amin et al., 2022; Wang et al., 2023a; Amin et al., 2024) are the two main categories of past work on this generating problem. Initial efforts in DRS-to-Text generation identified key challenges such as lexicalization, aggregation, and generating referencing expressions (Basile and Bos, 2011). A recent practical implementation of text generation utilized bidirectional LSTM (bi-LSTM) based sequence-to-sequence models to produce English text from DRS (Wang et al., 2021a; Amin et al., 2022). To address the difficulties in generating text from DRS, including condition ordering and variable name issues, tree-LSTM-based techniques have gained popularity (Liu et al., 2021). The development of the mBART-based multilingual DRS-to-Text generation model coincided with the emergence of state-of-the-art Transformer models (Wang et al., 2023a).

3 Method and Results

Our study departs from the standard rule-based and neural network-based methods for DRS parsing and text generation. We offer a novel perspective that takes advantage of the DRS reversible capabilities that do not require any explicit design of rules or external tools, in contrast to rule-based systems like Boxer or the more recent multilingual DRS parser which rely on hand-crafted rules and commercial dependency parsers (Bos, 2008; Poelman et al., 2022a). Instead, our work presents a pipeline-based approach for semantic parsing and text generation that takes advantage of the complementary benefits offered by LLMs. Our approach cascades these reversible processes into two different pipelines, Pars-Gen-Pars and Gen-Pars-Gen, so as to automatically fix problems that might occur in the generation or parsing phase, without requiring extra rule engineering or model training.

The model architecture, which is based on byT5 (Xue et al., 2022)—a fine-tuned model on an augmented version of the PMB dataset— is described in this section. It describes the pipeline configurations for Pars-Gen-Pars and Gen-Pars-Gen that are intended to reduce errors in processes related to semantic parsing and text generation, respectively. A discussion of the evaluation metrics used, such as SMATCH (Cai and Knight, 2013) for semantic parsing and BLEU (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005), COMET (Rei et al., 2020), chrF (Popović, 2015), and BERTScore (Hanna and Bojar, 2021) for text generation.

3.1 Basic Text-To-Text Transfer Transformer Model

In our experimentation, we employed the standard transformer model belonging to the Text-To-Text Transfer Transformers (T5) family (Unanue et al., 2023), specifically the byT5 (Xue et al., 2022) variant, due to its superior performance compared to other T5 variants, including mT5 (Xue et al., 2021) and T5 (Unanue et al., 2023) itself. Our approach deviates from traditional experimental methods in the following key aspects: (1) Conventional methods can be computationally expensive and time consuming, as they frequently require pre-training or fine-tuning a large language model (LLM) for task-specific applications. On the other hand, our implementation does not require any additional model pre-training or fine-tuning. (2) While the pre-training of byT5 was performed on the mC4 dataset, which implies no prior knowledge of DRS, we leveraged a fine-tuned version of the byT5 model obtained from the Hugging Face repository[3]. These two fine-tuned models (one for parsing and one for generation) are state-of-the-art models for semantic parsing and text generation tasks related to DRS.

[3]We are not providing the link to this model to maintain anonymity, which will be shared upon acceptance.

3.2 Pars-Gen-Pars Pipeline

The Pars-Gen-Pars pipeline is designed to mitigate errors in the semantic parsing task by propagating the input text through three stages: parsing, generation, and parsing again. The pipeline operates as follows: (1) The input text is first processed by the parser model, which generates a DRS. (2) The generated DRS is then passed to the generator model, which produces a text output based on the DRS representation. (3) Finally, the generated text is fed into the same parser model, resulting in a new DRS representation. Figure 2 displays the graphical representation of the proposed Pars-Gen-Pars pipeline.

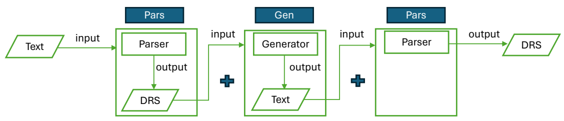

Figure 2: Graphical representation of Pars-Gen-Pars pipeline.

3.3 Gen-Pars-Gen Pipeline

Similarly, the Gen-Pars-Gen pipeline is designed to address errors in the text generation task by propagating the input DRS through three stages: generation, parsing, and generation again. The pipeline operates as follows: (1) The input DRS is first processed by the generator model, which produces a text output. (2) The generated text is then passed to the parser model, resulting in a new DRS representation. (3) Finally, the parsed DRS is fed into the same generator model, producing a new text output. Graphically, the Gen-Pars-Gen pipeline is shown in Figure 3.

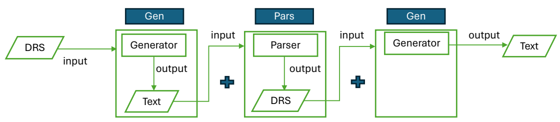

Figure 3: Graphical representation of Gen-Pars-Gen pipeline.

