Cross-Domain Semantic Fidelity Evaluation for Meaning-to-Text Generation

Davan Harrison and Marilyn Walker
Natural Language and Dialogue Systems Lab
University of California, Santa Cruz
{vharriso, mawalker}@ucsc.edu

Abstract

Slot Error Rate (SER) is the standard metric for evaluating semantic accuracy in meaning-to-text generation, but computing it has historically required domain-specific scripts that do not generalize across datasets. We present a cross-domain SER evaluation framework that replaces hand-crafted rules with a learned slot extraction model. We adapt Llama-3.2-3B-Instruct with LoRA, updating only 0.34% of its parameters, and show that this small adapted model outperforms prompted frontier LLMs by a wide margin on structured extraction across 23 dialogue domains. We further apply overgenerate-and-rank to the extraction task itself, generating multiple candidate meaning representations and selecting the best one with a trained ranker, which improves SER-Accuracy from 75% to 88%. We combine the extraction model with a Natural Language Inference (NLI) verification baseline through learned per-example routing, achieving 90.0% accuracy on held-out evaluation pairs without any domain-specific rule engineering. We compare our framework against published rule-based SER tools and show that our learned approach matches or outperforms hand-crafted scripts on all six comparable domains.

1 Introduction

Meaning-to-text generation systems in task-oriented dialogue must faithfully realize all slots and values in a structured meaning representation (MR) while allowing variation in surface form. When a generator produces "a family-friendly Italian place near Riverside" for an MR containing `food: Italian`, `familyFriendly: yes`, and `near: riverside`, the text is semantically faithful. When it adds "with free parking" or drops the food type, it introduces a semantic error. Slot Error Rate (SER) measures these errors by counting insertions, deletions, and substitutions of slot-value pairs relative to the input MR (Wen et al., 2015). Table 1 illustrates these error types with examples from two domains.

Despite its importance, computing SER has been a persistent bottleneck for the Natural Language Generation (NLG) evaluation community. Existing SER tools rely on domain-specific scripts that use pattern matching, keyword dictionaries, and hand-crafted rules. Wen et al. (2015) developed separate scripts for the restaurant, hotel, laptop, and TV domains. Juraska (2022) built a more general slot aligner that achieves 95–100% precision on ViGGO and E2E NLG Challenge outputs but still requires per-domain configuration of slot types and synonym dictionaries. Each new domain, dataset, or generation model potentially requires a new evaluation script. This cost is especially problematic for modern LLM-based generators, which produce more lexically diverse outputs than template-based systems and therefore challenge the fixed vocabularies of rule-based tools.

We propose a learned approach to SER evaluation that removes the need for per-domain engineering. Our framework has three requirements. First, it must be *form-agnostic*, insensitive to surface variation that does not alter meaning, so that "cheap," "affordable," and "won't break the bank" all count as valid realizations of `priceRange: cheap`. Second, it must be *cross-domain*, applicable to restaurants, hotels, video games, auto repair, and other task-oriented settings without re-engineering. Third, it must be *automatable* and usable at the scale required for generation pipelines where thousands of candidates must be scored per experiment.

We address these requirements with a two-stage pipeline. In the first stage, we extract a predicted MR from the generated text using a cross-domain slot extractor. In the second stage, we compare the extracted MR to the gold MR using a deterministic alignment procedure to compute substitutions, deletions, and insertions. We experiment

Proceedings of the Fifth Workshop on Generation, Evaluation and Metrics (GEM), pages 443–455

with three extraction approaches of increasing sophistication: prompt-based learning (PBL) with frontier LLMs, LoRA (Hu et al., 2022) adaptation of a small instruction-tuned model (Grattafiori et al., 2024), and overgenerate-and-rank (OGR) extraction that produces multiple candidate MRs and selects the best one with a trained ranker. We also implement an NLI-based baseline following Dušek and Kasner (2020) and develop a per-example routing strategy that combines the extraction and NLI approaches.

We train and evaluate our system[1] on a multi-domain corpus assembled from six public meaning-to-text datasets spanning 23 topic domains. We evaluate on both in-domain and out-of-domain topics and compare against published rule-based SER tools where available. We make the following contributions:

1. We present a cross-domain SER evaluation framework that generalizes across 23 dialogue domains without handcrafted rules or per-domain configuration.
2. We show that LoRA adaptation of a small 3B-parameter model outperforms prompted frontier LLMs on structured slot extraction, and that overgenerate-and-rank improves extraction accuracy by 13 points.
3. We develop per-example routing strategies that combine learned extraction with NLI-based verification. Gradient-boosted routing (Chen and Guestrin, 2016) with cost-sensitive weighting achieves 90.0% accuracy, capturing 87% of the gap to the per-example oracle.
4. We provide a comparative evaluation against published rule-based SER tools, demonstrating that our learned approach matches or outperforms domain-specific scripts on all six comparable domains.

2 Related Work

Slot Error Rate and rule-based tools. SER was introduced for evaluating statistical NLG in task-oriented dialogue (Wen et al., 2015). Computation has historically relied on domain-specific scripts: the RNNLG ERRScorer[2] handles four RNNLG domains with separate rule sets, the E2E challenge script (Dušek and Kasner, 2020) covers the E2E restaurant domain, and Juraska's slot aligner[3] (Juraska, 2022) supports ViGGO, E2E, and MultiWOZ (Budzianowski et al., 2018) through a five-type slot classification system augmented with synonym dictionaries. While the slot aligner is the most general of these, it still requires per-domain configuration of slot types and value mappings. We compare against all three tool families in our experiments.

MR and Surface Realization	**Error Annotation**
Domain: video_games	SER = 0.333 (2*I* / 6)
MR: name: RollerCoaster Tycoon; rating: good; genres: strategy; platforms: PC; steam: not_on_steam; macOS: not_released *"If you enjoy strategy and simulators, try RollerCoaster Tycoon. It's received good ratings and is available for **Xbox**, PC, as well as **Linux**, though it's not on Steam and doesn't have a Mac release."*	**Insertions:** *Xbox* and *Linux* as platforms not in MR. **Deletions:** — **Substitutions:** — The model hallucinated two platform values. An extraction-based evaluator must detect that Xbox and Linux are not grounded in the input MR.
Domain: auto_repair	SER = 0.286 (2*I* / 7)
MR: appointment: Confirming; customer: Michael Brown; time: 7am; store: intellignet auto solutions; year: 2018; vehicle: CRV; reason: noise under hood *"Confirming an appointment for you, Michael Brown at 7am at intellignet auto solutions for your 2018 **honda** CRV. Service will include **oil change** and attention to noise under hood."*	**Insertions:** *honda* (brand not in MR); *oil change* (service not in MR). **Deletions:** — **Substitutions:** — The model added plausible but unsupported details. Rule-based tools would need a separate script for each domain to catch these.

