Confidence-Routed LLM Reranking for Cost-Effective Dialogue Disentanglement

Tim Yang and **Jonathan K. Kummerfeld**
The University of Sydney
jonathan.kummerfeld@sydney.edu.au

Abstract

Dialogue disentanglement (DD) separates interleaved conversations. Prompting a large language model (LLM) for every message is expensive, and utterance-level prompting has performed poorly. We combine an LLM with a DD system that scores candidate antecedents, routing only cases below a tuned confidence threshold to the LLM. Although weaker than the strongest specialised systems, the LLM improves multiple systems on every Ubuntu IRC metric, while transfer to three other datasets is mixed. For our strongest selectively routed result, querying the LLM for only 42% of messages raises conversation F1 by 20.5 points, and a random-routing control attributes 83% of this gain to confidence selection rather than routing volume. Link F1 gains largely hold with an open-weight LLM. These results show the potential for LLMs to help even when their performance lags specialised methods, a finding that could extend to other structured tasks.

1 Introduction

Platforms such as WhatsApp and Slack allow many users to communicate in a channel concurrently. This can lead to multiple concurrent conversations that are interwoven, which is a particular issue when communication is rapid, with no explicit indication of what each message is a response to. The task of Dialogue Disentanglement (DD) involves taking a series of messages and identifying the distinct conversations. Beyond UX benefits for chat applications, DD improves performance in downstream tasks like sentiment analysis, text summarisation, and conversational question answering by clearly demarcating each conversation's start and end (Yu et al., 2024). Prior work has focused on specially-developed models, with LLMs such as BERT used to generate message embeddings (Yu et al., 2024; Ma et al., 2022; Liu et al., 2021). Attempts to use LLMs via prompting alone have performed poorly (Li et al., 2024).

DD remains relevant for downstream tasks like question answering despite the existence of long-context reasoning models. Without explicit disentanglement, models must jointly select relevant messages and separate distinct conversational threads, increasing the risk of cross-thread interference. This makes it important for downstream correctness beyond context-length considerations.

We introduce a method for improving Dialogue Disentanglement models that output probabilities for possible parent messages given a child query message. Where the base model is confident, we keep its choice. Where it is not, we prompt a generative LLM with the base model's top-k predictions. Confidence is measured by the top probability the base model assigns to its top prediction.

To evaluate the approach, we tried combining with a wide range of models. Despite working with past authors, we could not replicate some of the prior leading approaches. On the systems we could run, confidence-routed assistance improves every metric on Ubuntu IRC and most metrics on IRC New Domains, while transfer to Channel Two and Dramatic Dialogue is mixed. The LLM boosts the strongest base models we could run despite being weaker overall, because the two fail on different messages. A random-routing control confirms that confidence selection, rather than routing volume alone, drives the gains for the selectively routed base models. Selective routing also costs less than prompting an LLM for every message.

Our hybrid approach may be a promising avenue for cheaply and effectively using LLMs in other challenging structured tasks.

2 Related Work

Early supervised DD work used binary classifiers with hand-engineered features (Elsner and Charniak, 2010; Kummerfeld et al., 2019). Subsequent work has improved results by treating the task as

a pointing link prediction problem (PointerNet Yu and Joty, 2020), using DAG-LSTMs (Pappadopulo et al., 2021), combining BERT and Bi-LSTMs (Li et al., 2022), deriving structural features using relational-graph convolutions (Ma et al., 2022), and applying reinforcement learning (Bhukar et al., 2023). Unfortunately, in many cases, the code for these models are not released publicly or in response to requests, and even when released, some results cannot be reproduced.

Unsupervised DD is harder, but avoids the costs of manual reply-to or cluster annotations. Qiu et al. (2020) first reframed DD as a text generation problem and used a Variational RNN to extract reply-to links. Chi and Rudnicky (2021) extended this idea of reframing by adapting BERT models by pretraining a BERT model on response selection before adapting it for link prediction. Other methods include Liu et al. (2021)'s co-training of complementary message-to-message and message-to-conversation models, and Huang et al. (2022)'s contrastive learning with expectation maximisation to cluster same-session messages.

LLM Use in Disentanglement Language models are widely used as encoders for DD, but prompting them directly performs poorly: zero-shot GPT-3.5 reaches just 10% pairwise link F1 (Li et al., 2024), and out-of-the-box GPT-4o recovers no complete conversations (Pal et al., 2026). Fine-tuning (Pal et al., 2026) and automatic prompt optimisation of open-weight models (Takada et al., 2026) both fall short of supervised systems on conversation F1. Takada and Mori (2026) obtain the strongest cluster-level scores to date by reformulating the task as dialogue-level assignment with subsequent context, but query a frontier LLM for every utterance and do not quantify its API cost. All of these replace the supervised system. To our knowledge ours is the first to improve one, invoking an LLM only where the base model is uncertain, and to quantify what that costs.

Confidence-Based Routing Routing inputs by model confidence is well established: selective prediction abstains on low-confidence cases (Geifman and El-Yaniv, 2017), and LLM cascades escalate them to a larger model to cut cost (Chen et al., 2023). Our setting inverts this: we defer to a weaker model, and the deferred decision is a graph link, not an independent label.