By iteratively propagating the data through these reversible pipelines, errors introduced in the initial parsing (generation) stage can be potentially corrected in the subsequent generation (parsing) and parsing (generation) stages, leveraging the complementary strengths of the pre-trained models.

3.4 Experimentation and Results

For our experiments, we leveraged two state-of-the-art models—a generator (DRS-to-Text) based on byT5 and a parser (Text-to-DRS) based on byT5—that were fine-tuned on the augmented PMB dataset. These models were used straight out of the literature, without performing any additional pre-training or fine-tuning, and they performed better than earlier methods. We assessed two suggested pipelines using these pre-trained models, Pars-Gen-Pars and Gen-Pars-Gen.

3.4.1 Pars-Gen-Pars Evaluation

We used the method by (Poelman et al., 2022b) to convert the linearized DRS into the Penman format (Kasper, 1989) for the Pars-Gen-Pars pipeline. Next, we computed the overlap between the system output and the gold standard by computing the F1-score of matched triples using SMATCH—a typical assessment tool used in Abstract Meaning Representation (AMR) parsing (Cai and Knight, 2013). Our findings show that the Pars-Gen-Pars pipeline significantly enhances semantic parsing performance compared to the standalone parser. The Pars-Gen-Pars pipeline produced an improved F1-score of 94.05, indicating a considerable increase in accuracy, compared to the parser model's 93.56 SMATCH F1-score—see Table 1 for semantic parsing result comparing Pars-Gen-Pars pipeline with standalone parser and literature based implementations.

3.4.2 Gen-Pars-Gen Evaluation

We evaluated the quality of the generated text for the Gen-Pars-Gen pipeline using three types of automatic assessment metrics: (1) Rule-based automatic measures: BLEU (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005), and chrF (Popović, 2015), which are based on the word or character overlap between the generated text and the gold reference; (2) Neural model-based measure: COMET (Rei et al., 2020), a neural evaluation metric trained on human ratings of machine translation outputs; and (3) Pre-trained model-based measure: BERT-Score (Hanna and Bojar, 2021), which leverages pre-trained BERT models to compute the semantic similarity between the generated and reference texts. The outcomes clearly show that the Gen-Pars-Gen pipeline performed better than the standalone generation model in every evaluation criteria. Notably, the BLEU score improved from 73.45 to 74.18, METEOR increased from 55.61 to 55.97, COMET rose from 95.81 to 95.89, chrF increased from 84.96 to 85.30, and BERT-Score improved from 98.54 to 98.58. Text generation results comparing the Gen-Pars-Gen pipeline with the standalone generator are shown in Table 1.

These improvements demonstrate how well our method, which makes use of the reversible nature of the processes and the complementary advantages of pre-trained language models, mitigates errors in semantic parsing and text generation tasks.

4 Analysis and Discussion

In this section, we delve into a detailed exploratory analysis of the errors produced by the standalone parser and generator models and examine the types of corrections facilitated by the Pars-Gen-Pars (Section 4.1) and Gen-Pars-Gen (Section 4.2) pipelines. Additionally, in Section 4.3 we investigate when and why the pipeline works to mitigate errors—revealing its strength.

4.1 Parser Errors and Corrections with Pars-Gen-Pars Pipeline

The standalone parser makes certain types of errors when it generates DRS from input text (Wang et al., 2023a; Zhang et al., 2024). We categorize these errors and show how our Pars-Gen-Pars pipeline effectively reduces these errors.

Wrong WordNet Sense Assignment. The parser frequently assigns the wrong WordNet sense numbers to nouns, adjectives, adverbs, and verbs in the generated DRS. In the sentence "Let's fly a kite.", for instance, the parser wrongly assigns the verb "fly" to fly.v.01 whereas the gold DRS links it with the meaning fly.v.05. Such sense defects are successfully corrected by the Pars-Gen-Pars pipeline, yielding in this instance the accurate sense fly.v.05 (see Table 2, example 1).

Missing Logical Concepts. Sometimes the parser is unable to produce all of the logical concepts needed to correctly represent the input text in the DRS. The concepts "time.n.08 EQU now" and "Time -1" for the text "Is your father Spanish?" are included in the gold DRS but are left out by the parser. Nevertheless, the Pars-Gen-Pars pipeline incorporates these absent concepts accurately, improving the correctness of DRS (see Table 2, ex. 2).

