Beyond Supervised Clarification: Input Rewriting with LLMs for Dialogue Discourse Parsing

Yiming Liu[1], Ziyue Zhang[1], Zhichao Xu[2], Xin Yu[2], Yingheng Tang[3], Tianyu Jiang[4], Jie Cao[1]

[1]University of Oklahoma

[2]University of Utah

[3]Lawrence Berkeley National Laboratory

[4]University of Cincinnati

GALLOGLY COLLEGE OF ENGINEERING
SCHOOL OF COMPUTER SCIENCE
The UNIVERSITY of OKLAHOMA

OUNLP
OU NATURAL LANGUAGE PROCESSING LAB

2026 SIGdial Atlanta, GA, USA

Task: incremental SDRT dialogue discourse parsing

u1 | Shawnus: i can trade wood

u2 | ztime: just spent it all

u3 | ztime: sorry

u4 | somdechn: 1 sheep for 1 wood?

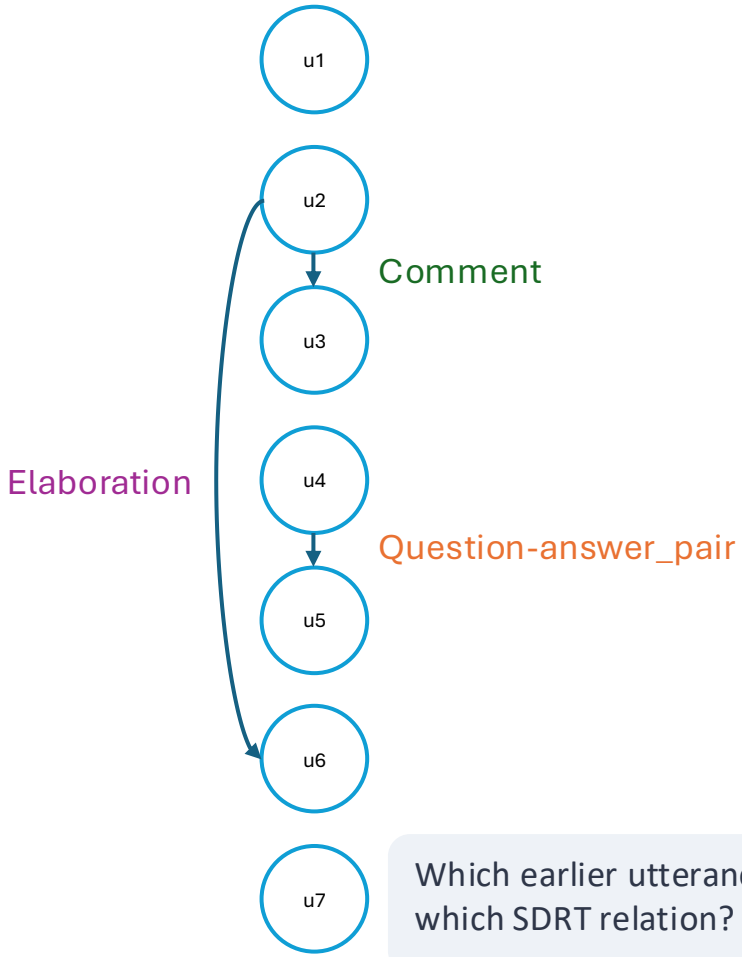
u5 | Shawnus: 2 sheep 1 wood

u6 | ztime: sorry empty

u7 | ztime: tough times..

Parser prediction: u6, u7: Comment

Gold prediction: u6, u7: Explanation



Which earlier utterance does u7 attach to, and with which SDRT relation?

Clarify the last utterance

u1 | Shawnus: i can trade wood

u2 | ztime: just spent it all

u3 | ztime: sorry

u4 | somdechn: 1 sheep for 1 wood?

u5 | Shawnus: 2 sheep 1 wood

u6 | ztime: sorry empty

u7 | ztime: tough times..

Parser prediction: u6, u7: Comment

Gold prediction: u6, u7: Explanation

Ambiguity

- Just commiserating (*Comment*)
- Saying why he is empty (*Explanation*)

Clarification 1: So tough times..

Clarification 2: tough times, sorry empty

Clarification 3: I'm out of resources.

Motivation

- ***Rewriting the input to a frozen model*** is a pattern that works well in many parts of NLP, even more in the tool usage of agentic AI

1 Machine translation

Rewrite the source sentence
before an unchanged MT system

Ki & Carpuat, 2025

2 Retrieval & RAG

Rewrite the query so the
frozen retriever finds better evidence

Ma et al., 2023

3 Conversational search

Rewrite the user turn to be
self-contained across the session

Zhu et al., 2025

- Same architecture: an LLM front-end rewriting the input to a frozen component it cannot retrain

Background

- Fan et al. (2025) train a discourse-aware clarifier to resolve ambiguous utterances before parsing