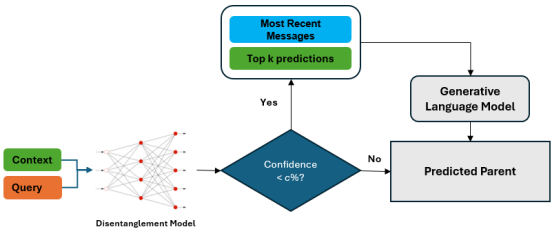


Figure 1: System architecture.

3 Methodology

For each message, the base model gives each earlier message a probability of being its parent. We call the highest of these probabilities the base model's *confidence*. If the confidence is at least c, we keep the base model's choice. Otherwise, an LLM chooses the parent from the base model's top-k candidates (Figure 1). The zero-shot prompt has three parts:

1. The top-k Predictions from the base model. A low k restricts the search space to likely high-quality candidates, allowing it to allocate more tokens to reason or consider each possible parent utterance.
2. The l messages before the query message. This context aids the LLM in challenging cases where pairwise comparison is insufficient.
3. Instructions (i) that ask the model to consider all options and choose the best answer, taking into account the context messages. For the full prompt, see Appendix Figure 6.

Data We focus on the primary dataset for the task: Kummerfeld et al. (2019)'s IRC Disentanglement dataset. We also evaluate on the Dramatic Dialogue (Chang et al., 2023), the IRC New Domains (Gouravajhala et al., 2023), and IRC Channel Two datasets (Kummerfeld et al., 2019).

Language Model We use OpenAI's GPT-5 (*gpt-5-2025-08-07*) and the open-weight DeepSeek V4.1 Flash (*deepseek-v4.1-flash-1017*) with identical prompts and c.

Base Models We evaluated using the best-performing link-prediction-based models that have reproducible results: Feedforward (FF) (Kummerfeld et al., 2019) and Zero-shot (ZS) (Chi and Rudnicky, 2021). We applied our technique to the Structural Encodings (ST) system (Ma et al., 2022),

Data	Model	Link Level			Conversation (Cluster) Level						
		ΔP	ΔR	ΔF1	ΔP	ΔR	ΔF1	ΔS-VI	ΔARI	ΔNMI	ΔS-F1
Ubuntu IRC	FeedForward	6.3	6.1	6.2	22.8	17.7	20.5	4.5	22.0	5.0	9.9
	Structure	2.9	2.8	2.8	4.8	3.4	4.2	1.3	7.9	1.5	2.4
	Zero-shot	16.3	15.7	16.0	12.0	7.1	9.2	2.6	9.4	1.0	5.8
New Domains	FeedForward	5.6	5.6	5.6	19.5	7.6	13.6	3.7	7.2	1.3	4.4
	Structure	4.0	4.0	4.0	9.1	6.5	7.9	0.9	-7.1	1.5	0.5
	Zero-shot	17.3	17.2	17.2	20.5	8.7	12.3	2.3	-0.5	0.4	1.3
Channel Two	FeedForward	8.1	8.1	8.1	11.5	2.6	7.9	4.2	7.7	0.7	-1.4
	Structure	3.8	3.8	3.8	0.5	-2.6	-0.1	1.9	4.4	2.4	2.9
	Zero-shot	17.6	17.5	17.6	3.4	-7.6	-5.5	-4.1	-9.9	-13.7	-12.8
Dramatic Dialogue	FeedForward	-0.2	-0.2	-0.2	0.9	0.0	0.1	-0.4	-1.9	-0.6	-1.3
	Structure	0.4	0.4	0.4	0.9	-0.5	-0.1	-0.1	-0.6	-0.8	-0.9
	Zero-shot	20.6	20.6	20.6	12.0	2.1	4.5	1.4	0.5	-1.6	-0.7

Table 1: Absolute score changes (percentage points) from GPT-5 re-ranking of each base model. S-VI: scaled VI; S-F1: Shen F1.

which has released and runnable code, but we did not achieve the reported results.

We compare to the top link-model (E2E-RL) but do not apply our approach as we could not replicate their results despite extensive effort. We received valuable input from the E2E-RL authors, but their approach depends on ST, and we were unable to replicate the ST results and were not able to get assistance from the ST authors. Details of our work on running other models are in appendix A.2.2.

Evaluation We use the standard DD metrics defined by Kummerfeld et al. (2019). For definitions, see appendix A.12.

Hyperparameters We set k=5 and l=10 (see App. A.3 for context-length and prompting ablations). For c, we measure the LLM's link precision on a development sample (66.7% GPT-5, 71.7% DeepSeek) and route every confidence bin where the base model's development-set precision is lower, giving 70% for FF and 90% for ST with both LLMs. For ZS we route every message (c=100%, App. A.6). c varies across base models, so it must be tuned on labelled validation data (see Limitations).

4 Results

Table 1 reports absolute score changes: our method improves every metric on Ubuntu IRC and most on New Domains, but has mixed results on Channel Two and Dramatic Dialogue.[1] There are particularly large gains in precision of complete conversations. With DeepSeek as the re-ranker, FF and ST gains are within 0.5 link F1 of GPT-5 on all four datasets, and ZS gains are 0.5-1.7 points lower (App. A.4).

[1]For absolute values and results of other models, see appendix tables 9, 10, 11, and 12.

