Exploiting Task Reversibility of DRS Parsing and Generation: Challenges and Insights from Multilingual Perspective

Muhammad Saad Amin, Luca Anselma and Alessandro Mazzei
Department of Computer Science, University of Turin, Italy
{muhammadsaad.amin, luca.anselma, alessandro.mazzei}@unito.it

Abstract

Semantic parsing and text generation exhibit reversible properties when utilizing Discourse Representation Structures (DRS). However, both processes—text-to-DRS parsing and DRS-to-text generation—are susceptible to errors. In this paper, we exploit the reversible nature of DRS to explore both error propagation, which is commonly seen in pipeline methods, and the less frequently studied potential for error correction. We investigate two pipeline approaches: Parse-Generate-Parse (PGP) and Generate-Parse-Generate (GPG), utilizing pre-trained language models where the output of one model becomes the input for the next. Our evaluation uses the Parallel Meaning Bank dataset, focusing on Urdu as a low-resource language, Italian as a mid-resource language, and English serving as a high-resource baseline. Our analysis highlights that, while pipelines are theoretically suited for error correction, they more often propagate errors, with Urdu exhibiting the greatest sensitivity, Italian showing a moderate effect, and English demonstrating the highest stability. This variation highlights the unique challenges faced by low-resource languages in semantic processing tasks. Further, our findings suggest that these pipeline methods support the development of more linguistically balanced datasets, enabling a comprehensive assessment across factors like sentence structure, length, type, polarity, and voice. Our cross-linguistic analysis provides valuable insights into the behavior of DRS processing in low-resource contexts, demonstrating both the potential and limitations of reversible pipeline approaches.

1 Introduction

DRS offers a distinct advantage in multilingual semantic processing through its language-neutral representation capabilities (Kamp and Reyle, 1993). This characteristic is particularly valuable for languages with limited computational resources. Derived from Discourse Representation Theory (DRT), DRS provides a comprehensive formal framework (Kamp et al., 2010) that captures complex linguistic phenomena including anaphors, presuppositions, temporal expressions, multisentence discourses, and the nuanced semantics of negation and quantification (Kamp and Reyle, 2013; Jaszczolt and Jaszczolt, 2023). This universal applicability makes DRS especially relevant for developing semantic processing capabilities across diverse linguistic contexts (Bos, 2021).

DRS applications span various NLP tasks, including machine translation (van Noord et al., 2018), semantic parsing (Noord, 2019; van Noord et al., 2019), and text generation (Wang et al., 2021; Amin et al., 2022; Liu et al., 2021; Amin et al., 2024). These tasks exhibit inherent reversibility—the output of one serving as input to the other—a property that holds particular promise for languages with limited NLP infrastructure. Traditional approaches, predominantly focused on English, require separate models for each task and language, creating significant barriers for languages with limited available data.

While pre-trained language models have transformed NLP capabilities, their impact on semantic parsing and text generation varies significantly across languages. The challenge is particularly evident in cases where explicit meaning representation is not inherently integrated into the training of these models (Amin et al., 2024). Despite recent advances, both DRS parsing and generation remain challenging (Wang et al., 2023), with parsing mistakes leading to incorrect meaning representations and generation errors resulting in disfluent text (Wang et al., 2021).

Our work introduces a novel pipeline approach leveraging the reversible nature of semantic parsing and text generation, focusing particularly on Urdu and Italian. Without requiring additional model training, we implement two pipeline setups using

Proceedings of the First Workshop on Language Models for Low-Resource Languages (LoResLM 2025), pages 268–286
January 20, 2025. ©2025 Association for Computational Linguistics

pre-trained language models: 1) Parse-Generate-Parse (PGP), where input text is parsed, used to generate text, and then parsed again; and 2) Generate-Parse-Generate (GPG), where a DRS is used to generate text, which is parsed and then used to regenerate text. We utilized the pipeline approaches (PGP and GPG) to examine three categories of examples: (i) those showing improved performance, indicating error correction or mitigation; (ii) those remaining unchanged, highlighting the deterministic behavior of neural models in DRS processing through pipelines; and (iii) those with decreased performance, signaling error amplification or propagation (see Table 2 for exact results).

We conduct our evaluation on the Parallel Meaning Bank[1] (PMB) dataset (Abzianidze et al., 2017), focusing specifically on Urdu as a low-resource language, Italian as a mid-resource language, and English as a high-resource baseline. The selection of these languages is based on their representation of distinct linguistic families, each characterized by unique syntactic structures, word-order variations, morphological complexity, and differing levels of resource availability. This diversity enables a comprehensive comparative analysis, offering valuable insights into how resource availability and linguistic characteristics influence the performance of DRS-based semantic processing across languages.

The research questions addressed in this paper are:

1. How does the reversible nature of semantic parsing and text generation with DRS affect error propagation and correction across different languages?

2. Can language models be effectively utilized in a pipeline approach to investigate error dynamics without additional model training?

3. What are the performance changes achieved by the proposed reversible pipelines compared to baseline models across different languages?

4. Which types of errors are more effectively addressed or amplified by the PGP and GPG pipelines in each language?

5. What are the capabilities and limitations of the reversible pipeline approaches in different linguistic contexts?

The key contributions of this paper are: (1) proposing a method for investigating error dynamics in DRS-based NLP tasks by exploiting reversibility, (2) demonstrating the varied effects of pipeline approach across multiple languages using pre-trained language models without costly retraining, and (3) analyzing the capabilities and limitations of the proposed pipelines through rigorous cross-linguistic error analysis. To the best of our knowledge, this study represents the first attempt to exploit the reversible nature of DRS parsing and generation to analyze error dynamics in a diverse multilingual context[2]. While previous research has primarily focused on either monolingual or multilingual semantic parsing and generation tasks, our work uniquely investigates the interplay between these tasks through their reversibility.

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 and pipeline configurations; Section 4 displays multilingual experimental results in detail; Section 5 presents a detailed error analysis with the discussion regarding the mitigation or amplification of errors; finally Section 6 concludes the paper, highlights limitations, and suggests directions for future research.

2 Background and Related Work

This section outlines the DRS formalism (§ 2.1) used in this study and reviews key research in semantic parsing (§ 2.2) and text generation (§ 2.3).

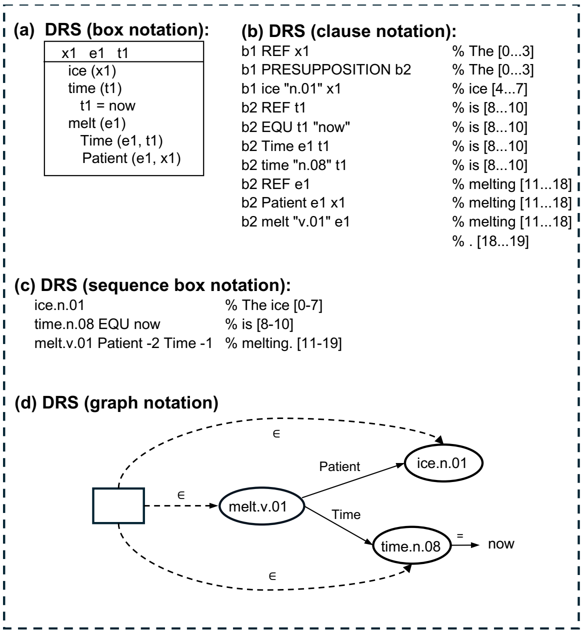
2.1 Discourse Representation Structure

As a formal meaning representation, DRS was developed to address semantic and pragmatic issues related to anaphora and tense (Kasper, 1989). It deals with a number of linguistic occurrences, such as temporal expressions and presuppositions (Bos, 2023). Unlike other formalisms used in large-scale semantic annotation initiatives, e.g., 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 dis-

[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] https://github.com/saadamin2k13/reversible-parsing-and-generation.