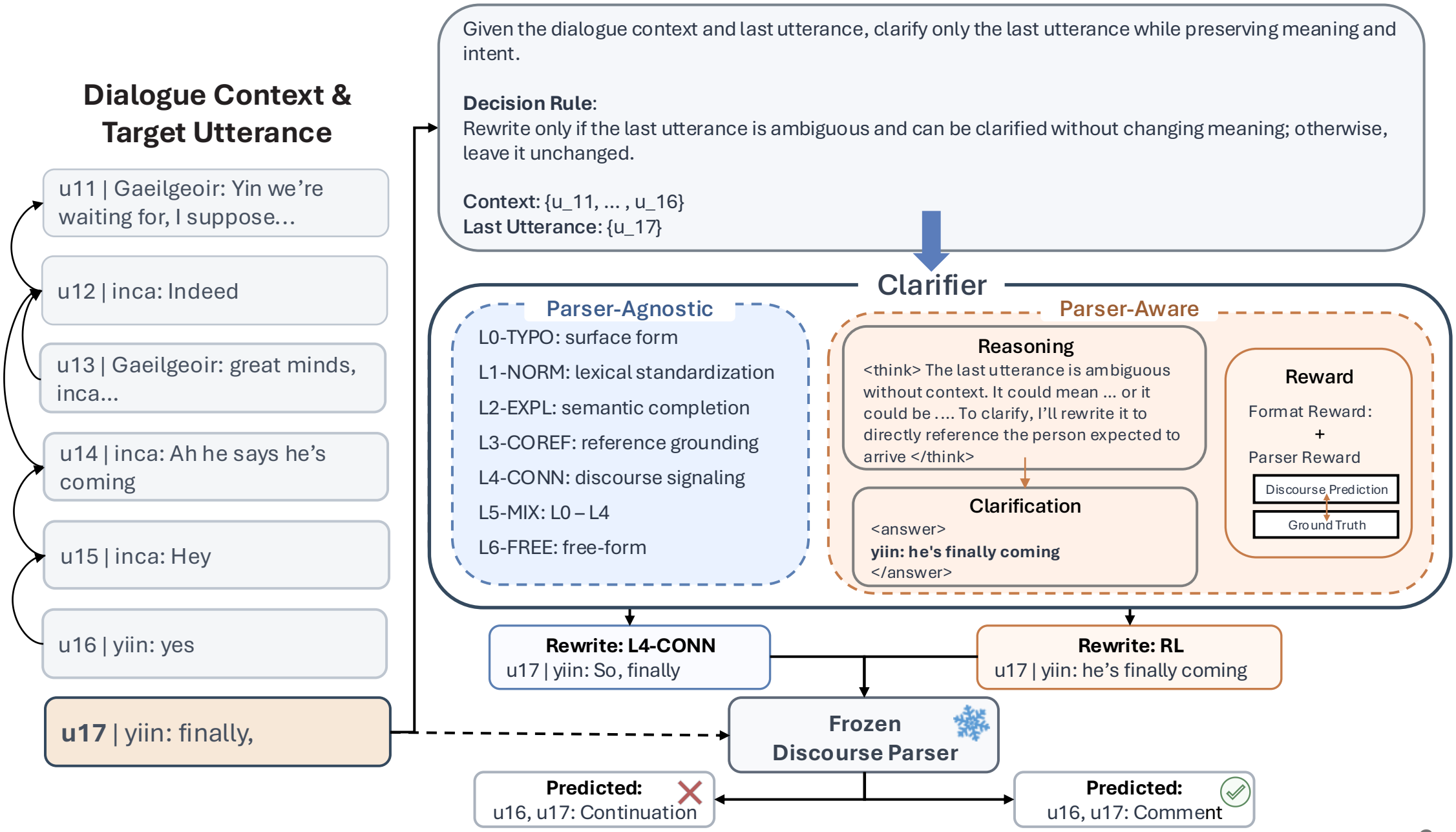
What they show

- Trained on supervised clarification pairs, then tuned with preference-based RL.
- It works — real gains on dialogue discourse parsing.