Table 1: Semantic errors across two domains. Bolded text in the surface realization marks content not present in the input MR. Both examples contain only insertion errors, but our framework also detects deletions and substitutions.

NLI for semantic accuracy in NLG. Dušek and Kasner (2020) propose using NLI to evaluate semantic accuracy in data-to-text generation by converting each slot-value pair into a template hypothesis and checking entailment against the generated text. Huidrom et al. (2022) reproduce and confirm this approach. Most directly related to our work, Lee et al. (2023) propose Schema-Guided

[1] Code available at github.com/Vrindiesel/xdomain-ser.

[2] github.com/shawnwun/RNNLG

[3] github.com/jjuraska/slug2slug

Semantic Accuracy (SGSAcc), which extends NLI-based evaluation to handle categorical slots on the Schema Guided Dialogue dataset (Rastogi et al., 2020). SGSAcc and our NLI baseline share the core idea of template-based entailment checking, but differ in scope and method. SGSAcc evaluates individual slot realizations independently, while our extraction pipeline recovers a full structured MR, enabling detection of substitution errors where the text realizes a slot with the wrong value. We also combine NLI with learned extraction through routing, which neither prior NLI-based approach explores.

Faithfulness evaluation beyond NLG. The broader faithfulness evaluation literature has developed complementary approaches. NLI-based methods such as SummaC (Laban et al., 2022) and AlignScore (Zha et al., 2023) aggregate sentence-pair entailment scores for document-level consistency checking. Claim decomposition methods such as FActScore (Min et al., 2023) break generated text into atomic facts and verify each against a knowledge source. QA-based methods such as Data-QuestEval (Rebuffel et al., 2021) generate questions from the source and answer them using the generated text. Small task-specific models such as MiniCheck (Tang et al., 2024) achieve GPT-4-level fact-checking at a fraction of the cost. These systems produce aggregate quality scores or binary judgments rather than the per-slot error attribution (substitutions, deletions, insertions) that SER requires. Our work addresses the specific need for structured, slot-level evaluation in meaning-to-text generation.

LLM-based NLG evaluation. Recent work has explored using LLMs as evaluators for NLG quality. G-Eval (Liu et al., 2023) uses GPT-4 with chain-of-thought prompting and probability-weighted scoring, achieving strong correlation with human judgments on summarization. The survey by Gao et al. (2025) categorizes LLM-based evaluation into metrics derived from LLMs, prompting, fine-tuning, and human-LLM collaboration. These methods target broad quality dimensions such as fluency and coherence. We address a different problem, evaluating whether specific slot-value pairs are correctly realized and producing interpretable error counts rather than scalar scores. We adopt the probability-weighted scoring technique from G-Eval for our ranking model.

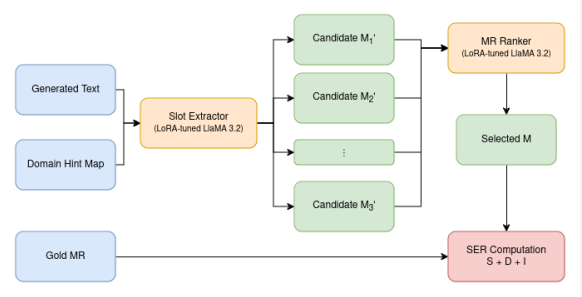

Figure 1: Cross-domain SER evaluation pipeline. The extractor reconstructs an MR from the generated utterance using a domain hint map. In the OGR variant, k candidate extractions are scored by a ranking model before comparison to the gold MR.

Parameter-efficient fine-tuning for structured extraction. LoRA (Hu et al., 2022) and related parameter-efficient methods have been widely adopted for adapting LLMs to structured tasks. Recent work demonstrates that LoRA-adapted small models can match or outperform much larger prompted models on named entity recognition, slot filling, and relation extraction (Sainz et al., 2024; Zhou et al., 2024). We apply this insight to SER evaluation, showing that a 3B-parameter model with 0.34% trainable parameters outperforms prompted ChatGPT 4o (OpenAI, 2024) by 43% on cross-domain slot extraction.

3 Approach

We frame SER computation as a two-stage pipeline. Given a generated utterance and a reference MR, we first extract a predicted MR from the text using a domain-aware slot extractor, then compare the predicted MR to the reference MR using deterministic alignment to identify substitutions (S), deletions (D), and insertions (I). SER is computed as $(S + D + I)/N$ where N is the number of slots in the reference MR. Figure 1 illustrates the full pipeline. We describe each component below.

3.1 Domain Hint Maps

To support cross-domain generalization without per-domain rules, we provide the extraction model with a *domain hint map*: a structured description of the target domain's slot schema listing each slot type with a short natural-language description and example values. For instance, the E2E hint map includes an entry for `priceRange` described as "price range, e.g., cheap, moderate, expensive." The hint map tells the model what kinds of information to look for without dictating the surface forms those

values might take. Adapting the system to a new domain requires only writing a hint map of slot names and descriptions, rather than engineering extraction rules or populating gazetteers.

3.2 Slot-Value Extraction

We experiment with three extraction approaches. All three share the same input-output interface: given a domain hint map, optional exemplars, and a generated utterance, the model outputs a serialized list of extracted slot-value pairs. They differ in how the model acquires the extraction capability.

Prompt-based learning (PBL). We prompt instruction-tuned LLMs with three components: a domain schema listing possible slot types, 5 in-domain exemplars showing utterance-MR pairs, and the input utterance. We evaluate ChatGPT 4o (OpenAI, 2024) via the OpenAI API and Llama-3.2-3B-Instruct (Grattafiori et al., 2024) run locally. Neither model is fine-tuned on extraction data, making this a fast, zero-training baseline.

LoRA-adapted extraction. We adapt Llama-3.2-3B-Instruct using Low-Rank Adaptation (Hu et al., 2022). LoRA freezes the pretrained weights and learns a low-rank update $\Delta W = BA$ where $B \in \mathbb{R}^{d\times r}$ and $A \in \mathbb{R}^{r\times d}$, with $r \ll d$. We use rank $r = 4$, scaling factor $\alpha = 16$, and dropout $p = 0.05$, targeting both attention and MLP projections. We update only 0.34% of the model's parameters. We initialize adapters using Explained Variance Adaptation (Paischer et al., 2024) and apply Rank-Stabilized scaling (Kalajdzievski, 2023). The model receives the same prompt structure as the PBL baseline but has been fine-tuned on task-specific extraction data from 20 training topics.