(a) DRS (box notation):

```
x1 e1 t1
-----------
ice (x1)
time (t1)
  t1 = now
melt (e1)
  Time (e1, t1)
  Patient (e1, x1)
```

(b) DRS (clause notation):

```
b1 REF x1                 % The [0...3]
b1 PRESUPPOSITION b2      % The [0...3]
b1 ice "n.01" x1          % ice [4...7]
b2 REF t1                 % is [8...10]
b2 EQU t1 "now"           % is [8...10]
b2 Time e1 t1             % is [8...10]
b2 time "n.08" t1         % is [8...10]
b2 REF e1                 % melting [11...18]
b2 Patient e1 x1          % melting [11...18]
b2 melt "v.01" e1         % melting [11...18]
                          % . [18...19]
```

(c) DRS (sequence box notation):

```
ice.n.01                          % The ice [0-7]
time.n.08 EQU now                 % is [8-10]
melt.v.01 Patient -2 Time -1      % melting. [11-19]
```

(d) DRS (graph notation)

Figure 1: Different graphical representations of DRS for the text “The ice is melting.” or (Urdu: “barf peghal rahi hay.”)

course 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, $\neg$). 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). In our experiments, we utilized the SBN representation (Figure 1(c)) and the directed acyclic graph (DAG) format (Figure 1(d)) for semantic processing tasks.

2.2 Semantic 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., 2022) 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., 2023; Amin et al., 2024). Beyond sequence-to-sequence models, two distinct research directions focus on tree-based approaches (Liu et al., 2021) and graph-based methods (Fancellu et al., 2019; Fu et al., 2020). Notably, Fu et al.’s (2020) marks the first effort toward multilingual DRS parsing.

2.3 Text Generation

While DRS parsing has long been a well-established area, NLP researchers have recently shifted their focus toward generating text from DRS (Basile and Bos, 2011; Wang et al., 2021; Amin et al., 2022; Wang et al., 2023; Amin et al., 2024). Similar to DRS parsing, past work on generating text from DRS has mainly fallen into two categories: rule-based methods (Basile and Bos, 2011) and neural network-based methods (Wang et al., 2021; Amin et al., 2022; Wang et al., 2023; Amin et al., 2024). 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., 2021; Amin et al., 2022, 2024). 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., 2023).

3 Methods

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., 2022). Instead, our work presents a pipeline approach that takes advantage of the complementary benefits offered by pre-trained language models. Our approach cascades these reversible processes into two different pipelines, PGP and GPG, so as to identify error mitigation or amplification that might occur in the generation or parsing phase, without requiring extra rule engineering or model training.

In our PGP and GPG pipelines, we employed byT5 (Xue et al., 2022) due to the following factors: (i) multilingual model can generalize better across languages and tasks; (ii) char-level/byte-level tokenization strategy helps the model understand complex language patterns, scripts, characters, and semantic information; (iii) when it comes to spelling and pronunciation-sensitive tasks, byte-level models outperform other models due to their greater resilience to noisy data; (iv) byT5 is also referred to as a token-free model as it operates directly on raw UTF-8 data without generating sub-word or word-based vocabulary; and (v) most importantly, byT5 has the state-of-the-art results on multilingual NLP benchmarks outperforming other models (Xue et al., 2022; Stankevičius et al., 2022; Belouadi and Eger, 2023).

PMB is a multilingual dataset comprising semantic representations in English, Italian, German, Dutch, and Chinese. Leveraging the language-neutral nature of DRS, we transformed English DRS-Text pairs into Urdu through a systematic approach involving syntactic structure, concept and word alignment, grammatical genders, and cross-lingual adaptation through named entities. This hybrid methodology resulted in the first comprehensive semantic resource for Urdu[3], comprising 3,000 manually annotated data instances. DRS transformations were achieved through rule-based techniques and human annotation. Text translations were initially generated using the Google Translate API and subsequently verified through manual inspection. Urdu examples were divided into 1,200 training, 900 development, and 900 test examples. For Italian, the dataset consisted of 5,061 training examples, 555 development examples, and 555 test examples. For English, the dataset contained 152,808 training examples, 1,132 development examples, and 1,132 test examples.

To enhance dataset diversity and complexity, we applied multi-dimensional augmentation strategies, including named entities, lexical (encompassing common nouns, adjectives, adverbs, and verbs), and grammatical augmentations. This approach resulted in a ninefold increase in the training data examples applied to all three languages, i.e., EN, IT, and UR. For experimentation, we fine-tuned byT5 on our fully augmented DRS-Text pairs, achieving state-of-the-art performance in both semantic parsing and text generation tasks[4]. We implemented a two-stage fine-tuning strategy consistent with (Zhang et al., 2024). The first stage involved fine-tuning the model on silver data for 3 epochs to establish foundational DRS knowledge. The second stage focused on gold data fine-tuning for 10 epochs. Experimental parameters included AdamW optimizer, polynomial learning rate decay ($1e-4$), batch size of 32, maximum sequence length of 512, and GeGLU activation function. To evaluate the impact of the pipeline approach, we utilized SMATCH for semantic parsing (Cai and Knight, 2013), while BLEU (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005), ROUGE (Lin, 2004), COMET (Rei et al., 2020), chrF (Popović, 2015), and BERTScore (Hanna and Bojar, 2021) were applied to assess text generation outcomes.

3.1 PGP

The PGP pipeline is designed to identify error dynamics—mitigation or amplification—in the se-

[3] Urdu PMB is not part of the official website yet, but can be provided freely for scientific purposes.

[4] All six models, encompassing three languages (EN, IT, UR) and two tasks (parsing and generation), are available at https://huggingface.co/saadamin2k13

Experimentation	Language	S-Parsing	Generation Results					
Type	Type	S-F1	BLEU	MET.	CMT.	chrF	B_Scr.	ROUGE
without pipeline	EN	93.56	71.01	87.67	95.81	84.97	98.54	–
with pipeline		93.06	69.25	86.73	95.33	83.77	98.35	–
without pipeline	IT	90.56	56.76	72.67	89.97	70.59	92.85	–
with pipeline		89.19	53.06	69.68	88.53	67.54	91.88	–
without pipeline	UR	79.77	53.31	53.07	–	51.49	88.33	59.40
with pipeline		76.42	48.72	45.98	–	44.87	86.27	53.07

Table 1: Experimental results of parsing and generation with and without pipeline approach on standard test sets for English, Italian, and Urdu. The best results are underlined. Note: S-Parsing = Semantic Parsing; S-F1 = SMATCH F1-Score; MET. = METEOR; CMT. = COMET; B_Scr. = BERT-Score.

mantic 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 PGP pipeline.

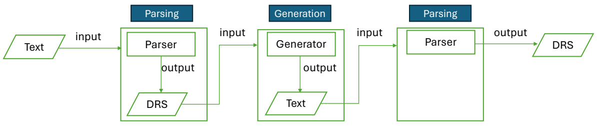


Figure 2: Graphical representation of PGP pipeline.

3.2 GPG

Similarly, the GPG pipeline is designed to identify error dynamics 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 GPG pipeline is shown in Figure 3.

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

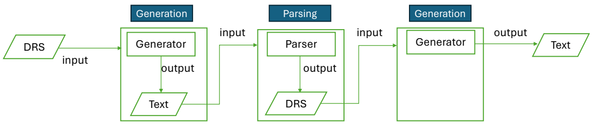


Figure 3: Graphical representation of GPG pipeline.

4 Results

We experimented with three distinct languages—Urdu (UR), Italian (IT), and English (EN)—using the standard test set from the dataset. The results reveal complex patterns of performance changes across languages and metrics as shown in Table 1.

4.1 PGP Evaluation

The PGP pipeline was evaluated using SMATCH, an overlap-based metric typically used in AMR parsing (Cai and Knight, 2013), which computes the F1-score of matched triples between system-generated and gold standard DRS representations. The results in Table 1 indicate that the PGP pipeline generally retains parsing accuracy across multiple languages, but with variations depending on language complexity.