Model	Link F1	Conv P	Conv R	Scaled VI	ARI
FeedForward (FF)	71.7	31.4	36.9	91.2	63.0
+ Assist@70%	77.9	54.2	54.6	95.7	85.0
+ Assist (DS)	77.9	53.1	52.4	95.7	86.0
Structure (ST)	73.3	36.8	45.6	92.7	65.4
+ Assist@90%	76.1	41.6	49.0	94.0	73.3
+ Assist (DS)	76.1	42.2	49.6	94.0	73.3
Zero-shot (ZS)	46.6	27.0	24.5	85.3	51.8
+ Assist@100%	62.6	39.0	31.6	87.9	61.2
+ Assist (DS)	61.9	35.7	25.1	86.1	60.1
Pure GPT-5	67.7	24.1	15.8	87.2	60.1
Pure DeepSeek	68.5	25.0	20.0	88.5	65.7
DLA+SC	–	**58.8**	**63.9**	**97.2**	**92.2**
E2E RL	**83.3**	51.2	52.3	96.2	78.5
DAG-LSTM	73.9	42.4	41.7	–	80.1
PointerNet	73.1	44.9	44.2	94.2	80.1

Table 2: Ubuntu IRC results. LLM rows use GPT-5 unless marked DS (DeepSeek V4.1 Flash). Pure: LLM alone. @c%: confidence threshold. Bottom: published results.

Large Gains on Ubuntu IRC Table 2 shows scores, with our approach on +Assist lines. FF+Assist improves every metric but trails DLA+SC (Takada and Mori, 2026), the strongest published cluster result, which queries a frontier LLM for every utterance and reports neither link accuracy nor API cost. For ZS the threshold is c=100%, so every message is re-ranked. Its gain therefore comes from replacing a weak base model, not from selective routing. Even so, restricting GPT-5 to the top-5 ZS candidates raises conversa-

tion F1 from 19.1 (Pure GPT-5) to 34.9. Direct prompting is not competitive with supervised systems, but it is much stronger than in prior work. Pure DeepSeek reaches 68.5 link F1, the highest unsupervised link F1 we are aware of, but both LLMs alone trail ZS+Assist on conversation F1. For full metrics, see App. table 9.

Channel Two Shows Variable Results While our approach improves the FeedForward and Structure systems on most metrics on Channel Two, the results for the Zero-shot model are mixed. One possible explanation is the nature of the discussion, which has several extremely long conversations (the data is one continuous sample of communication). That makes it possible to improve links between messages within a conversation while introducing errors that mix or split conversations.

The Dramatic Dialogue Dataset is an Outlier. Our approach is less successful on the dramatic dataset for two reasons. First, the LLM is rarely used because the base models (FF, ST) have higher average confidence for their predictions. Second, the dataset focuses on turn prediction rather than dialogue structure (Chang et al., 2023), with nearly 90% of the links being to the previous line, compared with 15-58% for the IRC family of datasets.

Confidence selection drives the gain. To test whether the choice of re-ranked messages matters, we re-ranked the same number of messages, but randomly with ten different random seeds. Random routing is not effective. For FF, conversation F1 rises only from 33.9 to 37.4$\pm$1.3. For ST it does not improve at all, moving from 40.8 to 40.2$\pm$1.1. This shows that the benefits depend on the approaches having different strengths, and the routing method being able to appropriately direct decisions. The base models we considered were consistently able to estimate uncertainty effectively enough to make routing work.

Why an LLM with low accuracy still helps. It may be surprising that the LLM helps at all, since alone it is weaker and cannot do well where the base model is confident (App. Figs. 4, 5). The two systems fail in different places. On Ubuntu IRC messages that FF routes, FF is correct for 52.3% and the GPT-5 re-ranker for 67.3%, and on messages FF keeps, 88.2% vs. 81.8%. ST is similar (38.6 vs. 60.1% routed, 80.3 vs. 78.2% kept). The base model's accuracy drops steeply with its confidence and the re-ranker's much less, so routing gives each message to the system more likely to be right. The re-ranker's lower accuracy on kept messages may reflect annotation conventions that supervised models learn in training.

Common corrections and mistakes On Ubuntu IRC, re-ranking changed 1030 FF, 407 ST and 2196 ZS predictions, yet only 26 correct and 47 incorrect changes were shared by all three, suggesting the re-ranker corrects each model's own errors and may therefore transfer to other link-based DD models.

Costs Re-ranking is more cost-effective than pure LLM inference as it targets only a subset of data. A higher c implies more assistance but higher cost, so tuning c on validation data can help optimise this tradeoff when inferring on a larger test set. Only 42% of the IRC test set for the FF model is required for LLM re-ranking, costing about $13 USD to achieve the improvements in Table 2. In contrast, pure GPT-5 inferring on the entire test set cost $28 USD. DeepSeek reduced the FF re-ranking cost from about $13 to $1.05 USD.

Cluster metrics vary more than link metrics. Cluster metric gains as small as 1–2% have been used to claim SOTA improvements (Yu and Joty, 2020; Pappadopulo et al., 2021; Jiang et al., 2022). However, retraining FF or Structure models across five seeds shows cluster metrics can vary by 4–15%, which is 5–6x more than link metrics (App. Tab. 8). We believe this difference has two causes: (1) each link error creates two cluster errors, and (2) there are about 5x more links than clusters. Together, these mean each error will shift a conversation metric much more than a link metric. Link metrics are therefore more reliable for comparing models whose cluster scores are close.