Overgenerate-and-rank (OGR) extraction. For each input utterance, we generate $k = 10$ candidate extractions using beam search, then score each candidate with a trained ranking model and select the top-scoring extraction. A single extraction attempt may miss a slot or hallucinate an extra one, but across multiple attempts the correct extraction is likely to appear at least once. The ranker's job is to identify which candidate is most faithful.

3.3 MR Ranking Model

We train a ranking model to select the best extraction from a pool of k candidates. The ranker is a separate LoRA-adapted Llama-3.2-3B-Instruct fine-tuned on tuples of surface text, gold MR, candidate MR, and quality grade. We assign quality grades on a 7-point scale (0–6) based on micro slot-F1 between each candidate MR and the gold MR, then balance the training data by sampling equally per grade within each topic.

At inference, we do not simply predict a single grade via argmax. Following the probability-weighted scoring approach of Liu et al. (2023), we extract next-token logits for the seven digit tokens, apply softmax, and compute a weighted score that concentrates on the upper grades (3–6) where fine-grained discrimination matters most. The resulting continuous score in $[0, 3]$ enables finer ranking among high-quality candidates than discrete prediction allows.

3.4 NLI-Based Verification Baseline

As a complementary approach, we implement an NLI-based baseline following Dušek and Kasner (2020). For each slot-value pair in the gold MR, we construct a natural language hypothesis sentence and check whether the generated text entails it using RoBERTa-large fine-tuned on MultiNLI (Liu et al., 2019; Williams et al., 2018). We design templates for all 133 slot types across our evaluation domains, covering three categories: value-substitution templates (e.g., `name` → "The name is {value}."), Boolean templates with conditional positive/negative hypotheses, and a small number of conditional templates for domain-specific semantics. We recover an MR by including slot-values whose entailment probability exceeds a threshold $\tau = 0.3$, selected by sweeping $\tau \in \{0.3, 0.4, 0.5, 0.6, 0.7\}$.

The NLI approach has complementary strengths to extraction. We find that NLI achieves higher substitution accuracy (0.928 vs. 0.872) and insertion accuracy (0.956 vs. 0.910), while extraction achieves higher deletion accuracy (0.900 vs. 0.865). We exploit this complementarity in our routing approach.

3.5 Per-Example Routing

We develop three routing strategies that select between LoRA extraction and NLI verification on a per-example basis. The first uses a single threshold on the LoRA ranker's top score: when the ranker is confident (score ≥ 2.90), we use the LoRA extraction, otherwise we fall back to NLI. The second trains a logistic regression classifier on 22 per-example features in four groups: ranking

Source	Topics	Turns	Slots	Avg MR
Training				
Taskmaster-1	6	36,082	8–25	1.8–2.6
Taskmaster-2	6	34,716	11–31	1.9–2.7
Taskmaster-3	1	116,926	21	3.1
E2E NLG	1	42,061	8	5.4
RNNLG	2	6,125	13–16	2.6–4.8
Test: In-Domain (9 topics)				
TM1/E2E/RNNLG	9	9,820	8–16	1.8–6.9
Test: Out-of-Domain (3 topics)				
ViGGO	1	1,026	13	4.8
RNNLG hotel	1	563	12	2.5
RNNLG laptop	1	2,637	20	4.9

Table 2: Dataset summary. Turns = agent utterances. Slots = unique slot types per domain. Avg MR = mean slot-value pairs per utterance. The corpus spans 23 topics from 6 public datasets. Full per-topic statistics appear in Appendix A.

confidence (top score, score gap, score spread), MR complexity (gold slot count, extracted slot count, their ratio), NLI confidence (mean and minimum entailment probability, coverage fraction), and topic indicators. The third replaces logistic regression with gradient-boosted trees (Chen and Guestrin, 2016) using cost-sensitive sample weighting that focuses learning on disagreement instances where the two methods produce different results. We tune all routing parameters on a stratified 50/50 dev/test split.

4 Experimental Setup

4.1 Datasets

We assemble a multi-domain corpus from six public meaning-to-text datasets: E2E NLG Challenge (Novikova et al., 2017), RNNLG (Wen et al., 2016), Taskmaster-1, Taskmaster-2, Taskmaster-3 (Byrne et al., 2019), and ViGGO (Juraska et al., 2019). Together these span 23 topic domains in task-oriented dialogue. We use 20 topics for training and hold out 12 for testing: 9 in-domain topics (seen during training) and 3 out-of-domain topics (hotel, laptop, and video games) whose schemas were never seen during training. Table 2 summarizes the corpus by source dataset, and full per-topic statistics appear in Appendix A.

4.2 Data Selection

We select training examples using a Facility-Location objective (Wei et al., 2014) that balances slot-value coverage and surface diversity. We use $K = 200$ examples per topic, producing roughly 4,000 training examples from over 275,000 available turns. We show in Appendix C that 200 examples per topic produces near-optimal performance, outperforming both $K = 100$ (insufficient coverage) and $K = 400$ (mild overfitting to training domains).

Model	P	R	F1	SER-Acc
LoRA Llama 3.2	.845	.795	.819	.752
ChatGPT 4o	.590	.555	.573	.499
Llama 3.2 Instruct	.507	.531	.519	.343

Table 3: Overall extraction results (micro-averaged over all 12 test topics).

4.3 Evaluation Protocol

We evaluate in two complementary settings.

Evaluation 1: MR extraction quality. We measure how accurately the model recovers MRs from clean text using micro slot-F1 and SER-Accuracy ($1 - \text{SER}$).

Evaluation 2: SER agreement. We test whether the system produces correct SER scores on text-MR pairs with known semantic errors. We construct 9,042 evaluation pairs by systematically perturbing gold MRs to create modified MRs with known substitution, deletion, and insertion counts, then pair each modification with the original text. We report per-error-type accuracy (S_{acc}, D_{acc}, I_{acc}), all-correct accuracy (All_{acc}, where all three error counts must match), and SER mean absolute error (MAE). This evaluation directly tests the system's utility as an automatic SER tool.

4.4 Baselines

We compare six system configurations: three extractors (ChatGPT 4o, Llama-3.2-3B-Instruct, LoRA-adapted Llama-3.2-3B) each with and without OGR ranking. For the SER agreement evaluation, we additionally compare NLI verification, two routing strategies, and published rule-based tools where available: the RNNLG ERRScorer (Wen et al., 2015), the E2E slot error script (Dušek and Kasner, 2020), and Juraska's slot aligner (Juraska, 2022).