What it assumes

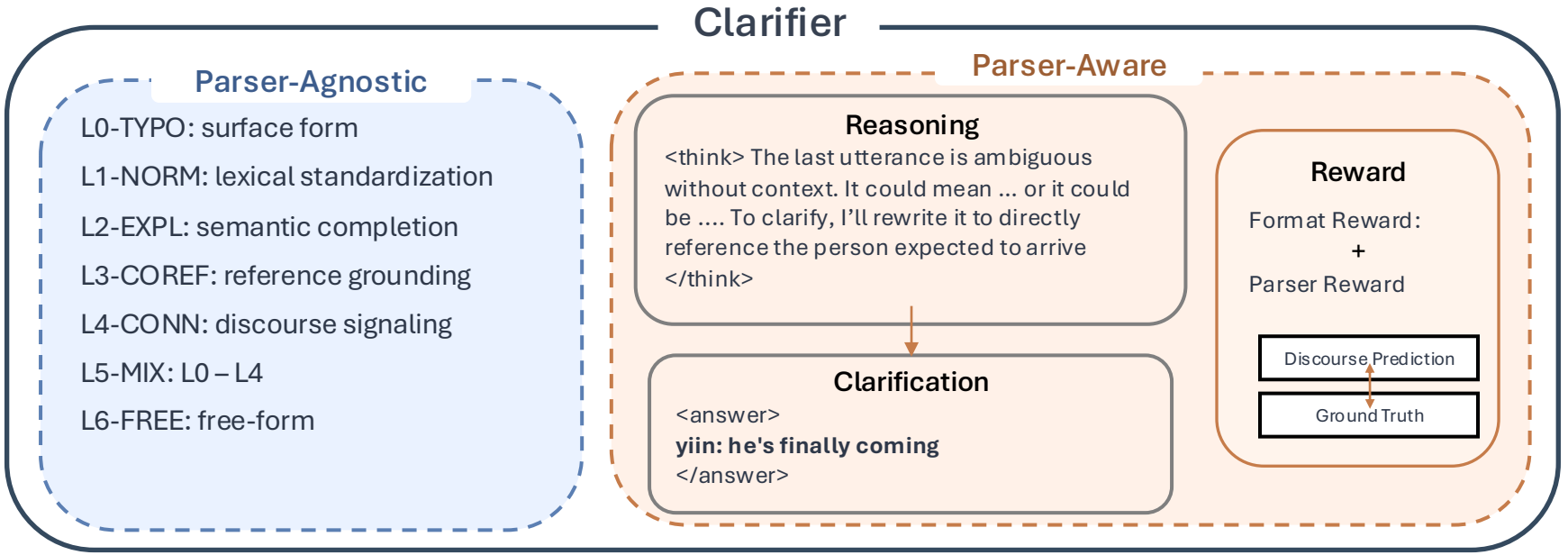
- GPT-4 generates clarification pairs using gold discourse labels and parser-induced ambiguities.
- Preference supervision is parser-specific and may need regeneration for a new parser.

Can last-utterance clarification improve Dialogue Discourse Parsing without supervised clarification examples?


Dialogue Context & Target Utterance
u11 | Gaeilgeoir: Yin we're waiting for, I suppose...
u12 | inca: Indeed
u13 | Gaeilgeoir: great minds, inca...
u14 | inca: Ah he says he's coming
u15 | inca: Hey
u16 | yiin: yes
u17 | yiin: finally,
Given the dialogue context and last utterance, clarify only the last utterance while preserving meaning and intent.
Decision Rule:
Rewrite only if the last utterance is ambiguous and can be clarified without changing meaning; otherwise, leave it unchanged.
Context: {u_11, ... , u_16}
Last Utterance: {u_17}
Clarifier
Parser-Agnostic
L0-TYPO: surface form
L1-NORM: lexical standardization
L2-EXPL: semantic completion
L3-COREF: reference grounding
L4-CONN: discourse signaling
L5-MIX: L0 – L4
L6-FREE: free-form
Parser-Aware
Reasoning
<think> The last utterance is ambiguous without context. It could mean ... or it could be .... To clarify, I'll rewrite it to directly reference the person expected to arrive </think>
Clarification
<answer>
yiin: he's finally coming
</answer>
Reward
Format Reward:
+
Parser Reward
Discourse Prediction
Ground Truth
Rewrite: L4-CONN
u17 | yiin: So, finally
Rewrite: RL
u17 | yiin: he's finally coming
Frozen Discourse Parser
Predicted:
u16, u17: Continuation
Predicted:
u16, u17: Comment

Clarifier: Two settings, one intervention point

Both operate without any clarification supervision



Experimental Setup: Datasets

STAC (Asher et al., 2016):

- online *Settlers of Catan* game chats
- 9507 / 1084 / 1045

Molweni (Li et al., 2020):

- Ubuntu Chat Corpus (technical support forum)
- 9215 / 1026 / 1107

MSDC (Thompson et al., 2024):

- Minecraft situated, task-oriented dialogues
- 1732 / 1016 / 989

Experimental Setup: Models

2 Parsers

- Generative approaches: Qwen3-8B / 14B (Yang et al., 2025)
- Discriminative approaches: SDDP (Chi and Rudnicky, 2022)
  - BERT encoder + biaffine, global decoding

2 Clarifiers

- Parser-agnostic: Qwen3-8B prompted
- Parser-aware: Qwen2.5-7B-Instruct for RL
  - due to compatibility constraints for our training framework