For English, the pipeline performed deterministically, with only a marginal decrease in SMATCH F1-score from 93.56 to 93.06, a mere 0.5% decrease. This demonstrates that the pipeline introduces minimal errors, making it highly efficient for semantic parsing tasks in a rich-resourced language i.e., English. For Italian, a slight decrease in the F1-score (from 90.56 to 89.19) was observed, representing a 1.37% decrease. While Italian's more complex sentence structure and grammar present challenges, the PGP pipeline still performs admirably, showing promise for further language-specific improvements. In Urdu, the F1-score decreased more noticeably, from 79.77 to 76.42

Language	Imp. Type	Ex. Testset	Ex. Improved	Ex. Same	Ex. Decreased
English	Parsing	1132	49 (+4.33%)	975 (86.13%)	108 (-9.54%)
	Generation		35 (+3.09%)	1015 (89.66%)	82 (-7.24%)
Italian	Parsing	555	29 (+5.23%)	446 (80.36%)	80 (-14.41%)
	Generation		24 (+4.32%)	438 (78.92%)	93 (-16.76%)
Urdu	Parsing	900	114 (+12.66%)	449 (49.88%)	337 (-37.44%)
	Generation		114 (+12.66%)	401 (44.55%)	385 (-42.77%)

Table 2: Performance metrics of multilingual semantic parsing and generation indicating the total number of examples, with the number and percentage of improved, same, and decreased categories.

(a 3.35% drop), reflecting the greater challenges posed by its rich morphology and syntax. Despite these challenges, the pipeline holds potential even without extensive pre-training or fine-tuning, suggesting that further adaptation could yield improved results for morphologically complex languages.

The parsing performance breakdown (see Table 2) further highlights language-specific trends. For English, out of 1132 examples, 49 (4.33%) improved, 975 (86.13%) remained the same, and 108 (9.54%) showed decreased performance. Italian demonstrated similar trends with 29 (5.23%) improvements, 446 (80.36%) unchanged examples, and 80 (14.41%) showing decreased performance out of 555 examples. Urdu, however, showed the most variability, with 114 (12.66%) examples showing improvement, 449 (49.88%) remaining the same, and a notable 337 (37.44%) showing decreased performance out of 900 examples.

4.2 GPG Evaluation

For the GPG pipeline, we evaluated text generation performance using both rule-based BLEU, METEOR, chrF, ROUGE, neural model-based COMET and pre-trained model-based BERT-Score metrics to assess the quality of generated text compared to reference text across English, Italian, and Urdu. COMET was not used for Urdu due to lack of specific evaluation datasets, and ROUGE was excluded for English and Italian as it is not ideal for evaluating text generation in rich-resource and mid-resource languages. Table 1 lists multilingual text generation results across different evaluation measures. The GPG pipeline maintains strong performance, especially for English text generation, with only minor declines across BLEU (71.01 to 69.25), METEOR (87.67 to 86.73), and chrF (84.97 to 83.77), indicating that the generated text remains highly comparable to the original output.

For Italian, although there was a slight decrease in BLEU (56.76 to 53.06), METEOR (72.67 to 69.68), and chrF (70.59 to 67.54), the GPG pipeline still performed commendably, demonstrating its capability to handle more linguistically diverse languages. In Urdu, despite its morphological complexity, the pipeline still captures the essence of sentence structure. However, larger declines in BLEU (55.31 to 48.72), METEOR (53.07 to 45.98), chrF (51.49 to 44.87), and ROUGE (59.40 to 53.07) indicate the need for further optimization in handling morphologically rich languages like Urdu.

The generation performance breakdown (see Table 2) complements these metric-based results. For English, 35 (3.09%) out of 1132 examples showed improvement, 1015 (89.66%) remained unchanged, and 82 (7.24%) showed decreased performance. In Italian, 24 (4.32%) out of 555 examples showed improvement, 438 (78.92%) remained the same, and 93 (16.76%) showed decreased performance. Urdu displayed the most variation, with 114 (12.66%) examples showing improvement, 401 (44.55%) remaining unchanged, and 385 (42.77%) showing decreased performance out of 900 examples.

In the broad spectrum of evaluation, both the PGP and GPG pipelines demonstrate potential for handling multilingual semantic parsing and text generation tasks. For English, the pipelines preserve much of the original performance with only minor fluctuations, underscoring their robustness. Even for Italian and Urdu, where challenges due to linguistic complexity are more pronounced, the pipelines provide a strong foundation for further improvements. The decrease in performance, particularly for Italian and Urdu, underscores areas for improvement but is balanced by the pipelines' overall effectiveness in multilingual contexts.

The results indicate that with minimal language-specific adaptations, especially for Urdu, the pipeline is capable of generating high-quality results. These experiments pave the way for further exploration into how reversible semantic parsing and text generation can be leveraged to enhance semantic processing in a multilingual context.

5 Analysis and Discussion

To understand why PGP and GPG pipeline approaches often result in error amplification rather than mitigation, we conducted a systematic analysis focusing on the impact of linguistic imbalance in the dataset (§ 5.1), error patterns in pipeline approaches (§ 5.2), performance impact through cross-lingual analysis (§ 5.3), and revealing the pipeline approach (§ 5.4).

5.1 Linguistic Imbalance in the Dataset

For linguistic imbalance, we conducted analysis across five linguistic dimensions: sentence length (*Short, Medium, Long*), sentence types (*Declarative, Exclamatory, Imperative, Interrogative*), structural complexity (*Simple, Complex, Compound, Compound-Complex*), polarity (*Affirmative, Negative*), and voice (*Active, Passive*). This multifaceted and multilingual analysis aims to identify specific linguistic phenomena that may contribute to pipeline performance degradation.

5.1.1 Sentence Length

In our analysis, English training data is biased towards longer sentences, while the test set favors medium-length sentences, contributing to performance degradation in short and medium categories. Italian shows a similar trend, with the test set dominated by medium and short sentences, creating challenges in handling complex, long sentences. Conversely, Urdu exhibits consistent medium-length sentence representation but suffers greater performance decline due to linguistic complexities such as SOV word order and morphology. This disparity across languages and sentence lengths suggests that each language's unique structural properties, combined with length mismatches, significantly impact pipeline performance (see Appendix C.1 with Table 5 for sentence splits and Table 6 for results).

5.1.2 Sentence Type

English training data is heavily skewed toward declarative sentences, while the test set has a more balanced representation of declarative and interrogative sentences. This distribution shift impacts pipeline performance, particularly in declarative. Italian maintains stable declarative representation between training and test sets but still experiences significant performance degradation, especially with interrogative sentences. Urdu also has a high proportion of declarative sentences in both training and test sets but suffers the most severe performance drops across types, particularly for imperative sentences. Appendix C.2 explains in detail the sentence type imbalance (see Table 7) and results (see Table 8). These findings suggest language-specific distribution imbalances contribute to pipeline performance inconsistencies.

5.1.3 Structural Complexity

The analysis shows that Urdu and Italian data are heavily skewed towards simple sentence structures, with simple sentences comprising over 88% in both training and test sets. English data, while still dominated by simple sentences, has a more balanced distribution with greater representation of complex and compound structures in the training set. This imbalance across sentence structures results in a general performance decline for all languages as structural complexity increases, with the pipeline approach showing some advantage in handling compound sentences in Italian but lagging for complex structures in English and Urdu. These findings highlight the need for language-specific strategies to address structural complexity. We have listed a detailed analysis in Table 9 and Table 10 (see Appendix C.3).

5.1.4 Polarity

The analysis of sentence polarity reveals that English and Urdu exhibit a strong bias toward affirmative sentences, with English showing 84.73% and Urdu 88.09% affirmative sentences in the training set. In contrast, Italian is predominantly biased towards negative sentences, constituting 60.80% in the training set. This pattern persists in the test sets, with English (91.34%) and Urdu (90.00%) maintaining high percentages of affirmatives, while Italian continues to favor negatives (63.42%)—see Table 11 and Table 12 in Appendix C.4. Despite these biases, the non-pipeline approach generally outperforms across all languages, suggesting robust processing capabilities across both affirmative and negative sentence types.