5 Results

5.1 Extraction: PBL vs. LoRA Adaptation

Table 3 shows overall extraction results across all test topics. The LoRA-adapted model achieves a

System	F1	SER-Acc	F1 (R)	SER-Acc (R)
ChatGPT 4o	.572	.50	.615	.56
Llama 3.2	.519	.34	.638	.48
LoRA Llama 3.2	.819	.75	.918	.88

Table 4: Extraction results with and without OGR ranking (R). Ranking uses $k = 10$ candidates and the trained slot-F1 ranker.

micro slot-F1 of 0.819, a 43% relative improvement over prompted ChatGPT 4o (0.573) and a 58% improvement over unadapted Llama 3.2 (0.519). We find that task-specific adaptation confers a large advantage over prompting alone, even when the prompted model is substantially larger.

The gains are consistent across both in-domain and out-of-domain topics. Even on the three OOD topics, the adapted model achieves F1 scores of 0.668 (video games), 0.846 (hotel), and 0.786 (laptop). The hotel domain generalizes well because its schema overlaps with the restaurant domain seen in training. ViGGO is the hardest OOD topic because it includes slot types such as `has_multiplayer` and `steam_availability` that have no close analogs in the training data, and its utterances tend to be longer and more informationally dense. We find that very low LoRA ranks of $r = 2$–4 suffice for this task, with no improvement from higher ranks (Appendix B).

5.2 Overgenerate-and-Rank

Table 4 shows the effect of overgenerate-and-rank on extraction quality. Ranking reliably boosts all three extractors, with the largest gains for the LoRA model. F1 improves from 0.819 to 0.918 and SER-Accuracy from 0.75 to 0.88. This 13-point improvement reaches near-90% accuracy on semantic fidelity estimation without any domain-specific rules. We find that OGR is effective for evaluation, not just generation. The ranking stage simultaneously improves all three error-type accuracies and reduces SER MAE by nearly half.

5.3 SER Agreement with Routing

Table 5 presents the SER agreement results on 4,510 held-out evaluation pairs. The two base methods have complementary strengths. LoRA achieves higher deletion accuracy (0.893 vs. 0.854) while NLI achieves higher substitution accuracy (0.927 vs. 0.867) and insertion accuracy (0.952 vs. 0.906). All routing strategies substantially outperform either baseline. We find that LR routing achieves $\text{All}_{\text{acc}} = 0.868$, a 10-point improvement over LoRA and 6 points over NLI. Gradient-boosted routing with cost-sensitive weighting (XGB Routing) further improves to $\text{All}_{\text{acc}} = 0.900$, capturing 87% of the gap to the oracle upper bound of 0.914. We attribute the improvement over LR routing to the ability of tree-based models to capture nonlinear interactions between the LoRA ranker's confidence and NLI coverage signals. Cost-sensitive weighting focuses learning on the roughly 15% of instances where the two methods disagree and provides effective regularization against overfitting. All pairwise differences between XGB routing and baselines are significant by McNemar's test ($p < 0.001$).

Method	n	S	D	I	All	MAE
LoRA + Rank	4510	.867	.893	.906	.764	.078
NLI	4510	.927	.854	.952	.805	.050
Score Routing	4510	.933	.912	.963	.861	.035
LR Routing	4510	.929	.925	.964	.868	.036
XGB Routing	4510	.938	.947	.971	**.900**	**.029**
Oracle	4510	—	—	—	.914	—

Table 5: SER Agreement (Evaluation 2) on the held-out test split. S, D, I = per-error-type accuracy. All = all-correct accuracy. XGB Routing = gradient-boosted trees with cost-sensitive weighting. Best learned result in **bold**.

Topic	Method	S	D	I	All	MAE
restaurant	LR Routing	.989	1.00	1.00	**.989**	.003
	RNNLG	.972	.876	.957	.827	.055
hotel	LR Routing	.973	.979	.985	**.944**	.020
	RNNLG	.902	.658	.823	.504	.193
TV	Score Rt	.927	.870	.943	**.788**	.048
	RNNLG	.764	.538	.838	.336	.174
laptop	NLI	.951	.917	.960	**.870**	.024
	RNNLG	.873	.676	.881	.544	.117
E2E rest.	LR Routing	.836	.754	.942	**.721**	.043
	E2E script	.881	.730	.940	.699	.046
video games	LR Routing	.892	.868	.934	**.753**	.071
	ViGGO	.774	.916	.983	.720	.071

Table 6: Comparison to rule-based SER tools on the six topics where published tools are available. Best All-correct accuracy per topic in **bold**.

5.4 Comparison to Rule-Based Tools

Table 6 compares our learned methods against published rule-based SER tools on the six topics where both can be evaluated. Our routing approaches match or outperform the rule-based tools on all

six topics. The largest gains appear on hotel (+44 points) and TV (+45 points), where the RNNLG scripts suffer from low deletion accuracy due to limited paraphrase coverage. On video games, LR routing (0.753) exceeds the hand-tuned ViGGO aligner (0.720), closing a gap that LoRA extraction alone could not bridge. The only topic where the rule-based tool remains competitive is E2E NLG, where our routing approach still achieves a 2-point advantage (0.721 vs. 0.699).

The per-topic results reveal a consistent rule-based failure mode. Pattern-matching tools achieve high insertion accuracy but suffer from low deletion accuracy. Detecting deletions requires recognizing that the text realizes a slot even when using unexpected phrasing, and rule-based tools with fixed keyword dictionaries miss indirect realizations. Our learned models handle paraphrases more robustly.

5.5 Error Analysis

We observe an interesting interaction between extraction method and MR complexity. LoRA excels on small MRs (91.4% accuracy on 3–4 slots) but degrades to 63.2% on large MRs with 7–8 slots, while NLI is more robust (perfect on 3–4 slots, 77.4% on 7–8 slots). Routing helps precisely because it can select NLI for complex MRs where LoRA struggles. The improvements are largest on topics where neither baseline clearly dominates, such as pizza ordering where XGB routing achieves 0.841 versus 0.703 (LoRA) and 0.681 (NLI). Of the remaining 10.0% errors after XGB routing, roughly half are cases where both methods fail simultaneously on complex paraphrases. ViGGO remains the hardest topic across all methods due to its Boolean slot semantics, long utterances, and list-valued slots.