Experimental Setup: Evaluation Metrics

Micro-F1

- link
- link + relation

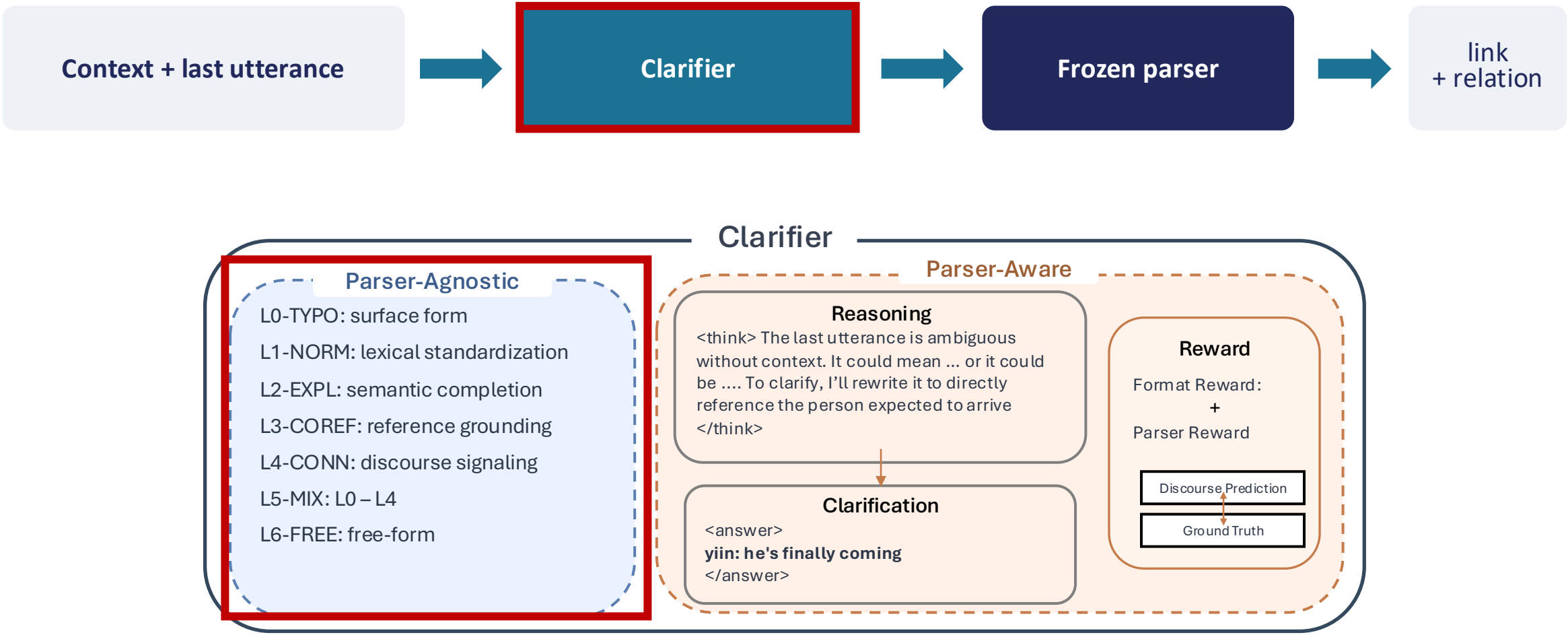
Repairs

- cases where the parser was wrong on the original utterance and correct on the rewrite

Regressions

- cases where the parser was correct on the original utterance and wrong on the rewrite

Experiments: Parser-Agnostic Clarifier



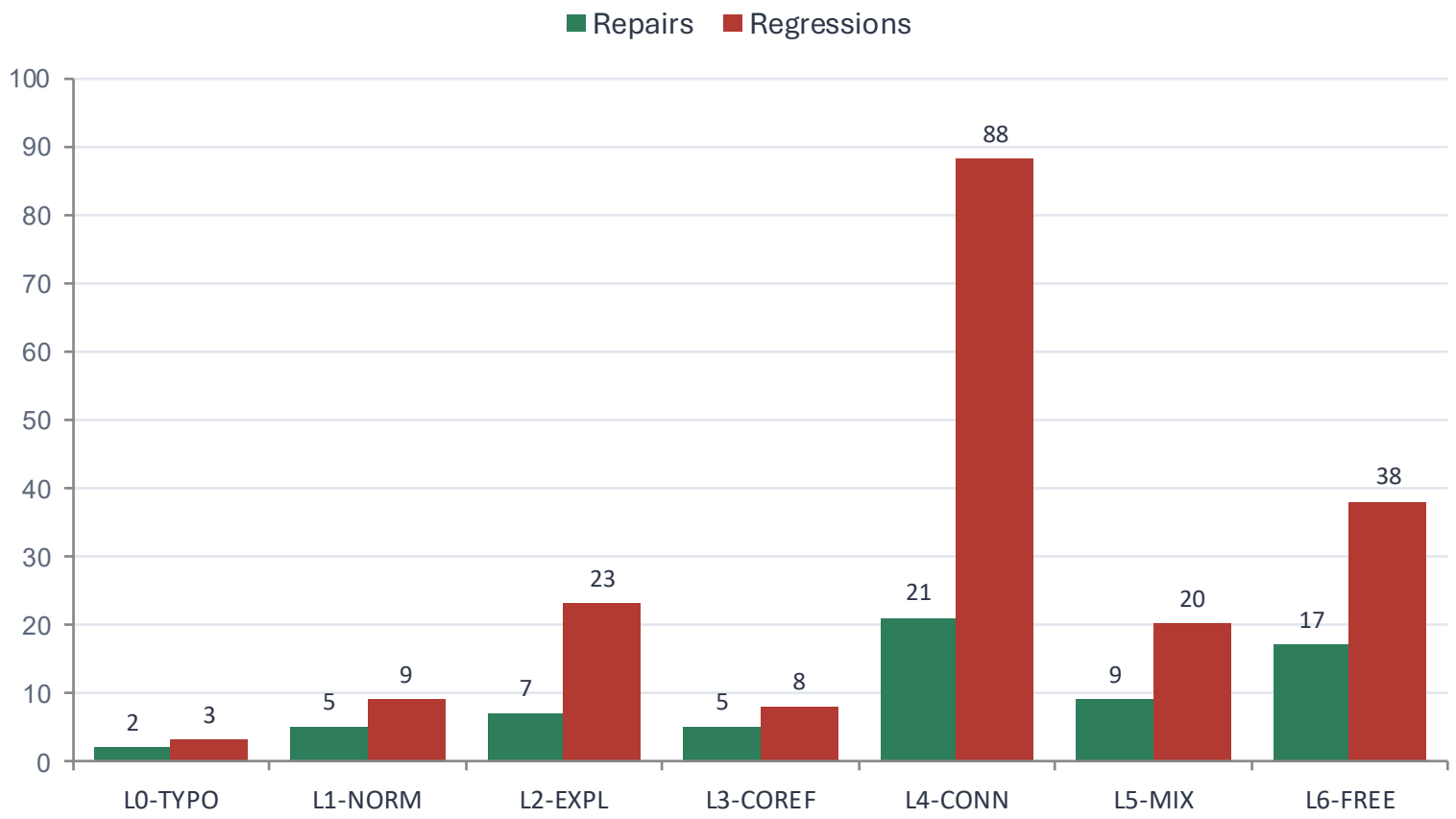
Seven strategies, increasing depth of intervention