Table 1: Experimental results of parsing and generation with and without pipeline approach. Bold represents the best scores in all experiments of semantic parsing and text generation. †shows that the pipeline results are statistically significant (using the Wilcoxon Signed Ranked Test) compared to the results without the pipeline. Note: S-Par. = Semantic Parsing; G = Gold; S = Silver; and B = Bronze version(s) of Parallel Meaning Bank (PMB). S-F1 = SMATCH F1-Score; MET. = METEOR; CMT. = COMET; B_Scr. = BERT_Score.

Experimentation Type	Model Type	PMB Type	S-Par. S-F1	Generation Results BLEU	MET.	CMT.	chrF	B_Scr.
(Amin et al., 2022)	bi-LSTM	G	–	52.30	41.53	–	–	–
(Amin et al., 2024)	byT5	G	–	57.15	45.90	–	–	97.02
(Wang et al., 2021a)	bi-LSTM	G+S	–	69.30	51.80	–	–	–
(van Noord et al., 2019)	NeuDRS	G+S	84.50	–	–	–	–	–
(Amin et al., 2022)	bi-LSTM	G+S	–	72.38	53.18	–	–	–
(Wang et al., 2023b)	bi-LSTM	G+S	91.00	–	–	–	–	–
(Wang et al., 2021b)	bi-LSTM	G+S	88.10	–	–	–	–	–
(Zhang et al., 2024)	DRS-MLM	G+S	91.50	71.90	54.90	93.00	–	–
without pipeline	byT5	G+S	93.56	73.45	55.61	95.81	84.96	98.54
with pipeline	byT5	G+S	**94.05†**	**74.18†**	**55.97†**	**95.89†**	**85.30†**	**98.58†**

Hallucinating Incorrect Thematic Roles. The generation of false or delusional logical notions that are inconsistent with the input text is another kind of error that the parser reports. The gold DRS, for instance, designates the thematic role "Agent -1" to represent the subject "I" in the text "I caught a fish!" However, the parser mistakenly produces "Recipient" in its place. By successfully avoiding these hallucinations, the Pars-Gen-Pars pipeline produces the accurate thematic role with the correct index "Agent -1" (see Table 2, ex. 3).

Wrong Index Assignment. In DRS, indices are essential for referring to and connecting various logical concepts. Occasionally, the parser assigns erroneous indices, resulting in logical ambiguities. In the case of the text, "Mayuko designed a dress for herself." for example, the gold DRS refers to the concept "female.n.02 ANA -4" (indicating "herself") using the thematic role index "Beneficiary +3". But the parser produces the incorrect thematic role index "Beneficiary +1" pointing erroneously to "time.n.08 TPR now". The Pars-Gen-Pars pipeline ensures logical coherence inside the DRS by appropriately assigning the correct thematic role indexes in each case e.g., "Beneficiary +3" for the example under discussion (see Table 2, ex. 4).

By propagating the data through the Pars-Gen-Pars pipeline, errors made by the initial parser are effectively corrected in the subsequent generation and parsing stages. The complementary strengths of the LLMs in the pipeline, combined with the reversible nature of the tasks, enable the mitigation of these diverse error types. The examination of errors shows the shortcomings of the standalone parser and emphasizes the benefits of the Pars-Gen-Pars pipeline in terms of improving the quality and comprehensiveness of the DRS representations that are produced.

4.2 Generation Errors and Corrections with Gen-Pars-Gen Pipeline

Our investigation identifies certain primary categories of problems that the standalone generator model produces when it generates text from DRS representations (Wang et al., 2023b; Amin et al., 2024). We classify and explain these mistakes, showing how the suggested Gen-Pars-Gen pipeline fixes them.

Grammatical Errors. The generator model sometimes produces grammatically incorrect text, as exemplified by the DRS "high.a.02 Value ? AttributeOf +1 mountain.n.01 Name "Mount Kinabalu"" and the incorrect generation "How high of Mount Kinabalu?" instead of the grammatically correct "How high is Mount Kinabalu?". Such grammatical faults are successfully mitigated by the Gen-Pars-Gen pipeline (see Table 3, example 1).

Word Position Swapping. Sometimes the generator model produces inaccurate outputs because it rearranges the words in the generated text. Considering the DRS "person.n.01 Name ? found.v.02 Agent -1 Time +1 Theme +3 time.n.08 TPR now striptease.n.02 club.n.07 Name "Chippendale" Theme -1", the generator yields the incorrect answer "Who founded the striptease club Chippen-