We also examined the extraction model's apparent errors more closely and found that 77.5% of slot-value mismatches between extracted and gold MRs are semantically equivalent paraphrases (e.g., "moderate" vs. "mid-range"), not genuine extraction failures. Only 20.8% of mismatches represent actual errors. We believe the extraction model is more accurate than exact-match evaluation indicates, and that soft evaluation metrics incorporating semantic similarity would better reflect true extraction quality.

We also evaluated our framework on 1,000 gold-annotated personality-conditioned outputs from a separate generation study, finding that score-threshold routing achieves 82.6% accuracy on this out-of-distribution data. Details appear in Appendix D.

Method	LLM Pers.	Seq2Seq	E2E (no style)
Rule-based	.788	.484	.699
LoRA	.804	.612	.719
NLI	.800	.720	.599
Score Routing	**.928**	.724	.719
LR Routing	.860	**.736**	**.721**

Table 7: All-correct SER accuracy across three conditions varying in degree of stylistic variation. LLM Pers. = LLM-generated personality outputs; Seq2Seq = seq2seq personality outputs; E2E = standard outputs with no personality conditioning.

5.6 Robustness to Stylistic Variation

A practical SER evaluator must handle not only standard NLG outputs but also stylistically varied text produced by personality-conditioned or otherwise controlled generators. We evaluate robustness by comparing method performance across three conditions that vary in the degree of surface variation: LLM-generated personality-conditioned outputs (high variation, novel vocabulary), seq2seq personality-conditioned outputs from PERSONAGE-style models (Oraby et al., 2018) (moderate variation, more formulaic surface forms), and standard E2E NLG Challenge outputs with no personality conditioning (no stylistic variation). The first two conditions are drawn from a gold-annotated evaluation of 1,000 personality-conditioned restaurant-domain outputs across five Big Five personality types. The third is drawn from the E2E evaluation in Table 6.

Table 7 reveals three patterns. First, the rule-based method is most sensitive to stylistic variation. It achieves .788 on LLM outputs, which tend to use standard restaurant vocabulary, but drops to .484 on seq2seq outputs, where personality conditioning introduces hedges, discourse markers, and aggregation patterns that fall outside the aligner's pattern-matching rules.

Second, LoRA and NLI show complementary sensitivity profiles. LoRA performs well on LLM outputs (.804) and E2E outputs (.719) but struggles with seq2seq outputs (.612), likely because the seq2seq surface forms differ from the instruction-tuned text the extraction model was trained on. NLI shows the opposite pattern, achieving the best single-method accuracy on seq2seq outputs (.720) but the worst on E2E outputs (.599), which we believe reflects the NLI model's sensitivity to utter-

ance structure.

Third, routing methods are the most robust across all three conditions, achieving the highest or near-highest accuracy in every column. The consistency of routing across conditions with very different surface characteristics supports our claim that combining LoRA and NLI through learned routing produces an evaluator that is robust to stylistic variation, not merely tuned to one generation paradigm.

6 Discussion

Our approach occupies a different point in the precision-portability trade-off than rule-based tools. The slot aligner achieves 95–100% precision on domains where its dictionaries are well-populated but requires per-domain configuration. Our model achieves roughly 85% precision without ranking and 92% with ranking, but operates across 23 domains without domain-specific engineering. The two approaches have complementary strengths, and a hybrid deployment that uses rule-based tools on well-configured domains and our learned approach elsewhere may be the most practical strategy.

We also compared our framework against two general-purpose factual consistency metrics, AlignScore (Zha et al., 2023) and MiniCheck (Tang et al., 2024), on our evaluation pairs. We find that our routing approach achieves 97.6% pairwise accuracy at ranking pairs by SER, compared to 87.2% for AlignScore and 78.6% for MiniCheck. The general-purpose metrics can distinguish broadly faithful from unfaithful text (AlignScore ROC-AUC = 0.849) but cannot match our framework's ability to discriminate fine-grained error levels or decompose errors into substitutions, deletions, and insertions.

We note two design decisions that we believe contribute to our results. First, the domain hint map is a lightweight abstraction that provides the extraction model with structured guidance about each domain's schema without requiring extraction rules. Writing a hint map for a new domain takes minutes rather than the hours required to engineer a rule-based evaluator. Second, the use of a separate ranking model rather than a self-scoring approach allows the extraction and selection tasks to be optimized independently, which mirrors findings in the generation literature where separating production from evaluation produces better results (Ramirez et al., 2023).

We acknowledge the model vintage of our experiments. Llama-3.2-3B, ChatGPT 4o, and RoBERTa-MNLI were selected as the best available options at the time of experimentation. The LoRA + OGR + routing methodology is model-agnostic, and we expect that applying it to newer small models would yield comparable or improved results. We also tested DeBERTa-v3-large-MNLI as a replacement for RoBERTa-MNLI and found that it performed substantially worse on slot-level verification ($\text{All}_{\text{acc}} = 0.479$ vs. 0.805), likely due to poor calibration of its entailment probabilities for this task. We interpret this as evidence that newer NLI models do not automatically improve slot-level evaluation and that task-specific calibration matters. We believe the key finding is methodological. Task-specific adaptation of a small model, combined with overgenerate-and-rank and method routing, outperforms prompting a much larger model for structured evaluation tasks.

7 Conclusion

We presented a cross-domain SER evaluation framework that replaces hand-crafted rules with a learned slot extraction pipeline combining LoRA adaptation, overgenerate-and-rank, and NLI-based routing. Gradient-boosted routing with cost-sensitive weighting achieves 90.0% accuracy across 23 dialogue domains without per-domain engineering, capturing 87% of the gap to oracle performance. Our framework matches or outperforms published rule-based tools on all six comparable domains. We show that evaluation, like generation, benefits from the combination of task-specific adaptation, diverse candidate production, and learned selection. We release our code and evaluation data to support further research on cross-domain NLG evaluation.[4]

Limitations

We identify four main limitations. First, our evaluation covers only English-language, task-oriented dialogue domains. We do not know how well the approach generalizes to other languages, open-domain settings, or more complex structured representations such as knowledge graphs. Second, we have not conducted a human evaluation of extraction quality, and our evaluation relies on automatic comparison to gold MRs and synthetically constructed error pairs. Third, the models used

[4]Code and data will be made available at `github.com/Vrindiesel/xdomain-ser`.

(Llama-3.2-3B, ChatGPT 4o, RoBERTa-MNLI) represent a specific point in time. While we expect the methodology to transfer to newer models, we have not verified this empirically. Fourth, even with our best routing method, overall accuracy is 90.0%, meaning roughly 10% of SER judgments are incorrect, with errors concentrated on domains with complex schemas and large MRs.