From edits that cannot plausibly disturb the parser, to edits that target discourse structure directly

L0	TYPO	**Surface form** ***(Ki and Carpuat, 2025)***	high-confidence typo fixes only	*"peurto rico" → "Puerto Rico"*
L1	NORM	**Lexical standardization** ***(Bitton et al., 2022)***	expand shorthand and slang	*"idk" → "I don't know"*
L2	EXPL	**Semantic completion** ***(Li et al., 2023)***	restore context-entailed ellipsis	*"tomorrow" → "I'll do it tomorrow"*
L3	COREF	**Reference grounding** ***(Choi et al., 2021)***	resolve pronouns and deictics	*"it" → "the wheat"*
L4	CONN	**Discourse signaling** ***(Aktas and Roth, 2025)***	insert one explicit connective	*"finally," → "So, finally,"*
L5	MIX	**All five rules available**	apply any safe subset of L0–L4	—
L6	FREE	**Free-form clarification**	minimally constrained rewriting	—

Finding 1: rewriting breaks more than it fixes

STAC, Qwen3-8B parser — no strategy is net positive



7 / 7

strategies produce more regressions than repairs — on all three datasets, both LLM parsers, and SDDP.

Link + Rel F1, STAC

no clarification	**0.598**
L0-TYPO	**0.596**
L6-FREE	**0.577**
L4-CONN	**0.540**

Doing nothing beats parser-agnostic clarification strategies we tried.

Finding 1: rewriting breaks more than it fixes

- Generative parsers: parser-agnostic clarification introduces more regressions than repairs

Method	STAC		Molweni		MSDC	
	F1	Fix/Reg	F1	Fix/Reg	F1	Fix/Reg
Llamipa	0.577	-	-	-	0.795	-
Qwen3-8B	0.598	-	0.571	-	0.800	-
L0-TYPO	**0.596**	**2/3**	0.565	28/51	**0.800**	**11/11**
L1-NORM	0.594	5/9	0.564	31/59	0.797	12/24
L2-EXPL	0.582	7/23	**0.568**	**33/47**	0.795	24/48
L3-COREF	0.594	5/8	0.556	38/96	0.794	14/40
L4-CONN	0.540	21/88	0.531	74/234	0.771	61/212
L5-MIX	0.589	9/20	0.555	50/114	0.793	25/56
L6-FREE	0.577	17/38	0.565	104/128	0.789	43/105
Qwen3-14B	0.591	-	0.554	-	0.805	-
L0-TYPO	**0.589**	**2/4**	0.552	17/27	0.803	2/10
L1-NORM	0.585	6/12	**0.552**	**34/42**	0.804	9/12
L2-EXPL	0.576	10/26	0.551	36/51	**0.799**	**21/47**
L3-COREF	0.587	3/7	0.538	30/94	0.799	12/45
L4-CONN	0.532	22/89	0.512	73/239	0.779	60/192
L5-MIX	0.584	9/17	0.549	49/74	0.799	13/44
L6-FREE	0.567	16/44	0.549	90/111	0.797	40/84