5 Conclusion

We introduce confidence-routed LLM re-ranking for dialogue disentanglement. Although weaker than the strongest supervised models, the LLM improves multiple base systems when applied to uncertain cases. This is possible because the router effectively estimates confidence and the models have different strengths. Selective routing costs less than querying every message and may transfer to other structured tasks where LLMs are expensive or ineffective alone.

6 Limitations

Our method applies only to base models that produce link predictions, not to those predicting conversation clusters directly. Combining a cluster model with a link model, using the clusters to constrain the links, could address this, but adds pipeline complexity for what we expect would be small gains.

The re-ranking LLMs have far more parameters than task-specific link models, so invoking them even below $c\%$ adds non-trivial computational overhead. Reducing this through smaller re-rankers, distillation, or tighter invocation criteria is an important direction, as is a cost-aware objective balancing marginal gains in link prediction against the invocation rate and the cost per call.

Our method adds one hyperparameter, the confidence threshold c, and setting it needs labelled development data: the base model's development-set precision and the LLM's precision on a development sample. Its value differs by base model (70% for FF and 90% for ST here) and, on test data, its best value also differs by dataset (App. Figs. 4 and 5). We set c on Ubuntu IRC development data only and reused it unchanged on the other three datasets, which may explain part of the mixed transfer. Applying the method to a new base model or domain therefore needs a small annotated sample. Setting c without labelled data is future work.

Our main experiments use GPT-5, the development ablations use GPT-4-Turbo, and newer models have since been released. The approach transferred to one open-weight model (DeepSeek V4.1 Flash) with the GPT-5 prompts unchanged, and the same threshold rule gave it the same c, but we ran each configuration once.

Despite working with past authors, we could not replicate some prior leading approaches. Applying our method to them may improve results further.

Our small-scale prompting experiments and some appendix ablations use subsets of the development data. We believe these were large enough to guide design decisions, but they may be noisy. All main results are on the full test set, as in prior work.

We evaluate on four datasets. Others such as the Discord (Subash et al., 2022) and movie dialogue (Liu et al., 2020) corpora remain untested due to time and cost.

7 Note on DeepSeek Use

Experiments with DeepSeek models were conducted by the first author after graduation. The second author did not download or use DeepSeek models in any way as part of this work and no University of Sydney data was involved in the experiments.

Acknowledgments

This material is partially funded by an unrestricted gift from Google, and by the Australian Research Council through a Discovery Early Career Researcher Award.

References

Karan Bhukar, Harshit Kumar, Dinesh Raghu, and Ajay Gupta. 2023. End-to-End Deep Reinforcement Learning for Conversation Disentanglement. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(11):12571–12579. Number: 11.

Kent Chang, Danica Chen, and David Bamman. 2023. Dramatic Conversation Disentanglement. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4020–4046, Toronto, Canada. Association for Computational Linguistics.

Lingjiao Chen, Matei Zaharia, and James Zou. 2023. FrugalGPT: How to Use Large Language Models While Reducing Cost and Improving Performance. *arXiv preprint*. ArXiv:2305.05176.

Ta-Chung Chi and Alexander Rudnicky. 2021. Zero-Shot Dialogue Disentanglement by Self-Supervised Entangled Response Selection. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4897–4902, Online and Punta Cana, Dominican Republic.

Micha Elsner and Eugene Charniak. 2010. Disentangling Chat. *Computational Linguistics*, 36(3):389–409.

Yonatan Geifman and Ran El-Yaniv. 2017. Selective Classification for Deep Neural Networks. In *Advances in Neural Information Processing Systems 30 (NeurIPS 2017)*.

Sai R. Gouravajhala, Andrew M. Vernier, Yiming Shi, Zihan Li, Mark S. Ackerman, and Jonathan K. Kummerfeld. 2023. Chat Disentanglement: Data for New Domains and Methods for More Accurate Annotation. In *Proceedings of the 21st Annual Workshop of the Australasian Language Technology Association*, pages 112–117, Melbourne, Australia. Association for Computational Linguistics.

Chengyu Huang, Zheng Zhang, Hao Fei, and Lizi Liao. 2022. Conversation Disentanglement with Bi-Level

Contrastive Learning. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2985–2996, Abu Dhabi, United Arab Emirates.

Lawrence Hubert and Phipps Arabie. 1985. Comparing partitions. *Journal of Classification*, 2(1):193–218.

Ziyou Jiang, Lin Shi, Celia Chen, Fangwen Mu, Yumin Zhang, and Qing Wang. 2022. MuiDial: Improving Dialogue Disentanglement with Intent-Based Mutual Learning. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, pages 4164–4170, Vienna, Austria.

Jonathan K. Kummerfeld, Sai R. Gouravajhala, Joseph J. Peper, Vignesh Athreya, Chulaka Gunasekara, Jatin Ganhotra, Siva Sankalp Patel, Lazaros C Polymenakos, and Walter Lasecki. 2019. A Large-Scale Corpus for Conversation Disentanglement. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3846–3856, Florence, Italy.

Bobo Li, Hao Fei, Fei Li, Shengqiong Wu, Lizi Liao, Yinwei Wei, Tat-seng Chua, and Donghong Ji. 2024. Revisiting Conversation Discourse for Dialogue Disentanglement. *ACM Trans. Inf. Syst.*, 43(1):15:1–15:34.