5.1.5 Voice

The analysis of sentence voice reveals a strong bias toward active voice across English, Italian, and Urdu (see Table 13 and Table 14 in Appendix C.5).

In the training data, active voice dominates with 90.58% in English, 92.06% in Italian, and 92.01% in Urdu. This trend continues in the test sets, where active voice sentences increase to 93.37%, 94.05%, and 93.78%, respectively. Notably, while both English and Italian demonstrate higher SMATCH scores for passive voice sentences despite their lower frequency, Urdu exhibits a consistent challenge in processing passive constructions compared to active ones. This suggests that, while active voice is favored across languages, the performance dynamics vary significantly, especially in Urdu.

5.2 Error Patterns in Pipeline Processing

The PGP and GPG pipeline approaches exhibit complex error dynamics that warrant detailed analysis. Our investigation focuses on examining specific types of errors that emerge and propagate through the pipeline stages. This analysis reveals systematic patterns in how errors evolve and amplify, providing insights into the limitations of pipeline processing for semantic parsing and generation tasks.

5.2.1 Semantic Parsing Errors

In the PGP pipeline, four key errors significantly impact processing accuracy. Table 3 in Appendix A reports these errors in detail.

Erroneous WordNet Sense Assignment occurs when the parser initially assigns the wrong sense to a word (`fly.v.01` vs. `fly.v.05`), as seen in the sentence "Let's fly a kite," leading to a cascade of incorrect interpretations through the pipeline stages.

Omission of Logical Concepts is another critical failure, illustrated by questions like "Is your father Spanish?" where the parser may neglect essential logical elements e.g., `time.n.08 EQU now`, resulting in a distorted semantic representation as the pipeline progresses.

Additionally, the *Generation of Incorrect Thematic Roles* manifests in examples like "I caught a fish!" where initial role assignments, such as `Agent/Recipient` and `Experiencer`, can deteriorate, creating complex misassignments that deviate from the original meaning.

Lastly, *Erroneous Index Assignment* occurs when the numeric indices that link logical concepts are incorrectly applied, as in the example "Mayuko designed a dress for herself." *Indices* are used to connect concepts, with positive indices pointing to subsequent logical concepts (`Beneficiary +1`) and negative indices indicating references to previously discussed concepts (`Agent -1`). These indices are crucial for determining word order and maintaining coreference relationships. When index errors occur, they disrupt the intended referential structure, leading to incoherent DRS representations that fail to capture the correct coreference and syntactic relationships, thus affecting the overall interpretation and meaning of the text.

5.2.2 Text Generation Errors

In the GPG pipeline, the most significant errors that disrupt the coherence and accuracy of generated outputs are mentioned in Table 4 in Appendix B. The major issues correspond to:

Grammatical Inaccuracies, that are evident in DRS representations like "high.a.02 Value ? AttributeOf +1 mountain.n.01 Name 'Mount Kinabalu,"' where initial grammatical mistakes (e.g., "How high of Mount Kinabalu?") can lead to severe semantic distortions in later stages.

Word Position Misalignment is another critical issue, as seen in cases like "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," where incorrect word order (e.g., "Who founded the striptease club Chippendale?") complicates the reconstruction of logical relationships in subsequent parsing.

Singular-Plural Discrepancies emerge when generating sentences such as "Jack's book is interesting," which may incorrectly transform into "Jack's books are interesting," affecting logical relationships and leading to deeper semantic inconsistencies.

Lastly, *Textual Representation Variations* can cause unexpected semantic divergences, as demonstrated by changes like representing "100" as "a hundred," which may trigger parsing errors due to differing interpretations of paraphrased expressions. These errors highlight how linguistic nuances can propagate through the pipeline, undermining the integrity of the generated text.

5.3 Cross-Lingual Analysis

The cross-linguistic analysis highlights distinct error patterns in semantic processing influenced by the structural characteristics of different languages. In English, errors primarily arise from sense assignments and logical concept handling in parsing, along with grammatical and word order issues in generation tasks. Italian's richer morphology leads to complex challenges, particularly in thematic role assignments during parsing and number agreement in generation. Urdu, characterized by word order and complex morphology, exhibits the most severe degradation across all categories, struggling to maintain flexibility and linguistic agreements. The analysis indicates that errors introduced at each stage of the pipeline tend to amplify rather than correct, resulting in a cyclical pattern of semantic drift that degrades output quality. This suggests a need for robust standalone models that can effectively handle complex semantic representations without relying on multiple transformation stages, thereby maintaining fidelity to language-specific features.

5.4 Revealing the Pipeline Approach

The analysis highlights significant shortcomings of the pipeline approach in semantic processing across languages. Error amplification was a major issue, with English maintaining stable parsing accuracy (93.56% to 93.06%), while Italian and Urdu experienced more substantial drops (90.56% to 89.19% for Italian and 79.77% to 76.42% for Urdu). Similarly, in the GPG pipeline, English BLEU scores decreased slightly, but Italian and Urdu showed larger declines, reflecting greater error accumulation in complex morphological contexts.

The linguistic complexity of Italian and Urdu exacerbated the pipeline's performance, as only 80.36% of Italian examples and a mere 49.88% of Urdu examples maintained parsing stability, compared to 86.13% for English. Furthermore, semantic drift occurred as outputs diverged from their intended meanings; parsing errors in sentences led to cascading inaccuracies, with Urdu's SMATCH score dropping from 81.32% to 77.40% for longer sentences.

The mismatch between surface forms and semantic content was evident, with Italian and Urdu experiencing significant declines in BLEU and METEOR scores during generation tasks. Additionally, the pipeline struggled with linguistic ambiguity, particularly in Urdu, where over 42.77% of examples exhibited performance declines due to polysemy. Finally, the inability to correct logical and thematic role errors compounded inconsistencies, with Urdu's SMATCH score dropping from 79.77% to 76.42%, underscoring critical weaknesses in maintaining logical coherence throughout the semantic processing chain.

Considering the question "When and Why does the pipeline work?", we provide here some speculations related to Example 3 of Table 4. 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.

6 Conclusion

We investigated the reversible nature of semantic parsing and text generation through DRS, leveraging pipeline approaches across Urdu, Italian, and English. The primary objective was to assess the impact of two distinct pipeline configurations (PGP, GPG) on error propagation or mitigation without additional model training. By employing pretrained language models, we explored how these reversible processes influence the performance of both parsing and generation tasks, providing valuable insights into cross-linguistic error dynamics. The key findings demonstrate that, while the reversible pipeline approach offers the potential for correcting errors, it more frequently leads to error amplification, particularly in languages with complex morphology and syntactic structures, such as Urdu and Italian. English showed the most stability, with only slight performance drops in parsing and generation tasks. In contrast, Urdu and Italian were more prone to error amplifications, as errors introduced in one stage of the pipeline tended to grow in later stages. Through a detailed analysis of error patterns across different linguistic dimensions, we provide an in-depth understanding of how specific language characteristics influence error propagation. We revealed that the reversible nature of DRS-based pipelines, while theoretically promising, is limited in practical effectiveness due to the compounding of errors in complex sentence structures and morphologically rich languages.

Limitations: The potential of our PGP and GPG pipelines to exploit the task reversibility of DRS offers opportunities for effective error dynamics, whether through propagation or mitigation. However, the predominance of error propagation over error mitigation is attributed to the dependency of these pipeline approaches on pre-trained language models. In our experimental implementation, we utilized the best-performing models with state-of-the-art results for the languages involved. Yet, the data examples used to train the English DRS processing models vastly outnumbered those for Italian and Urdu, posing a challenge in terms of model generalization and robustness capabilities. Furthermore, the limitations of traditional evaluation metrics, such as SMATCH (which only considers structural overlap) and BLEU and METEOR (which are based on n-gram overlap), further complicate the assessment of these results. In our analysis, we resorted to human evaluation, which is computationally expensive and time-consuming. Additionally, our analysis has highlighted the linguistic imbalance across the various DRS variants, which also poses a limitation to the fair evaluation of the models. These findings suggest the need for a more balanced dataset to train models that can overcome these limitations and deliver the best possible results.