Finding 1: rewriting breaks more than it fixes

- Discriminative parser — different architecture, same pattern

Method	STAC		Molweni		MSDC	
	F1	Fix/Reg	F1	Fix/Reg	F1	Fix/Reg
SDDP	0.557	–	0.543	–	0.696	–
L0-Typo	**0.552**	**0/4**	**0.537**	**16/40**	**0.696**	**11/18**
L1-Norm	0.551	6/10	0.536	18/46	0.696	13/20
L2-Expl	0.539	9/27	0.535	22/54	0.695	8/24
L3-Coref	0.549	3/13	0.529	22/75	0.694	19/37
L4-Conn	0.497	19/81	0.522	76/158	0.688	20/79
L5-Mix	0.547	10/22	0.530	45/96	0.694	12/32
L6-Free	0.525	17/47	0.533	74/111	0.693	19/43

Finding 2: about a quarter of errors are repairable by rewriting

Best-of-8 free-form rewrites per error: is the error fixable by ANY of them?

		STAC		Molweni		MSDC	
Category	**Succ. (of 8)**	**# Ex.**	**%**	**# Ex.**	**%**	**# Ex.**	**%**
Not repairable	0	475	80.0	1173	72.5	265	71.8
Rarely repairable	1–2	66	11.1	182	11.2	35	9.5
Moderately repairable	3–5	39	6.6	147	9.1	52	14.1
Highly repairable	6–8	14	2.4	116	7.2	17	4.6

Table 4: Distribution of originally incorrect examples on validation set by estimated repairability under a best-of-8 free-form rewriting baseline. Repairability is defined by how many of eight sampled rewrites, generated from the same free-form prompt, yield a correct prediction.

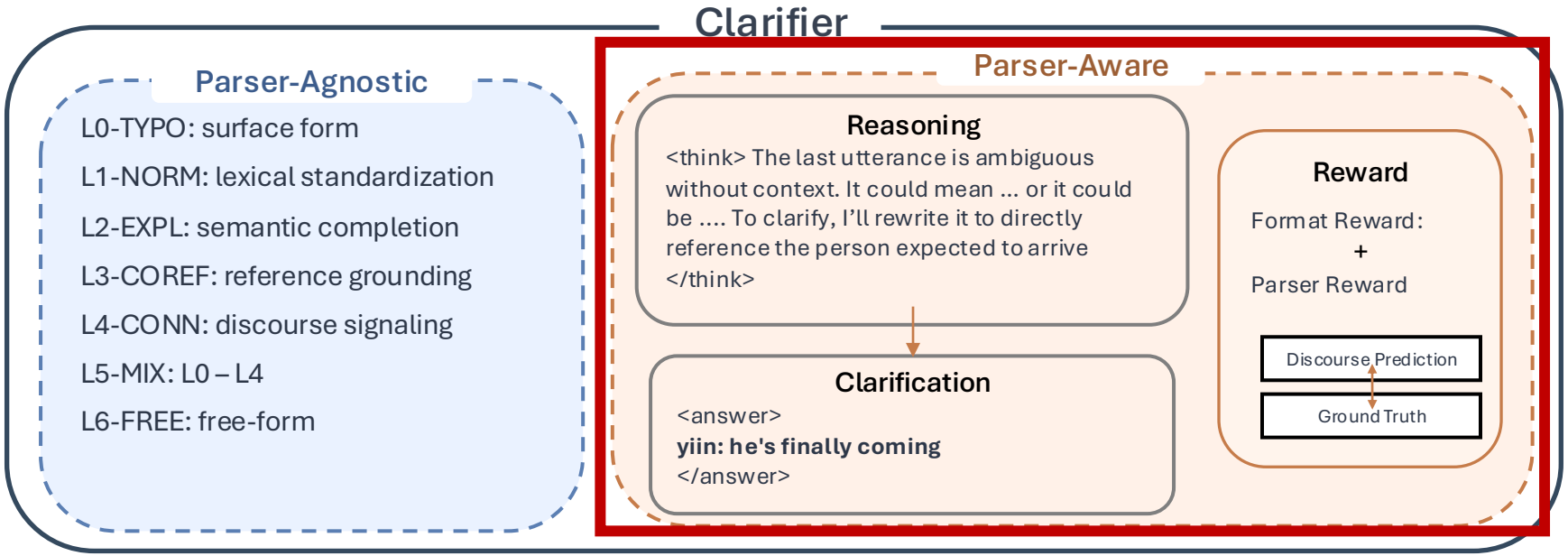
20–28%

of parser errors are fixed by at least one of eight independent free-form rewrites of the last utterance.

Two readings

- Real headroom exists — but bounded: the other 72–80% is unreachable by last-utterance rewriting.
- Repairability is a spectrum — the gain depends on finding the right instances.

Experiments: Parser-Aware Clarifier



Can the clarifier learn when to intervene?

GRPO against frozen-parser feedback — still no clarification labels

1. **Policy observes the prefix**

   and emits <think> reasoning, then <answer>

2. **Rewrite or copy**

   copying the utterance unchanged is a legal action

3. **Frozen parser runs twice**

   once on the original prefix, once on the rewrite

4. **Reward from the difference**

   both parses compared against the gold structure

Group-relative advantages, KL penalty to the base policy, parser frozen throughout.