Tianda Li, Jia-Chen Gu, Zhen-Hua Ling, and Quan Liu. 2022. Conversation- and Tree-Structure Losses for Dialogue Disentanglement. In *Proceedings of the Second DialDoc Workshop on Document-grounded Dialogue and Conversational Question Answering*, pages 54–64, Dublin, Ireland.

Hui Liu, Zhan Shi, Jia-Chen Gu, Quan Liu, Si Wei, and Xiaodan Zhu. 2020. End-to-End Transition-Based Online Dialogue Disentanglement. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, pages 3868–3874, Yokohama, Japan.

Hui Liu, Zhan Shi, and Xiaodan Zhu. 2021. Unsupervised Conversation Disentanglement through Co-Training. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2345–2356, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xinbei Ma, Zhuosheng Zhang, and Hai Zhao. 2022. Structural Characterization for Dialogue Disentanglement. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 285–297, Dublin, Ireland.

OpenAI. 2023. Generative pre-trained transformer 4. `https://openai.com/`. Accessed: 2024-05-20.

Debaditya Pal, Anton Leuski, Ron Artstein, David Traum, and Kallirroi Georgila. 2026. Disentangling Approaches to Conversation Disentanglement: Fine-Tune or Learn from Scratch? In *Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026)*, pages 2927–2941, Palma de Mallorca, Spain.

Duccio Pappadopulo, Lisa Bauer, Marco Farina, Ozan İrsoy, and Mohit Bansal. 2021. Disentangling Online Chats with DAG-structured LSTMs. In *Proceedings of *SEM 2021: The Tenth Joint Conference on Lexical and Computational Semantics*, pages 152–159, Online.

Liang Qiu, Yizhou Zhao, Weiyan Shi, Yuan Liang, Feng Shi, Tao Yuan, Zhou Yu, and Song-Chun Zhu. 2020. Structured Attention for Unsupervised Dialogue Structure Induction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1889–1899, Online.

C E Shannon. 1948. A Mathematical Theory of Communication. *The Bell System Technical Journal,*, 27:379–423, 623–656.

Dou Shen, Qiang Yang, Jian-Tao Sun, and Zheng Chen. 2006. Thread detection in dynamic text message streams. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '06, pages 35–42, New York, NY, USA. Association for Computing Machinery.

Keerthana Muthu Subash, Lakshmi Prasanna Kumar, Sri Lakshmi Vadlamani, Preetha Chatterjee, and Olga Baysal. 2022. DISCO: a dataset of discord chat conversations for software engineering research. In *Proceedings of the 19th International Conference on Mining Software Repositories*, MSR '22, pages 227–231, New York, NY, USA. Association for Computing Machinery.

Naoki Takada and Tatsunori Mori. 2026. Rethinking Dialogue Disentanglement for LLMs via Dialogue-Level Assignment and Subsequent Context. In *Proceedings of the 10th Linguistic and Cognitive Approaches to Dialog Agents Workshop (LaCATODA) at AAAI 2026*, volume 4178 of *CEUR Workshop Proceedings*. CEUR-WS.

Naoki Takada, Tomohiro Okatsu, Yin Min Pa Pa, Katsunari Yoshioka, and Tatsunori Mori. 2026. APO-DD: Dialogue Disentanglement via Reflective Optimization of Multi-Component Prompts. *arXiv preprint*. ArXiv:2606.07894v2.

Fei Yu, Hongbo Zhang, Prayag Tiwari, and Benyou Wang. 2024. Natural Language Reasoning, A Survey. *ACM Computing Surveys*. Just Accepted.

Tao Yu and Shafiq Joty. 2020. Online Conversation Disentanglement with Pointer Networks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6321–6330, Online.

A Appendix

A.1 Performance gain characteristics

App. Figures 4 and 5 show link and cluster F1 trends as the c parameter increases for the three models across all datasets. Nearly all exhibit a concave pattern: metrics rise, peak, then drop significantly past a 90% confidence threshold. The zero-shot model's link metrics only flatten.

Link and cluster F1 do not always align. For FF (Ubuntu IRC, New Domains IRC), cluster F1 continues rising after link F1 declines, while for ZS (all datasets), link F1 rises even as cluster F1 falls. In contrast, ST shows near-perfect synchronisation between the metrics. This indicates a trade-off between cluster and link metrics in some models.

A.2 Base Model Selection

A.2.1 Evaluated

We evaluated our methods on models m from the following list

Feedforward (FF) (Kummerfeld et al., 2019) - A supervised feedforward net approach that takes as input a GloVe embedding (trained on IRC DD dataset) of messages along with 77 handcrafted features to represent each message, and concatenates the representations pairwise between query and candidate parent utterances.

Structure (ST) (Ma et al., 2022) - A supervised model which performs better than FF in cluster metrics by using graph-convolutional neural nets (GCNs) to create inter-speaker, reply-to, and same-speaker relations between messages and speakers (Ma et al., 2022). The model was later adapted as an embedding module within Bhukar et al. (2023)'s reinforcement learning approach which achieved SOTA performance.

Zeroshot (ZS) (Chi and Rudnicky, 2021) - A self-supervised model based on BERT which is trained on the entangled response selection task (distinct from dialogue disentanglement) and adapted for zero-shot inference. This model is the only one within the unsupervised category to provide both link and cluster metrics (Chi and Rudnicky, 2021). Our investigation considers their $w = 0.25$ model trained only on the entangled response selection task that produced the best overall unsupervised performance for the dialogue disentanglement task. This model is the best unsupervised DD approach which we replicated.