Gold Text	Pars (DRS)	Pars-Gen (Text)	Pars-Gen-Pars (DRS)	Gold DRS
Let's fly a kite.	time.n.08 TSU now person.n.01 EQU speaker fly.v.01 Time -2 Agent -1 Theme +1 kite.n.03	Let's fly kites.	time.n.08 TSU now person.n.01 EQU speaker fly.v.05 Time -2 Agent -1 Theme +1 kite.n.03	time.n.08 TSU now person.n.01 EQU speaker fly.v.05 Time -2 Agent -1 Theme +1 kite.n.03
Is your father Spanish?	person.n.01 EQU hearer person.n.01 Role +1 father.n.01 Of -2 be.v.03 Theme -2 Source +1 country.n.02 Name "spain"	Your father is Spanish.	person.n.01 EQU hearer person.n.01 Role +1 father.n.01 Of -2 time.n.08 EQU now be.v.03 Theme -3 Time -1 Source +1 country.n.02 Name "spain"	time.n.08 EQU now person.n.01 EQU hearer person.n.01 Role +1 father.n.01 Of -2 be.v.03 Time -4 Theme -2 Source +1 country.n.02 Name "spain"
I caught a fish!	person.n.01 EQU speaker catch.v.08 Recipient -1 Time +1 Theme +2 time.n.08 TPR now fish.n.01	I caught a fish.	person.n.01 EQU speaker catch.v.08 Agent -1 Time +1 Theme +2 time.n.08 TPR now fish.n.01	person.n.01 EQU speaker catch.v.08 Agent -1 Time +1 Theme +2 time.n.08 TPR now fish.n.01
Mayuko designed a dress for herself.	female.n.02 Name "Mayuko" design.v.03 Agent -1 Time +1 Result +2 dress.n.01 Beneficiary +1 time.n.08 TPR now female.n.02 ANA -4	Mayuko designed this dress for herself.	female.n.02 Name "Mayuko" design.v.03 Agent -1 Time +1 Result +2 Beneficiary +3 time.n.08 TPR now dress.n.01 female.n.02 ANA -4	female.n.02 Name "Mayuko" design.v.03 Agent -1 Time +1 Result +2 Beneficiary +3 time.n.08 TPR now dress.n.01 female.n.02 ANA -4

Table 2: Analyzing parser errors and mitigating these errors through the Pars-Gen-Pars pipeline with the visualization of in-between transition states. The errors are highlighted in red and mitigations are in blue.

dale?" rather than the correct text "Who founded the Chippendale striptease club?". Such word order problems are effectively fixed by the Gen-Pars-Gen pipeline (see Table 3, ex. 2).

Singular Plural Inconsistencies. The generator model occasionally has trouble producing words in their correct singular or plural forms, as illustrated by the DRS "male.n.02 Name "Jack" book.n.01 Creator -1 time.n.08 EQU now interesting.a.01 AttributeOf -2 Time -1" for the gold text "Jack's book is interesting.". Nevertheless, the generator produces "Jack's books are interesting." inaccurately. Even though these singular plural inconsistencies are linguistically and contextually accurate, they are penalized by automatic evaluation measures. The proper singular or plural form is accurately identified and generated by the Gen-Pars-Gen pipeline (see Table 3, ex. 3).

Altered Textual Representations. Sometimes the generator model changes how some concepts are expressed textually, but the text that is produced is still accurate in terms of semantics and context. For instance, the generator generates "What is the square root of a hundred?" by substituting "a hundred" for "100" given the DRS "entity.n.01 EQU ? be.v.06 Theme -1 Co-Theme +1 square_root.n.01 Of +1 number.n.02 EQU 100", whereas the gold text is "What's the square root of 100?". Evaluation measures that emphasize on the precise textual overlaps, such as BLEU, METEOR, and chrF, punish these modifications even when they are accurate. Such representation modifications are mitigated by the Gen-Pars-Gen pipeline (see Table 3, ex. 4).

4.3 Revealing the Pipeline Approach

In this Section, we first consider the impact of the sentence length on the performance of the pipeline, and second, we speculate on the mechanism of the pipeline that corrects some errors.

Considering the question "When does the pipeline work?" we need to consider the length of input. In order to answer this question, we decide to analyze the performances of both parsing and generation pipelines for the sentences of the test set. The analysis (see Figure 4 for parsing, and Figures 6, 7, 5, 8, 9 for generation) reveals that both the parser and generator models exhibit performance variations across different sentence length ranges. For the semantic parsing task, the parser model struggles more with longer sentences, particularly in the token length range of 45 to 70 tokens. This performance degradation can be attributed to the increased complexity of capturing long-range dependencies and generating accurate logical concepts for longer sentences. Interestingly, the parser also exhibits a drop in performance for very short sentences, ranging from 10 to 15 tokens. This behavior suggests that the model may hallucinate or struggle to capture the exact semantic information for extremely short inputs. However, the parser performs relatively better for sentences with intermediate lengths, ranging from 20 to 45 tokens, indicating a more balanced performance in this range. Similar trends are seen in text generation[4],

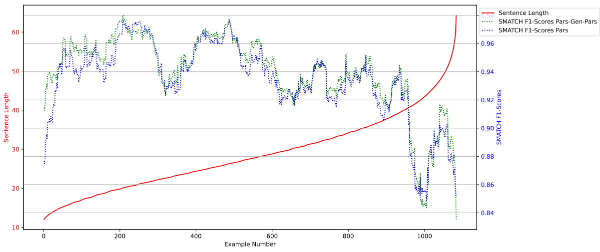

Figure 4: Sentence by sentence SMATCH F1-Scores along with sentence length for standalone Parser and Pars-Gen-Pars pipeline approaches.