Verifiable reward

+2	fixes an originally wrong prediction
−2	breaks an originally correct one
+1	right attachment, wrong relation label
0	neutral or ambiguous change

Plus a format reward: +1 for a well-formed <think> / <answer> block, −2 otherwise.

Finding 3: RL learns when not to rewrite

The gain is avoided harm, not better rewrites · STAC, frozen Qwen3-8B parser

	L6-FREE		RL
Edit %	49.9%	→	**21.4%**
Regressions	38	→	**24**
Repairs	17	→	**7**
Link+Rel F1	0.577	→	**0.586**

- **Repairs went down.**

  The gain comes from avoiding harmful edits — not from producing better rewrites.

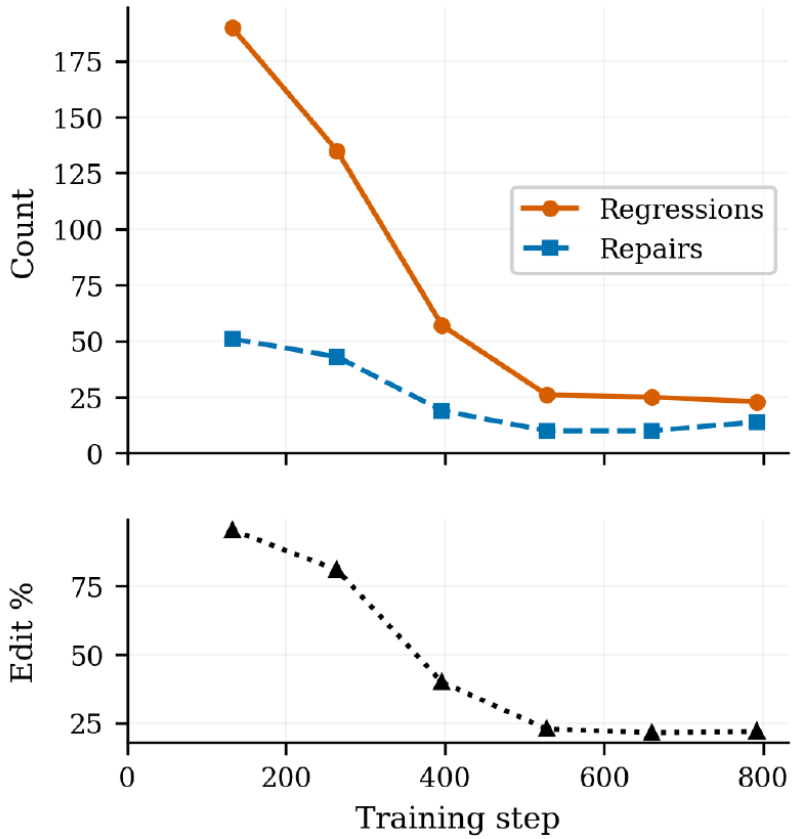


Fig. 2(a) · validation dynamics: regressions and edit rate fall together

Finding 3: Two regimes, one reward

Identical GRPO objective — opposite intervention rates — both beat L6-FREE

STAC *abstains*	**21.4%** edit rate	**0.577 → 0.586** Link+Rel
Molweni *rewrites everything*	**98.8%** edit rate	**0.565 → 0.571** Link+Rel
MSDC *unstable*	**33.8%** edit rate	**0.789 → 0.784** Link+Rel

Fig. 2(b) Molweni

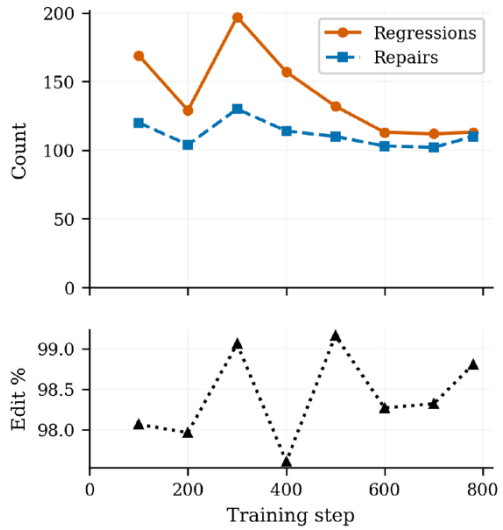
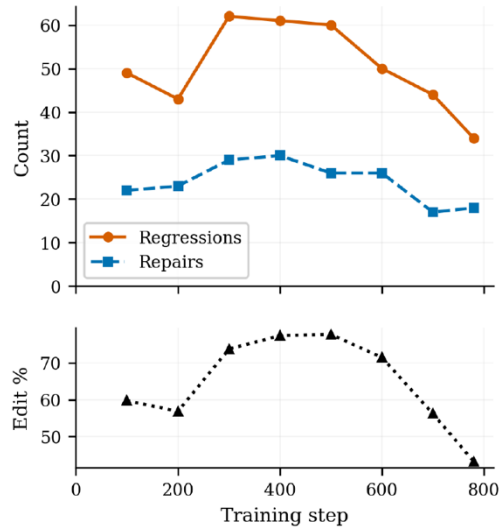