A.2.2 Considered

We considered several more models but experienced difficulty reproducing the results, primarily because of unavailable or incomplete source code. During evaluation, we show the best version of their reported metrics for our assessment.

PointerNet (Yu and Joty, 2020) - When running the public model code without Kummerfeld et al. (2019) hand-crafted features, results were far below those reported in the original paper. Even when using the hand-crafted features we could not closely match the model's metrics to those claimed as an improvement compared to the FF baseline model.

Reinforcement Learning (E2E RL) (Bhukar et al., 2023) - This is the strongest published supervised model on link-level metrics. We contacted the authors from IBM for guidance to replicate their results. However, we were informed that the lead author of the paper has since left the firm and no longer has access to the model's code and weights. We attempted to reproduce their RL method with guidance from correspondence, but saw no significant improvement from the base model they fine-tuned on (Structure BERT) (Ma et al., 2022).

Discourse Structure (Li et al., 2024) - A high-performing model that nears the performance of the ST model. However, the model has no publicly available source code.

Other models - Unsupervised approaches like Liu et al. (2021)'s Co-Training and Huang et al. (2022)'s Bi-level contrastive learning do not produce link metrics which our method relies on to make improvements. Additionally, Huang et al. (2022)'s model is not publicly available.

A.3 Development set comparisons

The experiments in this section each use 3 sets of 100 development messages, not the full test set.

Only local context is useful. Providing context messages, going from 0 to 10, increases the re-ranker's performance by 12% absolute on average across three trials. However, performance plateaus beyond 10 messages, likely because the assistant either ignores older messages or more context adds little value. For details, see appendix Tab. 5.

Prompting variants. Chain-of-thought, ReAct and few-shot prompting do not outperform zero-shot prompting in isolation. Full results, a comparison of re-ranker models, and all prompts are in App. Tabs. 3, 6, 4 and App. Figs. 10, 7, 8, 9.

LLM	Trial			Average
	1	2	3	
Zeroshot	70	74	71	71.7
React	67	69	67	67.7
Chain of Thought	74	69	42	61.7
Fewshot	64	59	60	61.0

Table 3: Direct Prompting: Comparisons of various prompting techniques on their own each having a 25 message context window (not re-ranking)

LLM	Trial			Average
	1	2	3	
GPT5	65	70	65	66.7
DeepSeek V4.1 Flash	75	69	71	71.7
GPT4Turbo	58	57	60	58.3
Claude Opus	55	54	53	54.0
Gemini Pro	49	51	49	49.7
Mistral 7B	22	33	27	27.3
Olmo2 13B	20	25	30	25.0

Table 4: Re-ranking: Correct link predictions out of 100 using different generative LLMs for an assistant (development set). The GPT-5 and DeepSeek rows are direct-prompting trials on the same development links.

Unlike standalone prompting (App. Tab. 9), re-ranking restricts the LLM to the base model's top-k candidates, so the relative ordering of LLMs can differ between the two settings.

LLM	Trial			Average
	1	2	3	
GPT4 (30 prev.)	57	58	61	58.7
GPT4 (20 prev.)	59	59	54	57.3
GPT4 (10 prev.)	58	57	60	58.3
GPT4 (no prev.)	47	46	46	46.3

Table 5: Assistant: Results using context vs without with 50% assistance threshold (development set, GPT-4-Turbo)

LLM	Trial			Average
	1	2	3	
GPT4 (zero-shot)	58	57	60	58.3
GPT4 (few-shot)	58	59	55	57.3

Table 6: Assistant: Zero-shot vs Few-shot with 50% assistance threshold (development set, GPT-4-Turbo)

A.4 Open-weight re-ranker

We replayed the saved GPT-5 requests byte-for-byte through DeepSeek V4.1 Flash (*deepseek-v4.1-flash-1017*), pinned to a single OpenRouter host serving the official FP8/FP4 checkpoint, with reasoning effort *medium* and a 16k-token output cap (reached by 3 of 37,337 answers). Requests routed across many hosts produced empty or runaway answers on some of them, so pinning one host was necessary. The threshold rule of App. A.6, applied to DeepSeek's dev-set precision (71.7%), gives the same c as for GPT-5 (70% for FF and 90% for ST, with ZS routing every message). Each configuration was run once. Both re-rankers lower Channel Two ARI, VI and NMI under ZS, with DeepSeek more so (ARI 2.2 vs 11.7, Tab. 11), consistent with link gains that merge long conversations.

		ΔLink F1		ΔConv F1	
Data	Model	GPT-5	DS	GPT-5	DS
Ubuntu IRC	FF	6.2	6.2	20.5	18.9
	ST	2.8	2.8	4.2	4.8
	ZS	16.0	15.3	9.2	3.8
New Domains	FF	5.6	5.8	13.6	13.6
	ST	4.0	3.7	7.9	8.1
	ZS	17.2	16.7	12.3	10.0
Channel Two	FF	8.1	7.7	7.9	9.5
	ST	3.8	3.4	-0.1	0.1
	ZS	17.6	16.9	-5.5	2.3
Dramatic	FF	-0.2	-0.3	0.1	-0.6
	ST	0.4	0.3	-0.1	-0.5
	ZS	20.6	18.9	4.5	4.2
Pure LLM, Ubuntu IRC†		67.7	68.5	19.1	22.2

Table 7: Gains over each base model with GPT-5 and DeepSeek V4.1 Flash (DS) as the re-ranker, same prompts and c. †Absolute scores, not gains.