Acknowledgments

The research is conducted at the Department of Computer Science, University of Turin, Italy, and is partially funded by the "HARMONIA" project (M4-C2, I1.3 Partenariati Estesi - Cascade Call - FAIR). The project is supported under the NextGenerationEU program, with the funding identification details CUP C63C22000770006 - PE PE0000013. We extend our gratitude to prof. Viviana Patti, the principal investigator of the HARMONIA research initiative, for facilitating the funding of this work.

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.

Jonas Belouadi and Steffen Eger. 2023. ByGPT5: End-to-end style-conditioned poetry generation with token-free language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7364–7381, Toronto, Canada. Association for Computational Linguistics.

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*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

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. 2022. Transparent semantic parsing with universal dependencies using graph transformations. In *29th International Conference on Computational Linguistics*, pages 4186–4192. Association for Computational Linguistics (ACL).

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.

Lukas Stankevičius, Mantas Lukoševičius, Jurgita Kapočiūtė-Dzikienė, Monika Briedienė, and Tomas Krilavičius. 2022. Correcting diacritics and typos with a byt5 transformer model. *Applied Sciences*, 12(5):2636.

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. 2023. Pre-trained language-meaning models for multilingual parsing and generation. *Preprint*, arXiv:2306.00124.

Chunliu Wang, Rik van Noord, Arianna Bisazza, and Johan Bos. 2021. 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.

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.

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*.

A Analyzing Error Dynamics for Semantic Parsing

Table 3 lists error dynamics regarding the PGP pipeline. In the first column, we have the Gold Text which is parsed to get the corresponding DRS representations i.e., Pars (DRS). This Pars (DRS) is used to generate textual representation—Pars-Gen (Text). Moreover, this textual representation is passed to a semantic parser to generate Pars-Gen-Pars (DRS) that is used to analyze the potential error dynamics in the PGP processing.

B Analyzing Error Dynamics for Text Generation

Table 4 lists error dynamics regarding the GPG pipeline. In the first column, we have the Gold DRS which is generated to get the corresponding textual representations of the DRS i.e., Gen (DRS). This text is parsed to extract its logical representation—DRS equivalence of the generated text which is passed to a generator to analyze the potential error dynamics in the GPG processing.

C Linguistic Distributional Imbalance in the Test Set

C.1 Impact of Sentence Length

To analyze the impact of sentence length on pipeline performance, we categorized sentences into three classes based on token count: short (0-4 tokens), medium (5-8 tokens), and long (9+ tokens). For token classification, we have adopted a rule-based custom tokenization strategy to split the sentences. Our analysis reveals significant distributional disparities between training and test sets across all three languages, which partially explains the suboptimal performance of our pipeline approaches. Table 5 shows the sentence splits corresponding to different sentence lengths based on tokens/words per sentence.

In **English**, while the training data shows a natural distribution skewed towards longer sentences (51.72% long, 44.48% medium, 3.81% short), the test set exhibits a markedly different distribution with a strong bias towards medium-length sentences (69.70%) and notably higher representation of short sentences (14.75%). This distributional mismatch appears to impact pipeline effectiveness, as evidenced by consistent performance degradation across all metrics and length categories. The impact is particularly pronounced in short sentences, where the SMATCH score drops from 90.89 to 89.69, suggesting that the pipeline struggles with concise expressions where each token carries significant semantic weight.

Italian displays an even more pronounced distributional shift between training and test sets. The test data is heavily concentrated in the medium-length category (70.27%) with a notable overrepresentation of short sentences (25.77%) compared to training. This imbalance appears to particularly affect the pipeline's performance on long sentences, where we observe the most substantial degradation across metrics (e.g., BLEU score drops from 47.98 to 41.68). The scarcity of long sentences in the test set (3.96%) compared to training (25.01%) suggests that the model may not have developed robust handling of complex, lengthy expressions.

Urdu presents the most concerning performance degradation among the three languages, with substantial drops across all metrics and length categories. The medium-length sentences, despite being the most represented in both training (68.12%) and test (66.78%) sets, show a significant performance decline in pipeline processing (SMATCH drops from 81.32 to 77.40). This suggests that beyond distributional mismatches, structural characteristics of Urdu, such as its SOV word order and complex morphology, may be amplifying errors through the pipeline stages.

A cross-linguistic analysis reveals that medium-length sentences consistently achieve the best baseline performance across all three languages, but also suffer from notable degradation in pipeline processing. This pattern suggests that while these sentences contain enough information for robust semantic parsing, the pipeline's sequential nature introduces compounding errors that overwhelm any potential error correction benefits. The performance degradation is most pronounced in metrics that evaluate structural similarity and semantic accuracy (SMATCH, METEOR) rather than surface-

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.01 Quantity + 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 myself caught a fish.	person.n.01 EQU speaker catch.v.08 Recipient Experiencer Of -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 on time for herself.	female.n.02 Name "Mayuko" design.v.03 Agent -1 Time +1 Result +2 Time +2 Beneficiary +1 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 3: Analyzing error patterns through the lens of semantic parsing.

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"	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 found the striptease club Chippendale 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 his 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 is the square root value of the number 100?	What's the square root of 100?

Table 4: Analyzing error patterns through the lens of text generation.

Lang.	**Data Type**	**Total Ex.**	**Sentence Splits (words/tokens)**		
			Short (%) (0–4)	**Medium (%)** (5–8)	**Long (%)** (9–)
English	Train	152788	3.81	44.48	51.72
	Test	1132	14.75	69.70	15.55
Italian	Train	5061	13.50	61.49	25.01
	Test	555	25.77	70.27	3.96
Urdu	Train	9057	13.07	68.12	18.80
	Test	900	18.33	66.78	14.89

Table 5: Sentence length distribution by language and data type.

level similarity (BLEU), indicating that the pipeline is particularly vulnerable to semantic drift during multiple transformation steps.

These findings suggest that the underperformance of pipeline stems from a combination of factors: (1) distributional mismatches between training and test sets across sentence lengths, (2) language-specific structural characteristics that amplify errors through multiple transformations, and (3) the inherent challenge of maintaining semantic consistency through sequential processing steps. The consistent degradation across all metrics and languages indicates that our current pipeline architecture may need fundamental modifications to achieve effective error mitigation. Table 6 lists multilingual results with the utilization of the impact of sentence length on the performance of pipeline approaches.

C.2 Performance Impact on Sentence Types

A systematic analysis of sentence type distributions reveals significant disparities between training and test sets across English, Italian, and Urdu. This imbalance manifests distinctly in each language, affecting the pipeline's error mitigation capabilities in different ways. Table 7 lists 4 different types of sentences present in the English, Italian, and Urdu data examples. We have used spaCy to extract these sentence types from the dataset.