Ethics Statement

This work focuses on evaluation methodology for meaning-to-text generation using publicly available datasets. We do not collect or process personally identifiable information. The datasets used (E2E, RNNLG, Taskmaster, ViGGO) are publicly released for research purposes. Our use of the OpenAI API for ChatGPT experiments followed their terms of service. We note that automated evaluation tools can be misused to falsely certify the semantic accuracy of generated text. Our system should be used as one component of a broader evaluation strategy rather than as a sole arbiter of output quality.

References

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. MultiWOZ – a large-scale multi-domain Wizard-of-Oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026.

Bill Byrne, Karthik Krishnamoorthi, Chinnadhurai Sankar, Arvind Neelakantan, Daniel Duckworth, Semih Yavuz, Ben Goodrich, Amit Dubey, Kyu-Young Kim, and Andy Cedilnik. 2019. Taskmaster-1: Toward a realistic and diverse dialog dataset. In *2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing*, Hong Kong.

Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794.

Ondřej Dušek and Zdeněk Kasner. 2020. Evaluating semantic accuracy of data-to-text generation with natural language inference. In *Proceedings of the 13th International Conference on Natural Language Generation (INLG)*, pages 131–137. Association for Computational Linguistics.

Mingqi Gao, Xinlu Hu, Jie Ruan, Xiao Pu, and Xiaojun Wan. 2025. LLM-based NLG evaluation: Current status and challenges. *Computational Linguistics*, 51(2):661–687.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, and 1 others. 2024. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

Rudali Huidrom, Ondřej Dušek, and Zdeněk Kasner. 2022. Two reproductions of a human-assessed comparative evaluation of a semantic error detection system. In *Proceedings of the 15th International Conference on Natural Language Generation: Generation Challenges (INLG)*, pages 77–83, Waterville, Maine, USA. Association for Computational Linguistics.

Juraj Juraska. 2022. *Diversifying Language Generated by Deep Learning Models in Dialogue Systems*. Ph.D. thesis, UC Santa Cruz.

Juraj Juraska, Kevin Bowden, and Marilyn Walker. 2019. ViGGO: A video game corpus for data-to-text generation in open-domain conversation. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 164–172, Tokyo, Japan. Association for Computational Linguistics.

Damjan Kalajdzievski. 2023. A rank stabilization scaling factor for fine-tuning with lora. *arXiv preprint arXiv:2312.03732*.

Philippe Laban, Tobias Schnabel, Paul N. Bennett, and Marti A. Hearst. 2022. SummaC: Re-visiting NLI-based models for inconsistency detection in summarization. *Transactions of the Association for Computational Linguistics*, 10:163–177.

Jinghong Lee, Evgeniia Razumovskaia, Arthur Guez, Tom Sherborne, Igor Shalyminov, and Arash Eshghi. 2023. Schema-guided semantic accuracy: Faithfulness in task-oriented dialogue response generation. *arXiv preprint arXiv:2301.12568*.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: Nlg evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained atomic evaluation of factual

precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 12076–12100, Singapore. Association for Computational Linguistics.

Jekaterina Novikova, Ondřej Dušek, and Verena Rieser. 2017. The e2e dataset: New challenges for end-to-end generation. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 201–206, Saarbrücken, Germany. Association for Computational Linguistics.

OpenAI. 2024. GPT-4o system card. *arXiv preprint arXiv:2410.21276*.

Shereen Oraby, Lena Reed, Shubhangi Tandon, T.S. Sharath, Stephanie Lukin, and Marilyn Walker. 2018. Controlling personality-based stylistic variation with neural natural language generators. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pages 180–190, Melbourne, Australia. Association for Computational Linguistics.

Fabian Paischer, Lukas Hauzenberger, Thomas Schmied, Benedikt Alkin, Marc Peter Deisenroth, and Sepp Hochreiter. 2024. Parameter efficient fine-tuning via explained variance adaptation. *arXiv preprint arXiv:2410.07170*.

Angela Ramirez, Mamon Alsalihy, Kartik Aggarwal, Cecilia Li, Liren Wu, and Marilyn Walker. 2023. Controlling personality style in dialogue with zero-shot prompt-based learning. *arXiv preprint arXiv:2302.03848*.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sber, George Guez, and Georges Suber. 2020. Towards scalable multi-domain conversational agents: The Schema-Guided Dialogue dataset. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8689–8696.

Clément Rebuffel, Thomas Scialom, Laure Soulier, Benjamin Piwowarski, Sylvain Lamprier, Jacopo Staiano, Geoffrey Scoutheeten, and Patrick Gallinari. 2021. Data-QuestEval: A referenceless metric for data-to-text semantic evaluation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8795–8806, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Oscar Sainz, Iker García de Lacalle, Oier López de Lacalle, Gorka Labaka, and Eneko Agirre. 2024. GoLLIE: Annotation guidelines improve zero-shot information-extraction. In *Proceedings of the Twelfth International Conference on Learning Representations (ICLR)*.

Liyan Tang, Philippe Laban, and Greg Durrett. 2024. MiniCheck: Efficient fact-checking of LLMs on grounding documents. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8818–8847, Miami, Florida, USA. Association for Computational Linguistics.

Kai Wei, Rishabh Iyer, and Jeff Bilmes. 2014. Fast multi-stage submodular maximization. In *Proceedings of the 31st International Conference on Machine Learning*, pages 1494–1502. PMLR.

Tsung-Hsien Wen, Milica Gašić, Nikola Mrkšić, Lina M. Rojas-Barahona, Pei-Hao Su, David Vandyke, and Steve Young. 2016. Multi-domain neural network language generation for spoken dialogue systems. In *Proceedings of the 2016 Conference on North American Chapter of the Association for Computational Linguistics (NAACL)*. Association for Computational Linguistics.

Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. *arXiv preprint arXiv:1508.01745*.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Yuheng Zha, Yichi Yang, Ruichen Li, and Zhiting Hu. 2023. AlignScore: Evaluating factual consistency with a unified alignment function. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11328–11348, Toronto, Canada. Association for Computational Linguistics.

Wenxuan Zhou, Sheng Jiang, Le Cui, Lifeng Wang, Xiangyao Xia, and Muhao Lee. 2024. UniversalNER: Targeted distillation from large language models for open named entity recognition. In *Proceedings of the Twelfth International Conference on Learning Representations (ICLR)*.

A Dataset Statistics

Table 8 provides per-topic statistics for the full corpus. We use 20 topics for training and 12 for testing (9 in-domain, 3 out-of-domain). The out-of-domain topics (ViGGO, hotel, laptop) were never seen during training and test cross-domain generalization.