A.5 Cost

GPT-4-Turbo costs $10 USD and $30 USD per million input and output tokens, respectively (OpenAI, 2023). Inference on the 5000-message Ubuntu IRC dataset test set without using OpenAI's batch API cost a total of $350 USD. $120 USD in total was spent running experiments on the 2,400-message New domains and 1,000-message Channel Two datasets, and $30 USD was spent on the 2,380-message dramatic dataset, all benefiting from OpenAI's batch API, which halves the cost per token. A further $50 USD was spent on the experiments in the analysis section.

We updated the re-ranking experiments with GPT-5, which has a lower token cost and this added another $150 USD in experimental costs.

The open-weight rerun with DeepSeek V4.1 Flash (all 37,337 re-ranking and direct-prompting requests, App. A.4) cost $18 USD via OpenRouter

($1.05 USD for the FF re-ranking on Ubuntu IRC, against about $13 USD with GPT-5), plus about $5 USD for pilots and answer parsing.

In total, approximately $725 USD was spent on running all experiments.

A.6 Hyperparameters

We measure the LLM's link precision on 3 samples of 100 randomly selected Ubuntu IRC development links (66.7% for GPT-5, 71.7% for DeepSeek) and set c to the upper edge of the highest 10% confidence bin in which the base model's development-set link precision is below that value, routing all lower bins too (App. Fig. 3 and Tab. 4). For both LLMs this gives 70% for FF and 90% for ST, covering 44% and 14% of dev. samples, respectively, on the Ubuntu IRC dataset (App. Fig. 2). We could not apply this rule to ZS because its development-set predictions were unusable, so we route every ZS message (c=100%). We further select $k = 5$, $l = 10$, and instructions from App. Fig. 6.

App. Figs. 4 and 5 show test-set sensitivity to c with GPT-5, in 10% steps. The development-set rule gives a c near, but not at, the test optimum: for FF on Ubuntu IRC, c=80% would give 78.2 link F1 rather than 77.9, and c=90% would give 57.7 conversation F1 rather than 54.4. On the three IRC datasets, every c from 40% to 90% raises FF and ST link F1. On Dramatic Dialogue, FF link F1 falls slightly for c above 60% (by 0.2 points at 70%). Routing every message (c=100%) lowers FF and ST link F1 on Channel Two and Dramatic Dialogue by up to 13.3 points. ZS link F1 is highest at c=100% on all four datasets (tied with 90% on Ubuntu IRC).

We select a small $k = 5$ as our preliminary experiments with larger k often resulted in the LLM generating extremely long reasoning traces, and hallucinating the index of the message that was supposed to be selected. Our reason for selecting $l = 10$ is similar, in addition to minimising the length of the prompt to lower inference costs. We apply these hyper-parameters re-ranking inference across all datasets.

A.7 Other Hyper-parameters

A.7.1 Structure

We used the exact Github hyper-parameters as of 15th of October 2024. We trained using a batch size of 2, a learning rate of 5e-6, a max sequence length of 128, and a seed of 42, with a 10% warmup proportion over 4 training epochs.[2]

A.7.2 Zeroshot

We used bert-base-uncased from Huggingface model hub as the BERT checkpoint. We used the exact GitHub parameters provided for both the pre-training and zero-shot inference stages. These include a batch size of 4, an epsilon of 1e-8 for the Adam optimizer, warmup steps of 100, weight decay of 1e-7, max grad norm of 1.0, learning rate of 5e-5, gradient accumulation steps of 1, a seed of 12345, and overall training over 3 epochs.[3]

A.7.3 Feed Forward

We adapted the model from the previous Dynet implementation to a modern PyTorch implementation which introduces no new architectural changes. We used the exact GitHub parameters for the implementation trained using the stochastic gradient descent optimizer. These hyper-parameters include a batch size of 4, hidden layer size of 512, 2 hidden layers, an epsilon of 1e-8 for the Adam optimizer, warmup steps of 100, max grad norm of 1.0, weight decay of 1e-7, learning decay rate of 0.103, gradient clipping threshold of 3.740, learning rate of 0.018804, gradient accumulation steps of 1, a seed of 10, and overall training over 20 epochs.[4]

A.8 Dataset Details

Ubuntu IRC This dataset contains 69,396/2,607/5,187 reply-to relationships across its train/dev/test sets respectively (Kummerfeld et al., 2019). Its conversations are sampled from a technical support channel at 173 points in time between 2004 and 2018 within a single domain.

New Domains IRC This is a multi-domain dataset that contains 2,400 annotated messages from a range of IRC networks (Gouravajhala et al., 2023). It is unique from existing datasets as it introduces 1) new types of conversations (e.g. meetings), and 2) new types of user relationships (business-customer). This data was developed to challenge models in "out-of-domain evaluation".

Channel Two IRC This IRC dataset comes from a different channel compared to the Ubuntu IRC Dataset and was used as an out-of-domain sample

[2] https://github.com/xbmxb/StructureCharacterization4DD/tree/main

[3] https://github.com/chijames/zero_shot_dialogue_disentanglement

[4] https://github.com/jkkummerfeld/irc-disentanglement

in Kummerfeld et al. (2019)'s model evaluations. It contains 1000 samples each for the dev and test set.