In **English**, the training data is heavily dominated by declarative sentences (86.76%), while the test set shows a more balanced distribution with declarative sentences comprising 61.31%. This imbalance is further highlighted in interrogative sentences, where the test set proportion (31.63%) significantly exceeds the training representation (8.91%). The impact of this disparity is evident

Lang.	Imp. Type (ex.)	Pipeline	SMATCH (F1)	BLEU	METEOR	COMET	ROUGE	chrF	BERT_Score
EN	Short (167)	Without	**90.89**	**71.04**	**84.13**	**95.47**	–	**82.25**	**98.46**
		With	89.69	68.17	83.04	94.43	–	80.50	98.06
	Medium (789)	Without	**94.85**	**71.89**	**88.66**	**96.34**	–	**85.85**	**98.65**
		With	94.47	70.29	87.78	95.99	–	84.77	98.52
	Long (176)	Without	**90.32**	**66.99**	**86.62**	**93.70**	–	**83.61**	**98.09**
		With	89.89	65.35	85.54	93.25	–	82.39	97.91
IT	Short (143)	Without	**90.52**	**53.61**	**63.14**	**87.62**	–	**63.44**	**90.27**
		With	89.48	49.32	60.15	85.65	–	59.89	89.14
	Medium (390)	Without	**90.66**	**58.41**	**76.22**	**90.98**	–	**73.29**	**93.79**
		With	89.14	55.07	73.39	89.86	–	70.55	92.95
	Long (22)	Without	**89.22**	**47.98**	**71.58**	**87.02**	–	**69.23**	**92.90**
		With	88.21	41.68	65.73	83.56	–	61.56	90.91
UR	Short (165)	Without	**79.14**	**52.17**	**49.20**	–	**56.90**	**49.60**	**87.43**
		With	76.46	44.86	40.93	–	49.17	41.59	85.34
	Medium (601)	Without	**81.32**	**57.38**	**55.29**	–	**60.99**	**52.97**	**88.87**
		With	77.40	51.35	48.97	–	55.49	47.20	87.07
	Long (134)	Without	**73.06**	**49.91**	**47.88**	–	**55.36**	**47.20**	**86.97**
		With	71.96	41.63	38.78	–	47.00	38.47	83.82

Table 6: Impact of sentence length on evaluation results with and without pipeline for EN, IT, and UR. Bold indicates the better results.

Sentence Type	EN		IT		UR	
	Train (%)	Test (%)	Train (%)	Test (%)	Train (%)	Test (%)
Declarative	86.76	61.31	87.39	87.57	93.82	87.22
Exclamatory	2.26	6.27	1.90	2.52	0.71	3.00
Imperative	2.06	0.80	0.57	0.18	0.76	0.89
Interrogative	8.91	31.63	10.14	9.73	4.71	8.89

Table 7: Sentence structure type distribution in training and test sets (EN, IT, UR).

in the pipeline's performance: declarative sentences show performance degradation from baseline SMATCH of 93.44% to 92.98% with the pipeline. Interrogative sentences, despite their underrepresentation in training, maintain relatively robust performance with a modest SMATCH decline from 93.94% to 93.37%. Notably, exclamatory sentences, though comprising only 2.26% of training data, achieve the highest baseline SMATCH score (94.97%) but still experience degradation through the pipeline (94.23%).

Italian demonstrates a more stable distribution of declarative sentences between training (87.39%) and test (87.57%) sets, yet the pipeline still shows consistent performance degradation. The baseline SMATCH score for declarative sentences (90.91%) drops to 89.45% with the pipeline approach. Interrogative sentences, representing 10.14% of training and 9.73% of test data, show a significant performance decline across all metrics when processed through the pipeline, with SMATCH dropping from 87.22% to 86.97% and more dramatic drops in BLEU (53.41% to 44.15%) and METEOR (67.04% to 60.41%). Exclamatory sentences, despite limited representation, show notable baseline performance (91.92% SMATCH) but experience substantial degradation through the pipeline (88.66%).

Urdu exhibits the most pronounced training-test distribution stability for declarative sentences (93.82% training, 87.22% test) but shows the most severe pipeline performance degradation. Declarative sentences suffer a significant SMATCH drop from 79.72% to 76.25%. Interrogative sentences, despite having lower representation in both training (4.71%) and test (8.89%) sets, achieve the highest baseline performance among all Urdu sentence types (83.90% SMATCH) but still deteriorate with pipeline processing (81.02%). Imperative sentences, with minimal representation in both sets, show the most dramatic performance decline, with SMATCH dropping from 72.06% to 62.25% and substantial degradation across all other metrics.

The analysis reveals a consistent pattern of pipeline performance degradation across all three languages, though with varying severity. English shows the most resilient performance with relatively modest degradation across sentence types. Italian demonstrates moderate performance drops, particularly pronounced in semantic metrics. Urdu exhibits the most severe degradation, suggesting that language-specific structural characteristics may amplify the challenges posed by distributional imbalances. This cross-linguistic comparison indicates that the pipeline's error amplification tendency is influenced both by training-test distribution mismatches and by inherent linguistic complexities specific to each language. Table 8 lists multilingual results with the utilization of the impact of sentence types on the performance of pipeline approaches.

C.3 Analysis based on Structural Complexity

The distribution analysis based on structural complexity reveals significant imbalances across different sentence types in both training and test sets. In the training data, simple sentences dominate across all three languages, with English showing the most balanced distribution (70.18% simple, 14.30% complex, 9.40% compound, and 6.12% compound-complex). Italian and Urdu display an even stronger bias toward simple sentences (88.05% and 93.31% respectively), with minimal representation of other structures. This imbalance becomes even more pronounced in the test sets, where simple sentences constitute approximately over 94% of the data across all languages, and compound-complex sentences are entirely absent. We have used spaCy to classify sentences based on structural complexity from the dataset. Table 9 shows the percentage-wise structural distribution of sentences in the training and test sets for EN, IT, and UR.

For **English** language performance, the results present interesting variations across different sentence types. In simple sentences, which comprise the majority of the test set (1079 examples), the non-pipeline approach generally outperforms, achieving higher scores across most metrics (SMATCH: 93.79%, BLEU: 71.18%, METEOR: 87.63%). However, the pipeline approach shows promising results in complex sentences, marginally outperforming in SMATCH (85.65% vs 85.45%), though falling behind in other metrics. For compound sentences, the performance between the two approaches remains remarkably close, with the pipeline approach achieving slight advantages in BLEU (67.80% vs 67.58%) and BERT Score (98.12% vs 98.11%).

Italian language results demonstrate distinct patterns across different sentence structures. For simple sentences, which form the vast majority of the test set (545 examples), the non-pipeline approach consistently outperforms across all metrics. However, the most interesting results appear in compound sentences, where despite the small sample size (7 examples), the pipeline approach demonstrates superior performance across multiple metrics, including BLEU (65.39% vs 64.71%), METEOR (82.15% vs 79.54%), COMET (91.78% vs 89.36%), and others. This suggests that the pipeline approach might be particularly effective for handling compound structures in Italian, though the limited sample size warrants cautious interpretation.

Urdu language results present a clear pattern favoring the non-pipeline approach across all sentence types and metrics. In simple sentences (854 examples), the non-pipeline approach maintains a significant lead across all metrics, with particularly notable gaps in BLEU (55.81% vs 49.23%) and METEOR (53.48% vs 46.42%). This pattern continues and even amplifies in complex and compound sentences, where the performance gaps become more pronounced. The compound sentences show the most dramatic differences, with the non-pipeline approach outperforming by substantial margins (e.g., BLEU: 48.83% vs 37.16%). All results are listed in Table 10.