Fig. 2(c) MSDC



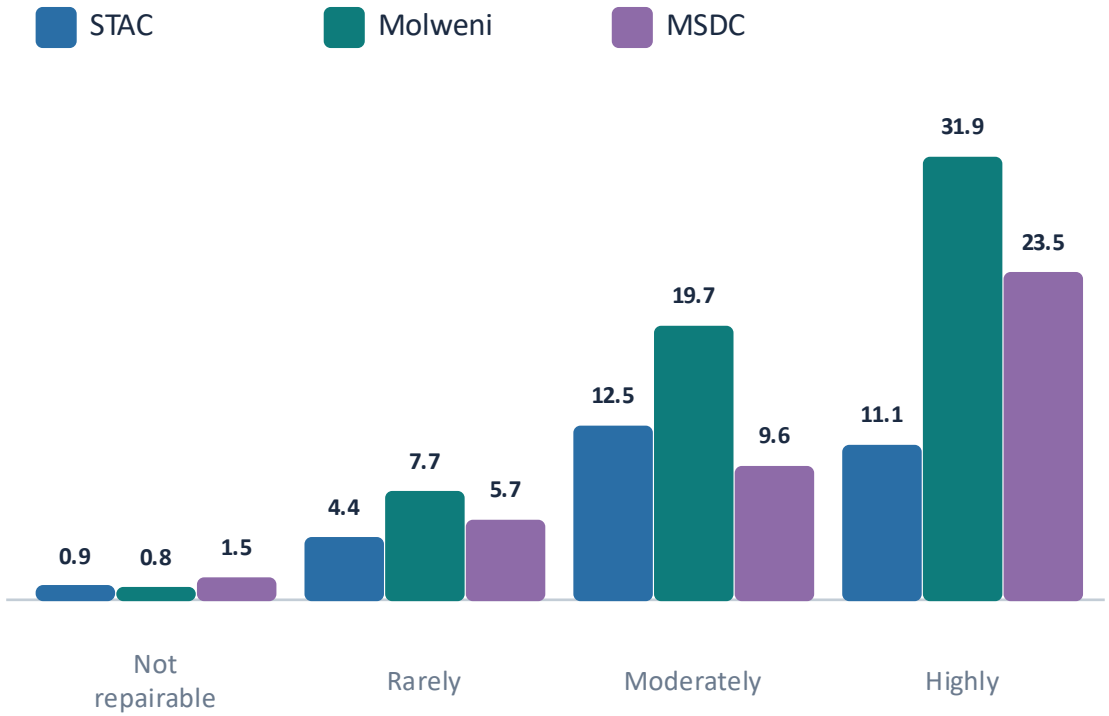
Gains are label-only — Link attachment is flat (Molweni 0.849 → 0.846; MSDC 0.873 → 0.872).

The reward shapes how often to intervene — not when. Molweni's 0.571 also equals the no-clarification parser: rewriting recovers its own cost, it does not beat leaving the input alone.

Finding 3: the rewrites themselves are good

Repair success rises sharply with repairability — and RL can exceed the templates

RL repair rate by estimated repairability bucket



Repair rate (%) on baseline-wrong validation examples; STAC's top bucket holds only 14 examples.

Beyond the best-of-8 candidate pool

Original	"tough times.."
Templates	"So, tough times.." — connective only
RL rewrite	**"I'm out of resources."**
Gold	**EXPLANATION**

Recovers a discourse function the template strategies cannot express.

Qualitative Analysis: what actually flips the parser

Two views: how fragile the parser is, and what RL finds when it does intervene

Meaning-preserving edits flip predictions

“yeah” → “yes” — fix

Acknowledgement → Question-answer_pair

“go nz” → “go New Zealand” — fix

Continuation → Result

“lol” → “laughing out loud” — break

Comment → Elaboration

“peurto rico” → “Puerto Rico” — break

Contrast → Comment

RL finds rewrites the templates cannot express

“really” — Clarification_question

templates: “really”

RL: “oh, you mean 7 again?”

“Sorry, sheep” — Correction

templates: “But, Sorry, sheep”

RL: “Sorry, I meant sheep”

“tough times..” — Explanation

templates: “So tough times..”

RL: “I’m out of resources.”

The parser is sensitive to surface form it should ignore · the policy can act on discourse function when it does intervene

Main Takeaways

- Parser-agnostic rewriting often hurts more than it helps.
- 20%-28% of errors are repairable by rewriting the last utterance
- RL learns how often to rewrite — not which utterances need it

Clarification is a selective intervention problem

- Missing capability: **rewritability prediction**
  - Decide whether an utterance is repairable before intervening

Thank you

Q&A

Yiming Liu
ymliu@ou.edu
University of Oklahoma