Topic	Turns	#Slots	Avg Tok	Avg MR	|V|
Training Topics					
tm1_auto_repair_appt	3,548	10	18.9	2.3	2,533
tm1_coffee_ordering	6,252	8	15.3	2.2	2,561
tm1_movie_tickets	22,558	25	15.9	2.2	149
tm1_pizza_ordering	6,335	9	17.1	2.6	2,860
tm1_restaurant_table	9,027	8	17.1	1.8	3,312
tm1_uber_lyft	7,634	10	14.8	1.8	2,995
tm2_flights	14,394	31	14.1	2.3	8,591
tm2_food_ordering	3,212	13	16.8	2.0	2,550
tm2_music	6,131	11	10.6	1.9	5,839
tm2_restaurant_search	13,147	18	15.1	2.2	11,925
tm2_sports (5 topics)	15,389	17–18	10.1–12.7	1.1–2.7	644–2,353
tm3_Movie_Tickets	116,926	21	20.3	3.1	40,624
e2e_nlg	42,061	8	21.7	5.4	77
rnnlg_tv	4,202	16	21.7	4.8	287
rnnlg_restaurant	1,923	13	11.2	2.6	701
In-Domain Test Topics					
tm1_auto_repair_appt	393	10	17.8	2.1	387
tm1_coffee_ordering	439	8	15.0	2.0	420
tm1_movie_tickets	633	10	15.9	2.1	14
tm1_pizza_ordering	458	9	16.9	2.4	452
tm1_restaurant_table	603	8	16.6	2.0	578
tm1_uber_lyft	531	10	14.2	1.8	546
e2e_nlg	4,693	8	26.5	6.9	56
rnnlg_tv	1,401	16	21.7	4.8	282
rnnlg_restaurant	669	13	11.2	2.7	435
Out-of-Domain Test Topics					
viggo (video games)	1,026	13	24.6	4.8	247
rnnlg_hotel	563	12	11.5	2.5	262
rnnlg_laptop	2,637	20	23.2	4.9	414

Table 8: Per-topic dataset statistics. Turns = agent utterances; #Slots = unique slot types; Avg Tok = mean tokens per utterance; Avg MR = mean slot-value pairs per utterance; |V| = unique slot values. The five TM2 sports topics are collapsed into one row for space.

B LoRA Rank Analysis

We vary the LoRA rank $r \in \{2, 4, 8, 16\}$ to assess how much adapter capacity the extraction task requires. Table 9 shows results. All four configurations achieve micro F1 in the range 0.819–0.823 and SER-Accuracy of 0.75–0.76 on the aggregate row. We find that very low ranks suffice for cross-domain slot extraction, and the task requires a relatively low-dimensional adaptation. We adopt $r = 4$ for all main experiments.

C Training Data Quantity

We compare models trained with $K \in \{100, 200, 400\}$ examples per topic, all selected using the Facility-Location objective.

Rank	F1	SER-Acc	P	R
$r = 2$	.821	.75	.853	.790
$r = 4$	.819	.75	.845	.795
$r = 8$	.823	.76	.841	.806
$r = 16$	.823	.76	.848	.799

Table 9: Extraction performance by LoRA rank (micro-averaged over all test topics). All ranks yield near-identical performance.

Table 10 shows that $K = 200$ produces the best overall performance. The improvement from 100 to 200 is driven primarily by recall gains (0.776 $\rightarrow$ 0.795). Performance at $K = 400$ slightly regresses, particularly on OOD topics (ViGGO SER-Acc drops from 0.54 to 0.50), suggesting mild overfitting to training-domain patterns.

K per topic	F1	SER-Acc	P	R
100	.806	.73	.839	.776
200	.819	.75	.845	.795
400	.802	.73	.824	.781

Table 10: Extraction performance by training set size (micro-averaged). $K = 200$ per topic produces the best trade-off between coverage and overfitting.

D Cross-Domain Validation on Personality Outputs

We evaluate all five SER methods on 1,000 gold-annotated personality-conditioned restaurant-domain outputs, with 500 from GPT-4o LLM outputs across five Big Five personality types and 500 from PERSONAGE seq2seq outputs. Each example was manually annotated with gold SER counts for substitutions, deletions, and insertions. Table 11 presents results on the 500-example test split.

Score-threshold routing achieves the best overall accuracy of 82.6%, outperforming both NLI (76.0%) and LoRA (70.8%) individually. The oracle accuracy of 90.8% indicates strong complementarity between the two methods. The routing improvements are concentrated on LLM outputs, where score-threshold routing reaches 92.8%. On seq2seq outputs, the gains are more modest (LR-Routing at 73.6% vs. NLI at 72.0%).

E Prompt Templates

Extraction prompt (PBL and LoRA). Figure 2 shows the extraction prompt template used for both the PBL baselines and the LoRA-adapted model. The model receives a domain schema with

Method	All_{acc}	MAE	LLM	Seq2seq
Aligner	.636	.059	.788	.484
LoRA	.708	.044	.804	.612
NLI	.760	.034	.800	.720
Score Routing	**.826**	**.024**	**.928**	.724
LR Routing	.798	.028	.860	**.736**
Oracle	.908	.014	.956	.860

Table 11: Five-way SER method comparison on 1,000 gold-annotated personality-conditioned outputs (500-example test split). LLM/Seq2seq columns show All_{acc} by source (250 examples each).

slot descriptions, 5 in-domain exemplars showing utterance-MR pairs, and the input utterance.

```
### Instruction:
Rewrite the input text content as a
list of (attribute: value) pairs.
Here is a hint map for {TOPIC}
attributes and values:
{HINT_MAP}
Here are some examples:
{PROMPT_EXAMPLES}

### Input:
{UTTERANCE_TEXT}
### Response:
```

Figure 2: Extraction prompt template. `HINT_MAP` is the domain schema, `PROMPT_EXAMPLES` contains 5 in-domain exemplars, and `UTTERANCE_TEXT` is the input to extract from.

Ranking prompt. Figure 3 shows the ranking model prompt. The model receives the domain schema, surface text, gold reference MR, and the candidate predicted MR, and outputs a single digit (0–6) indicating extraction quality.

Domain hint maps. Figure 4 shows hint maps for three representative domains. Each hint map lists slot types with descriptions and example values.