[4]Here we explain the behavior of COMET only as it correlates more with human evaluation (Wang et al., 2023a). Graphical representations for other generation measures like chrF are described in the appendix.

Gold DRS	Gen (Text)	Gen-Pars (DRS)	Gen-Pars-Gen (Text)	Gold Text
high.a.02 Value ? AttributeOf +1 mountain.n.01 Name "Mount Kinabalu"	How high of Mount Kinabalu?	high.a.02 Time +1 AttributeOf +2 time.n.08 EQU now mountain.n.01 Name "Mount Kinabalu"	How high is Mount Kinabalu?	How high is Mount Kinabalu?
person.n.01 Name ? found.v.02 Agent -1 Time +1 Theme +3 time.n.08 TPR now striptease.n.02 club.n.07 Name "Chippendale" Theme -1	Who founded the striptease club Chippendale?	person.n.01 Name ? found.v.01 Agent -1 Time +1 Theme +3 time.n.08 TPR now striptease.n.01 club.n.06 Name "Chippendale" Theme -1 club.n.06 EQU -1	Who founded the Chippendale striptease club?	Who founded the Chippendale striptease club?
male.n.02 Name "Jack" book.n.01 Creator -1 time.n.08 EQU now interesting.a.01 AttributeOf -2 Time -1	Jack's books are interesting.	male.n.02 Name "Jack" book.n.01 User -1 time.n.08 EQU now interesting.a.01 AttributeOf -2 Time -1	Jack's book is interesting.	Jack's book is interesting.
entity.n.01 EQU ? be.v.06 Theme -1 Co-Theme +1 square_root.n.01 Of +1 number.n.02 EQU 100	What is the square root of a hundred?	entity.n.01 EQU ? be.v.02 Co-Theme -1 Time +1 Theme +2 time.n.08 EQU now square_root.n.01 PartOf +1 entity.n.01 Quantity +1 quantity.n.01 EQU 100	What's the square root of 100?	What's the square root of 100?

Table 3: Analyzing generation errors and mitigating these errors through the Gen-Pars-Gen pipeline with the visualization of in-between transition states. The errors are highlighted in red and mitigations are in blue.

albeit with varying ranges of sentence length. For sentences that are between 12 and 17 tokens long i.e., short sentences, the generator model performs badly and hallucinates. The performance rapidly deteriorates with sentence length, indicating the difficulty faced by the model with longer and more intricate linguistic formulations. Surprisingly, the model shows comparably bad performance even for the token ranges from 28 and 31. Our analysis states that, for unseen tokens, the generation model also faces difficulties in capturing the exact semantic information.

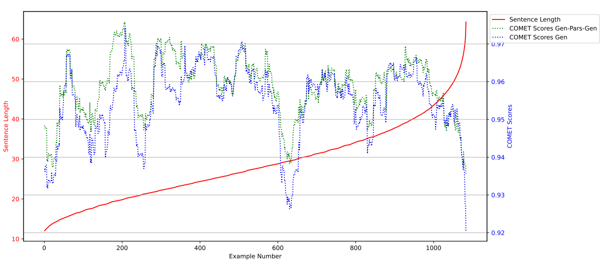

Figure 5: Sentence by sentence COMET score comparison of standalone Generator and Gen-Pars-Gen pipeline approaches.

Considering the question "Why does the pipeline work?", we provide here some speculations related to example 3 of Table 3. We note that the singular/plural feature is not explicitly denoted in the DRS, but it is only implicitly represented by the name "Jack". Moreover, we note that the only difference between the original input and the Gen-Pars output is the presence of the thematic role *USER* in contrast to *CREATOR*. Searching in the training set we found that the *USER* role has 729 instances while *CREATOR* has 220 instances. We can speculate that the standalone generator is not able to account for the standard singular form related to "Jack" since its original role, that is *CREATOR*, is not frequent in the training set. In contrast, the Gen-Pars-Gen system is able to realize the singular form of the verb since it has a more frequent semantic role, that is *USER*. In other words, we speculate that the role of the pipeline is to "correct" the input toward a more standard form, that is to transform the original input into a form closer to the instances that are in the training set.