The overall analysis reveals several key insights about structural complexity's impact on performance. Generally, performance tends to decrease as structural complexity increases across all languages. The gap between pipeline and non-pipeline approaches often widens with increased structural complexity, though this pattern varies by language. The results also highlight the challenge of evaluating performance on complex and compound structures due to limited sample sizes, particularly in Italian and Urdu. While the non-pipeline approach generally shows superior performance, the pipeline approach demonstrates specific strengths in certain contexts, particularly in Italian compound sentences and some aspects of English complex and compound sentence processing. These findings suggest that while the non-pipeline approach might be preferable as a general solution, there could be

Lang.	**Sent. Type (ex.)**	**Pipeline**	**SMATCH (F1)**	**BLEU**	**METEOR**	**COMET**	**ROUGE**	**chrF**	**BERT_Score**
EN	Declarative (694)	Without	**93.44**	**72.75**	**89.58**	**95.93**	–	**86.10**	**98.73**
		With	92.98	70.75	88.66	95.39	–	84.87	98.54
	Exclamatory (71)	Without	**94.97**	**56.22**	**71.55**	**95.95**	–	**76.40**	**97.52**
		With	94.23	54.74	70.27	95.65	–	75.08	97.32
	Imperative (9)	Without	76.09	**77.40**	**95.38**	**96.28**	–	**87.25**	**99.33**
		With	**76.83**	74.04	94.49	96.11	–	84.97	**99.35**
	Interrogative (358)	Without	**93.94**	**70.39**	**86.98**	**95.53**	–	**84.41**	**98.34**
		With	93.37	68.98	86.06	95.14	–	83.34	98.17
IT	Declarative (486)	Without	**90.91**	**57.48**	**73.76**	**90.00**	–	**70.96**	**93.22**
		With	89.45	54.31	71.08	88.54	–	67.99	92.25
	Exclamatory (14)	Without	**91.92**	**47.27**	**59.42**	91.44	–	**65.80**	**86.95**
		With	88.66	46.39	59.40	**92.49**	–	65.82	86.59
	Imperative (1)	Without	83.33	18.99	32.25	71.11	–	18.63	79.09
		With	**91.66**	18.99	32.25	71.11	–	18.63	79.09
	Interrogative (54)	Without	**87.22**	**53.41**	**67.04**	**89.53**	–	**69.53**	**91.32**
		With	86.97	44.15	60.41	87.67	–	63.77	90.19
UR	Declarative (785)	Without	**79.72**	**54.93**	**52.56**	–	**59.18**	**50.46**	**88.27**
		With	76.25	48.11	45.11	–	52.64	43.60	86.17
	Exclamatory (27)	Without	**71.14**	**30.05**	**27.17**	–	**32.76**	**32.37**	**80.88**
		With	71.77	25.76	24.75	–	28.87	28.23	79.11
	Imperative (8)	Without	**72.06**	**31.84**	**33.72**	–	**34.63**	**37.64**	**79.16**
		With	62.25	22.63	24.82	–	25.83	29.68	77.35
	Interrogative (80)	Without	**83.90**	**69.90**	**68.72**	–	**73.04**	**69.45**	**92.28**
		With	81.02	64.96	63.80	–	68.13	64.48	90.55

Table 8: Impact of sentence type on evaluation results with and without pipeline for EN, IT, and UR. Bold indicates the better results.

Structure Type	**EN**		**IT**		**UR**	
	Train (%)	Test (%)	Train (%)	Test (%)	Train (%)	Test (%)
Simple	70.18	95.32	88.05	98.20	93.31	94.89
Complex	14.30	1.94	5.77	0.54	2.49	2.22
Compound	9.40	2.74	4.64	1.26	4.09	2.89
Compound-complex	6.12	0.00	1.54	0.00	0.10	0.00

Table 9: Training and test set structure type percentages.

value in considering a hybrid approach that leverages the strengths of both methods in specific linguistic contexts.

This comprehensive analysis underscores the importance of considering both structural complexity and language-specific characteristics in developing and evaluating natural language processing systems. The varying performance patterns across different languages and sentence types suggest that a one-size-fits-all approach might not be optimal and that future developments might benefit from language-specific optimizations and structural considerations.

C.4 Polarity Impact on Performance

Polarity based distribution analysis reveals interesting patterns across languages in both training and test sets. English and Urdu show similar distributions with a strong bias toward affirmative sentences, while Italian presents a notably different pattern with a majority of negative sentences. Specifically, in the training set, English (84.73%) and Urdu (88.09%) heavily favor affirmative sentences, while Italian shows a reverse trend with 60.80% negative sentences. This pattern persists in the test sets, where English and Urdu maintain high percentages of affirmative sentences (91.34% and 90.00% respectively), while Italian continues its bias toward negative sentences (63.42%). We have used TextBlob to extract these sentence types from the dataset. Table 11 provides statistical numbers for affirmative and negative sentence types for EN, IT, and UR test sets.

For **English** language performance, the results

Lang.	Imp. Type (ex.)	Pipeline	SMATCH (F1)	BLEU	METEOR	COMET	ROUGE	chrF	BERT_Score
EN	Simple (1079)	Without	**93.79**	**71.18**	**87.63**	**95.89**	–	**85.03**	**98.55**
		With	93.26	69.44	86.76	95.48	–	83.92	98.38
	Complex (22)	Without	85.45	**67.32**	**89.98**	**93.75**	–	**84.01**	**98.11**
		With	**85.65**	59.96	83.99	89.50	–	77.50	97.19
	Compound (31)	Without	**91.15**	67.58	87.45	**94.41**	–	83.27	98.11
		With	91.11	**67.80**	**87.45**	94.39	–	**83.22**	**98.12**
IT	Simple (545)	Without	**90.55**	**56.58**	**72.49**	**89.96**	–	**70.38**	**92.80**
		With	89.20	52.91	69.52	88.51	–	67.26	91.83
	Complex (3)	Without	**90.93**	**68.73**	**88.03**	**91.90**	–	**86.57**	**98.16**
		With	89.98	50.53	68.59	83.68	–	67.06	92.21
	Compound (7)	Without	**91.60**	64.71	79.54	89.36	–	80.22	94.08
		With	88.39	**65.39**	**82.15**	**91.78**	–	**81.22**	**95.63**
UR	Simple (854)	Without	**79.95**	**55.81**	**53.48**	–	**59.81**	**51.82**	**88.49**
		With	76.71	49.23	46.42	–	53.54	45.24	86.47
	Complex (20)	Without	**73.23**	**42.42**	**39.91**	–	**49.14**	**41.37**	**83.58**
		With	67.08	41.65	39.26	–	46.58	40.33	82.86
	Compound (26)	Without	**78.87**	**48.83**	**49.62**	–	**54.06**	**48.55**	**86.34**
		With	73.95	37.16	36.70	–	42.53	36.26	82.39

Table 10: Impact of structural complexity on evaluation results with and without pipeline for EN, IT, and UR. Bold indicates the better results.

Polarity Type	EN		IT		UR	
	Train (%)	Test (%)	Train (%)	Test (%)	Train (%)	Test (%)
Affirmative	84.73	91.34	39.20	36.58	88.09	90.00
Negative	15.27	8.66	60.80	63.42	11.91	10.00

Table 11: Training and test set polarity type percentages.

show consistently strong performance across both affirmative and negative sentences, with the non-pipeline approach maintaining a slight edge. With affirmative sentences (1034 examples), the non-pipeline approach achieves better scores across all metrics (SMATCH: 93.56%, BLEU: 71.14%, METEOR: 87.89%). The performance on negative sentences (98 examples) is remarkably similar, with the non-pipeline approach again outperforming (SMATCH: 93.53%, BLEU: 69.64%, METEOR: 85.32%). The minimal performance difference between affirmative and negative sentences suggests that English processing is robust across polarity types.

Italian language results present an interesting case given its unique distribution favoring negative sentences. For affirmative sentences (203 examples), the non-pipeline approach shows strong performance (SMATCH: 90.18%, BLEU: 60.85%, METEOR: 76.15%). The performance on negative sentences (352 examples), which constitute the majority, remains strong with the non-pipeline approach (SMATCH: 90.78%, BLEU: 54.39%, METEOR: 70.66%). Notably, while the pipeline approach consistently trails behind, the performance gap remains relatively stable across both polarities, suggesting consistent handling of both sentence types.

Urdu language results reveal an interesting pattern where negative sentences, despite being the minority (90 examples), actually show slightly better performance than affirmative ones. The non-pipeline approach achieves higher SMATCH scores on negative sentences (82.45% vs 79.47% for affirmative), though other metrics remain comparable. This suggests that the processing of negative sentences in Urdu might be more straightforward than initially expected. The pipeline approach maintains the same pattern but with lower overall scores, showing larger performance gaps compared to the non-pipeline approach. Table 12 provides results for affirmative and negative sentence types with and without pipeline for EN, IT, and UR test sets.