F NLI Template Details

We design NLI templates for all 133 slot types across our evaluation domains. Templates fall into three categories:

Value-substitution templates (majority of slots) use a simple declarative pattern. For example, `name` maps to "The name is {value}." and `priceRange` maps to "The price range is {value}."

```
You are scoring how well a predicted
meaning representation (MR) matches
the reference text and gold MR.
Return only one digit: 0=bad,
1=poor, 2=weak, 3=mediocre, 4=good,
5=excellent, 6=perfect.

DOMAIN SCHEMA (allowed slots) &
description:
{HINT_MAP}

Text: {TEXT}
Gold Reference: {REFERENCE}
Predicted MR: {MR}

Score:
```

Figure 3: Ranking model prompt template. The model is trained to output a single digit (0–6) indicating how well the predicted MR matches the gold.

Boolean templates produce distinct hypotheses for positive and negative values. For example, `familyFriendly=yes` → "It is family-friendly." and `familyFriendly=no` → "It is not family-friendly."

Conditional templates handle domain-specific semantics. For example, the ViGGO `multiplayer_mode` slot maps "multiplayer" to "It has multiplayer." and other values to "It is single-player."

We sweep the entailment threshold $\tau \in \{0.3, 0.4, 0.5, 0.6, 0.7\}$ and find that $\tau = 0.3$ produces the best overall accuracy (All_{acc} = 0.817 on the full dataset). Performance degrades monotonically as the threshold increases, reflecting the fact that RoBERTa-MNLI tends to assign moderate probabilities to entailed pairs in this domain.

G Evaluation Pair Construction

We construct evaluation pairs for the SER Agreement evaluation (Evaluation 2) by pairing each surface text with modified MRs that differ from the gold in known ways. Starting from 2,288 test examples, we create modified MRs by substituting alternative gold MRs from the same topic domain at controlled distances. For each modified MR, we compute ground-truth error counts for substitutions (S), deletions (D), and insertions (I) deterministically from the known divergence between the modified and gold MRs.

We bin pairs into five difficulty levels based on SER-Accuracy: label 4 (≥ 0.90, near-perfect), label 3 (0.75–0.90), label 2 (0.50–0.75), label 1 (0.25–0.50), and label 0 (< 0.25, heavily distorted). The resulting dataset contains 9,042 pairs across 12 test

(a) E2E NLG

```
name: restaurant name (e.g., "The Waterman")
priceRange: price range (e.g., cheap, moderate)
customerRating: customer rating (low, average, high)
area_zone: area (e.g., city centre, riverside)
family_suitability: family-friendly / not-family-friendly
nearby_landmark: nearby landmark (free text)
cuisine_type: cuisine (e.g., English, Chinese)
venue_type: venue type (restaurant, pub, coffee shop)
```

(b) ViGGO (Video Games)

```
name: game title
review_rating: rating (excellent/good/average/poor)
genres: genres (e.g., action-adventure, shooter)
platforms: platform (PC, PlayStation, Xbox, Nintendo)
steam_availability: on_steam / not_on_steam
multiplayer_mode: single-player / multiplayer / both
developer: developer name
```

... (13 slots total)

Figure 4: Domain hint maps for two representative domains. The full system covers 23 domains with hint maps for all slot types.

topics. Table 12 summarizes the error characteristics by difficulty level.

Label	Quality	n	Mean S	Mean D	Mean I
4	Near-perfect	2,288	0.00	0.00	0.00
3	Minor errors	865	0.54	0.44	0.09
2	Moderate errors	2,174	0.91	0.42	0.22
1	Major errors	1,427	1.56	1.10	0.33
0	Heavily distorted	2,288	1.52	1.75	1.64

Table 12: Ground-truth error characteristics of the SER Agreement evaluation pairs by difficulty level.

By holding the text fixed and varying only the MR, we create pairs with precisely controlled error profiles without requiring error-text generation. We test the same computation the system performs in deployment, quantifying the discrepancy between a fixed text and a reference MR.

H Per-Topic SER Agreement

Table 13 provides the full per-topic breakdown of SER Agreement results on the test split for all four learned methods and, where available, published rule-based tools. The six topics with rule-based baselines are shown in the main paper (Table 6). Here we include the remaining six topics that lack rule-based comparisons.

Topic	Method	S	D	I	All	MAE
auto_repair	LoRA	.850	.937	.889	.761	.094
	NLI	.997	.964	.994	**.958**	.016
	Score Rt	.949	.985	.985	.937	.027
	LR Rt	.934	.988	.988	.928	.032
coffee	LoRA	.892	.952	.911	.840	.063
	NLI	.900	.807	.970	.773	.044
	Score Rt	.955	.970	.981	**.926**	.018
	LR Rt	.952	.959	.978	.918	.023
movie_tickets	LoRA	.887	.935	.952	.834	.065
	NLI	.949	.907	.969	.862	.054
	Score Rt	.944	.972	.989	**.924**	.029
	LR Rt	.941	.978	.983	.924	.035
pizza	LoRA	.831	.859	.869	.703	.115
	NLI	.869	.763	.934	.681	.119
	Score Rt	.934	.934	.944	**.841**	.047
	LR Rt	.919	.941	.934	.847	.053
rest._table	LoRA	.872	.946	.923	.801	.089
	NLI	.940	.920	.960	.860	.059
	Score Rt	.940	.977	.983	.912	.035
	LR Rt	.957	.972	.969	**.926**	.040
uber_lyft	LoRA	.895	.955	.940	.847	.081
	NLI	.965	.940	.997	.933	.019
	Score Rt	.975	.990	.997	**.968**	.014
	LR Rt	.959	.990	.997	.952	.022

Table 13: Per-topic SER Agreement for the six topics without rule-based baselines. Best All-correct per topic in **bold**. Routing improves over both baselines on all topics.

I Extraction Model Details

We use Llama-3.2-3B-Instruct as the foundation model for both the extraction model and the ranking model. Using LoRA, we update 6,078,464 of 1,809,542,144 parameters (0.34%). We target all attention projections (`q_proj`, `k_proj`, `v_proj`, `o_proj`) and MLP projections (`gate_proj`, `up_proj`, `down_proj`).

We initialize adapters using Explained Variance Adaptation (Paischer et al., 2024), which bases initialization on the SVD of layer input activations. We apply Rank-Stabilized scaling (Kalajdzievski, 2023), which sets the adapter scaling factor to $\alpha/\sqrt{r}$ rather than α/r.

We train with QLoRA using 4-bit quantization via `bitsandbytes`, a learning rate of 5×10^{-4}, gradient accumulation over 64 steps, per-device batch size of 2, and 3 training epochs. The extraction and ranking models use separate LoRA adapters but share the same frozen base model.