5 Conclusion

In this study, we propose a novel approach that leverages LLMs in two different pipeline setups, Pars-Gen-Pars and Gen-Pars-Gen, to take advantage of the reversible nature of semantic parsing and text generation tasks for DRS. Firstly, we demonstrate how the reversible nature of these tasks can be effectively utilized to automatically correct errors in both semantic parsing and text generation, without the need for additional model training (RQ1, RQ2). Our Pars-Gen-Pars pipeline iteratively propagates the input text through parsing, generation, and parsing stages, while the Gen-Pars-Gen pipeline follows a similar process, starting with a DRS representation. Through comprehensive experiments on the PMB dataset, we show that our proposed pipelines consistently outperform the standalone parser and generator models across various evaluation metrics, including SMATCH for semantic parsing and BLEU, METEOR, COMET, chrF, and BERT-Score for text generation (RQ3). Our detailed error analysis categorizes the major types of errors made by the standalone models and demonstrates how the Pars-Gen-Pars pipeline effectively mitigates errors such as wrong WordNet sense assignments, missing logical concepts, hallucinated concepts, and incorrect index assignments in the parsing task (RQ4, RQ5). Similarly, the Gen-Pars-Gen pipeline addresses errors like grammatical mistakes, word position swapping, singular/plural inconsistencies, and altered textual representations in the text generation task (RQ4, RQ5).

Limitations: While our approach shows promising results, we acknowledge and analyze limitations related to the impact of sentence length, hallucination behavior, and out-of-vocabulary issues. These limitations highlight the need for continued research and advancements in LLMs, as well as the development of more sophisticated techniques to handle linguistic complexities effectively. Moreover, our experiments and evaluations were conducted solely on English data from the PMB dataset. We truly believe that the proposed pipeline approach holds potential for applicability to other languages, including low-resource languages, as well as multilingual settings.

While many errors are successfully reduced by our pipeline approaches, issues with sentence length, hallucinations, and unseen tokens remain. These highlight the need for more research and improvements in pre-trained language models, as well as the emergence of more advanced methods to deal with linguistic complexities.

Acknowledgments

We thank "ABC at XYZ" for providing support.

References

Lasha Abzianidze, Johannes Bjerva, Kilian Evang, Hessel Haagsma, Rik van Noord, Pierre Ludmann, Duc-Duy Nguyen, and Johan Bos. 2017. The Parallel Meaning Bank: Towards a multilingual corpus of translations annotated with compositional meaning representations. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 242–247, Valencia, Spain. Association for Computational Linguistics.

Muhammad Saad Amin, Luca Anselma, and Alessandro Mazzei. 2024. Exploring data augmentation in neural DRS-to-text generation. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2164–2178, St. Julian's, Malta. Association for Computational Linguistics.

Muhammad Saad Amin, Alessandro Mazzei, and Luca Anselma. 2022. Towards data augmentation for drs-to-text generation. In *Proceedings of the Sixth Workshop on Natural Language for Artificial Intelligence (NL4AI 2022) co-located with 21th International Conference of the Italian Association for Artificial Intelligence (AI*IA 2022), Udine, November 30th, 2022*, volume 3287 of *CEUR Workshop Proceedings*, pages 141–152. CEUR-WS.org.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. *in Proc.*, 7:178–186.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Valerio Basile and Johan Bos. 2011. Towards generating text from discourse representation structures. *in ENLG'*, 11:145–150.

Claire Bonial, William Corvey, Martha Palmer, Volha V Petukhova, and Harry Bunt. 2011. A hierarchical unification of lirics and verbnet semantic roles. In *2011 IEEE Fifth International Conference on Semantic Computing*, pages 483–489. IEEE.

Johan Bos. 2008. Wide-coverage semantic analysis with boxer. In *Semantics in text processing. step 2008 conference proceedings*, pages 277–286.

Johan Bos. 2021. Quantification annotation in discourse representation theory. In *ISA 2021-17th Workshop on Interoperable Semantic Annotation, Groningen/Virtuel, Netherlands, June*, pages 1–29.

Johan Bos. 2023. The sequence notation: Catching complex meanings in simple graphs. In *15th International Conference on Computational Semantics*, pages 195–208. Association for Computational Linguistics (ACL).

Shu Cai and Kevin Knight. 2013. Smatch: an evaluation metric for semantic feature structures. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 748–752.

Federico Fancellu, Sorcha Gilroy, Adam Lopez, and Mirella Lapata. 2019. Semantic graph parsing with recurrent neural network DAG grammars. 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 2769–2778, Hong Kong, China. Association for Computational Linguistics.

Christiane Fellbaum. 1998. *WordNet: An electronic lexical database*. MIT press.

Qiankun Fu, Yue Zhang, Jiangming Liu, and Meishan Zhang. 2020. DRTS parsing with structure-aware encoding and decoding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6818–6828, Online. Association for Computational Linguistics.