The analysis reveals several key insights about polarity's impact on performance. First, the systems generally handle both polarities well, with

Lang.	Imp. Type (ex.)	Pipeline	SMATCH (F1)	BLEU	METEOR	COMET	ROUGE	chrF	BERT_Score
EN	Affirmative (1034)	Without	**93.56**	**71.14**	**87.89**	**95.78**	–	**85.14**	**98.53**
		With	93.07	69.37	86.88	95.32	–	83.98	98.36
	Negative (98)	Without	**93.53**	**69.64**	**85.32**	**96.09**	–	**83.14**	**98.61**
		With	92.86	67.58	85.16	95.52	–	81.58	98.30
IT	Affirmative (203)	Without	**90.18**	**60.85**	**76.15**	**92.15**	–	**74.94**	**93.77**
		With	89.14	57.98	73.59	90.94	–	72.08	92.77
	Negative (352)	Without	**90.78**	**54.39**	**70.66**	**88.69**	–	**68.09**	**92.32**
		With	89.22	50.22	67.42	87.13	–	64.77	91.37
UR	Affirmative (810)	Without	**79.47**	**55.36**	**53.23**	–	**59.51**	**51.59**	**88.28**
		With	76.25	48.47	45.92	–	52.89	44.86	86.20
	Negative (90)	Without	**82.45**	**54.85**	**51.65**	–	**58.46**	**50.59**	**88.65**
		With	77.92	50.85	46.48	–	54.66	44.93	86.89

Table 12: Impact of sentence polarity (affirmative and negative) on evaluation results with and without pipeline for EN, IT, and UR. Bold indicates the better results.

relatively small performance variations between affirmative and negative sentences within each language. Second, the non-pipeline approach consistently outperforms across all languages and polarities, suggesting its robustness in handling different sentence types. Third, the unique distribution in Italian, with its preference for negative sentences, doesn't seem to negatively impact performance, indicating that the systems have adequately adapted to this linguistic characteristic.

These findings carry important implications for system development and optimization. The consistent performance across polarities suggests that current approaches are well-balanced in handling both affirmative and negative constructions. However, the persistent advantage of the non-pipeline approach indicates that maintaining semantic coherence through unified processing might be particularly important for preserving meaning across different polarity types. The results also highlight the importance of considering language-specific characteristics in system development, as demonstrated by the successful handling of Italian's negative-heavy distribution and Urdu's superior performance on negative sentences despite their minority status in the training data.

C.5 Analyzing the Impact of Sentence Voices

The distribution analysis based on sentence voices shows a strong bias toward active voice across all three languages in both training and test sets. In the training data, the distribution is remarkably similar across languages, with active voice dominating at 90.58% for English, 92.06% for Italian, and 92.01% for Urdu. This pattern becomes even more pronounced in the test sets, where active voice sentences increase to 93.37%, 94.05%, and 93.78% respectively. The consistency of this distribution across languages suggests a universal preference for active voice constructions in natural language. We have used spaCy to classify these sentences based on the voice types from the dataset. Table 13 presents active and passive voice examples in training and test sets of EN, IT, and UR datasets.

English language results reveal some fascinating patterns in the handling of voice types. For active voice sentences (1057 examples), the non-pipeline approach demonstrates superior performance across all metrics (SMATCH: 93.57%, BLEU: 70.33%, METEOR: 87.36%). However, the most interesting findings emerge in passive voice sentences (75 examples), where we see a mixed pattern of success. The pipeline approach achieves a higher SMATCH score (94.88% vs 93.32%), marking one of the few instances where it outperforms the non-pipeline approach. Despite this, the non-pipeline approach maintains higher scores in other metrics for passive constructions, with notably higher BLEU (80.44% vs 78.21%) and METEOR (92.00% vs 90.36%) scores. Interestingly, both approaches achieve better scores on several metrics for passive sentences compared to active ones, suggesting that passive constructions, though less frequent, might be more straightforward to process.

Italian language performance shows a clear preference for the non-pipeline approach across both voice types. With active voice sentences (522 examples), the non-pipeline approach consistently outperforms (SMATCH: 90.46%, BLEU: 57.34%, METEOR: 73.07%). For passive voice sentences (33 examples), despite the small sam-

Voice Type	EN		IT		UR	
	Train (%)	Test (%)	Train (%)	Test (%)	Train (%)	Test (%)
Active	90.58	93.37	92.06	94.05	92.01	93.78
Passive	9.42	6.63	7.94	5.95	7.99	6.22

Table 13: Training and test set voice type percentages.

Lang.	Imp. Type (ex.)	Pipeline	SMATCH (F1)	BLEU	METEOR	COMET	ROUGE	chrF	BERT_Score
EN	Active (1057)	Without	**93.57**	**70.33**	**87.36**	**95.81**	–	**84.65**	**98.51**
		With	92.93	68.57	86.47	95.31	–	83.48	98.32
	Passive (75)	Without	93.32	**80.44**	**92.00**	**95.75**	–	**89.40**	**98.93**
		With	**94.88**	78.21	90.36	95.64	–	87.84	98.84
IT	Active (522)	Without	**90.46**	**57.34**	**73.07**	**90.15**	–	**70.91**	**92.89**
		With	89.11	53.72	70.11	88.66	–	67.75	91.93
	Passive (33)	Without	**92.19**	**47.55**	**66.23**	**86.94**	–	**65.66**	**92.16**
		With	90.60	42.63	62.83	86.32	–	62.45	91.04
UR	Active (844)	Without	**79.85**	**55.51**	**53.40**	–	**59.64**	**51.64**	**88.31**
		With	76.44	48.97	46.33	–	53.38	45.08	86.31
	Passive (56)	Without	**78.54**	**52.31**	**48.04**	–	**55.83**	**49.31**	**88.56**
		With	76.06	44.77	40.67	–	48.31	41.66	85.77

Table 14: Impact of sentence voice (active/passive) on evaluation results with and without pipeline for EN, IT, and UR. Bold indicates the better results.

ple size, the non-pipeline approach maintains its advantage with higher scores across all metrics (SMATCH: 92.19%, BLEU: 47.55%, METEOR: 66.23%). Notable is the fact that while SMATCH scores are actually higher for passive sentences, other metrics show lower performance compared to active voice, suggesting that while semantic preservation might be easier in passive constructions, generating natural language output becomes more challenging.

Urdu language results demonstrate a consistent pattern favoring the non-pipeline approach, but with some interesting nuances between active and passive voice handling. For active voice sentences (844 examples), the non-pipeline approach shows strong performance (SMATCH: 79.85%, BLEU: 55.51%, METEOR: 53.40%). In passive voice sentences (56 examples), while the non-pipeline approach still outperforms, there's a slight decline in performance across most metrics (SMATCH: 78.54%, BLEU: 52.31%, METEOR: 48.04%). This suggests that Urdu might find passive constructions more challenging to process compared to active ones, unlike the pattern seen in English and Italian. All evaluation results are presented in Table 14.

Several key insights emerge from this analysis about the impact of voice on processing performance. First, the high proportion of active voice sentences in training data doesn't necessarily translate to better performance on active constructions — in fact, both English and Italian show higher SMATCH scores for passive voice sentences. Second, the pipeline approach shows particular promise in handling English passive constructions, achieving its most notable success in this category. Third, the impact of voice on performance varies significantly by language, with Urdu showing a different pattern from English and Italian.

These findings have important implications for system development and optimization. The successful handling of passive voice despite its lower representation in training data suggests that current approaches are robust in managing syntactic variations. However, the varying patterns across languages indicate that voice handling might benefit from language-specific optimizations. The superior performance of the pipeline approach on English passive constructions also suggests that decomposing complex syntactic transformations might be beneficial in specific linguistic contexts. Future developments might consider leveraging these insights to create more nuanced, language-aware approaches to handling voice variations.