Michael Hanna and Ondřej Bojar. 2021. A fine-grained analysis of BERTScore. In *Proceedings of the Sixth Conference on Machine Translation*, pages 507–517, Online. Association for Computational Linguistics.

Kasia M Jaszczolt and Katarzyna Jaszczolt. 2023. *Semantics, pragmatics, philosophy: a journey through meaning*. Cambridge University Press.

Hans Kamp and Uwe Reyle. 1993. *From Discourse to Logic: Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*. Kluwer Academic Publishers, Dordrecht.

Hans Kamp and Uwe Reyle. 2013. *From discourse to logic: Introduction to modeltheoretic semantics of natural language, formal logic and discourse representation theory*, volume 42. Springer Science & Business Media.

Hans Kamp, Josef Van Genabith, and Uwe Reyle. 2010. Discourse representation theory. In *Handbook of Philosophical Logic: Volume 15*, pages 125–394. Springer.

Robert T Kasper. 1989. A flexible interface for linking applications to penman's sentence generator. In *Speech and Natural Language: Proceedings of a Workshop Held at Philadelphia, Pennsylvania, February 21-23, 1989*.

Jiangming Liu, Shay B. Cohen, and Mirella Lapata. 2021. Text generation from discourse representation structures. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 397–415, Online. Association for Computational Linguistics.

Rik van Noord. 2019. *Neural boxer at the IWCS shared task on DRS parsing*. in Proc. IWCS Shared Task on Semantic Parsing, Gothenburg, Sweden. Association for Computational Linguistics[.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Wessel Poelman, Rik van Noord, and Johan Bos. 2022a. Transparent semantic parsing with universal dependencies using graph transformations. In *29th International Conference on Computational Linguistics*, pages 4186–4192. Association for Computational Linguistics (ACL).

Wessel Poelman, Rik van Noord, and Johan Bos. 2022b. Transparent semantic parsing with Universal Dependencies using graph transformations. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4186–4192, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Inigo Jauregi Unanue, Gholamreza Haffari, and Massimo Piccardi. 2023. T3l: Translate-and-test transfer learning for cross-lingual text classification. *Transactions of the Association for Computational Linguistics*, 11:1147–1161.

Rik van Noord, Lasha Abzianidze, Hessel Haagsma, and Johan Bos. 2018. Evaluating scoped meaning representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Rik van Noord, Antonio Toral, and Johan Bos. 2019. Linguistic information in neural semantic parsing with multiple encoders. In *Proc. 13th International Conference on Computational Semantics-Short Papers*, pages 24–31. Association for Computational Linguistics (ACL).

Chunliu Wang, Huiyuan Lai, Malvina Nissim, and Johan Bos. 2023a. Pre-trained language-meaning models for multilingual parsing and generation. *Preprint*, arXiv:2306.00124.

Chunliu Wang, Rik van Noord, Arianna Bisazza, and Johan Bos. 2021a. Evaluating text generation from discourse representation structures. In *Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM 2021)*, pages 73–83, Online. Association for Computational Linguistics.

Chunliu Wang, Rik van Noord, Arianna Bisazza, and Johan Bos. 2021b. Input representations for parsing discourse representation structures: Comparing english with chinese. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pages 767–775. Association for Computational Linguistics (ACL).

Chunliu Wang, Xiao Zhang, and Johan Bos. 2023b. Discourse representation structure parsing for Chinese. In *Proceedings of the 4th Natural Logic Meets Machine Learning Workshop*, pages 62–74, Nancy, France. Association for Computational Linguistics.

Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. 2022. ByT5: Towards a token-free future with pre-trained byte-to-byte models. *Transactions of the Association for Computational Linguistics*, 10:291–306.

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.

Xiao Zhang, Chunliu Wang, Rik van Noord, and Johan Bos. 2024. Gaining more insight into neural semantic parsing with challenging benchmarks. *arXiv preprint arXiv:2404.08354*.

Appendix

In the appendix, we report the sentence-by-sentence scores of the text generation task using DRS to analyze the overall performance gain (see Appendix A.1).

A.1 Sentence-by-Sentence Evaluation of Parsing and Generation with and without Pipeline

Figure 6 depicts the relationship between sentence length and BLEU scores for the standalone generator model and the Gen-Pars-Gen pipeline approach. The x-axis represents the sentence length (in tokens), while the y-axis shows the BLEU scores. As observed in the figure, both the generator and Gen-Pars-Gen models exhibit a similar trend, where the BLEU scores vary with the sentence length. This trend can be attributed to the increased complexity and linguistic variations present in sentences, making it challenging for the models to generate accurate and fluent text. However, it is evident that the Gen-Pars-Gen pipeline consistently outperforms the standalone generator. This improvement in BLEU scores highlights the effectiveness of the proposed pipeline approach in mitigating errors and improving the quality of generated text, even for longer and more complex sentences.

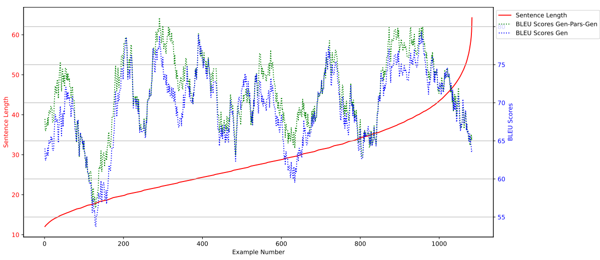

Figure 6: Sentence by sentence BLEU score comparison of standalone Generator and Gen-Pars-Gen pipeline approaches.

Figure 7 illustrates the relationship between sentence length and METEOR scores for the generator and Gen-Pars-Gen models. The x-axis represents the sentence length in tokens, while the y-axis shows the METEOR scores. Notably, the Gen-Pars-Gen pipeline consistently achieves higher METEOR scores compared to the standalone generator across various sentence length ranges. This improvement in METEOR scores suggests that the pipeline approach effectively mitigates errors and enhances the semantic similarity between the generated text and the reference, even for longer and more complex sentences. For very short sentences (less number of tokens in the text), the model hallucinates which can be seen from the lowest spike in the graph.

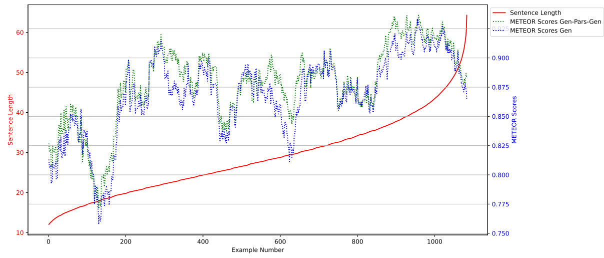

Figure 7: Sentence by sentence METEOR score comparison of standalone Generator and Gen-Pars-Gen pipeline approaches.

The chrF (character n-gram F-score) metric evaluates the quality of generated text by comparing character-level n-gram overlap between the generated text and the reference. In Figure 8, the Gen-Pars-Gen pipeline consistently achieves higher chrF scores compared to the standalone generator across various sentence variants. This improvement in chrF scores suggests that the pipeline approach effectively mitigates errors and enhances the character-level overlap between the generated text and the reference, even for longer and more complex sentences.

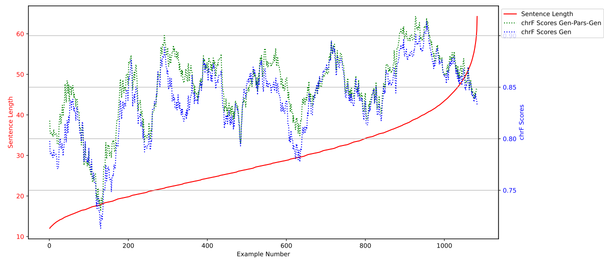

Figure 8: Sentence by sentence chrF score comparison of standalone Generator and Gen-Pars-Gen pipeline approaches.

Figure 9 depicts the relationship between sentence length and BERT-Score for the generator and Gen-Pars-Gen models. The x-axis represents the sentence length in tokens, while the y-axis shows the BERT-Score. As observed in the figure, both models exhibit a similar trend, where the BERT-Score shows variations as the sentence length changes. This trend can be attributed to the increased complexity and linguistic variations present in different sentences, making it challenging for the models to generate text that aligns well with the reference in terms of semantic similarity, as measured by the BERT-Score metric. However, the Gen-Pars-Gen pipeline consistently achieves higher BERT-Scores compared to the standalone generator across various sentence length ranges. This improvement in BERT-Score suggests that the pipeline approach effectively mitigates errors and enhances the semantic similarity between the generated text and the reference, even for longer and more complex sentences.

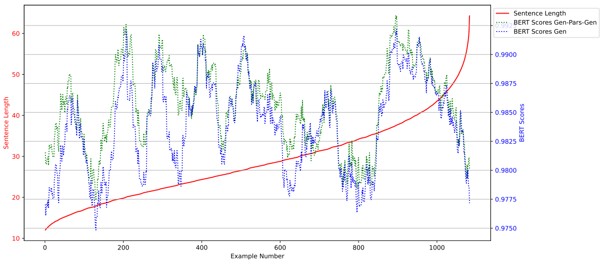

Figure 9: Sentence by sentence Bert Score comparison of standalone Generator and Gen-Pars-Gen pipeline approaches.