Dramatic Dialogue This annotated dataset contains randomly sampled conversations from 831 titles, including 340 movies and 491 TV series (Chang et al., 2023). It contains 2,209 threads totaling 10,033 dialogue turns. After removing script lines that do not contain annotations (e.g. actions) and invalid ones (e.g. ones where the query links to parent message id which does not exist), its train/dev/test sets contain 10,288/2,352/2,379 annotated reply-to links.

A.9 Use of AI

We used OpenAI's ChatGPT in part to assist with visualising the model data output collected.

A.10 Licenses of artifacts

For the feed-forward model, and associated evaluation scripts, we found the source code was published under the ISC license and the IRC Disentanglement dataset(s) were published under CC-BY-4.0. We could not find any explicit licenses for the structure or zero-shot models, or the dramatic dialogue dataset. We only used these artifacts for research, as consistent with the license.

A.11 Metric Variability

Table 8 shows the range (i.e., max score - min score) when training each of the three models with five different seeds. This captures the sensitivity of the metrics to small variations in the models.

A.12 Definition of Metrics

Scaled Variation of Information (Scaled VI) - Measures information gained or lost when changing from one clustering to another, which is calculated as $H(Y|X) + H(X|Y)$, where "X and Y are clusterings of the same set of items" (Kummerfeld et al., 2019).

Adjusted Rand Index (ARI) - Evaluates the quality of clustering algorithms based on the rand index (RI), where an ARI of 1 is a perfect match between the predicted and true clusters, an ARI of 0 indicates the clustering as good as a random assignment, and an ARI of less than 0 indicates the clustering is worse than random (Hubert and Arabie, 1985)

Normalised Mutual Information (NMI) - A normalised version of the mutual information metric that scales results between 0 (very poor conversation clustering) and 1 (perfect conversation clustering), reported here as a percentage. Mutual information quantifies the number of bits of information shared between the two clusterings or conversations (Shannon, 1948).

Metric/Range	FF	ST	ZS
Link Precision	0.9	1.8	6.7
Link Recall	0.8	1.8	6.9
Link F1-Score	0.8	1.8	6.6
Matched Clusters Precision	5.7	12.6	6.3
Matched Clusters Recall	4.5	8.2	3.9
Matched Clusters F-Score	4.5	11.1	5.1
Scaled VI	1.4	3.1	1.4
Adjusted Rand Index	11.6	14.5	7.2
Shen F1	4.4	6.8	1.6

Table 8: Result Stability: Range in model results from training with 5 different seeds each (UIRC test set).

Shen-F1 - This is a modified version of the F-1 metric for clusters which performs a "multi-way matching between conversations in two annotations" (Elsner and Charniak, 2010; Shen et al., 2006).

Precision, Recall, F1 Scores - These metrics are calculated based on the exact clusters (for cluster metrics), or links (for link metrics) that match in the prediction and the annotation, making this a very challenging metric to achieve.

A.13 Key Metrics

Free-text answers from direct prompting (the Pure rows) are mapped to an utterance index by a pinned decision model (mammoth-1, build 20261021). A rule-based parser and a majority vote of three parsers give scores within 0.5 Link F1.

		Link Level			Conversation (Cluster) Level						
Model	**Type**	**P**	**R**	**F1**	**P**	**R**	**F1**	**Scaled VI**	**ARI**	**NMI**	**Shen F1**
FeedForward	Sup	73.1	70.4	71.7	31.4	36.9	33.9	91.2	63.0	91.7	79.1
+ Assist @ 70%	Sup	79.4	76.5	77.9	**54.2**	**54.6**	**54.4**	95.7	85.0	**96.7**	**89.0**
+ Assist (DS)	Sup	79.3	76.5	77.9	53.1	52.4	52.8	95.7	**86.0**	96.4	88.9
Structure	Sup	74.6	71.9	73.3	36.8	45.6	40.8	92.7	65.4	91.7	83.9
+ Assist @ 90%	Sup	77.5	74.7	76.1	41.6	49.0	45.0	94.0	73.3	93.2	86.3
+ Assist (DS)	Sup	77.5	74.7	76.1	42.2	49.6	45.6	94.0	73.3	93.3	86.4
Zero-shot	Unsup	47.5	45.8	46.6	27.0	24.5	25.7	85.3	51.8	83.8	66.0
+ Assist @ 100%	Unsup	63.8	61.5	62.6	39.0	31.6	34.9	87.9	61.2	84.8	71.8
+ Assist (DS)	Unsup	63.1	60.8	61.9	35.7	25.1	29.5	86.1	60.1	81.9	65.9
Pure GPT-5	Unsup	68.9	66.5	67.7	24.1	15.8	19.1	87.2	60.1	84.7	65.2
Pure DeepSeek	Unsup	69.8	67.2	68.5	25.0	20.0	22.2	88.5	65.7	87.9	70.3
Pure GPT-4-Turbo	Unsup	68.2	65.7	66.9	27.5	30.1	28.8	87.9	57.5	90.7	70.8
E2E RL	Sup	**83.3**	**83.3**	**83.3**	51.2	52.3	51.9	**96.2**	78.5	–	–
DAG-LSTM	Sup	75.2	72.7	73.9	42.4	41.7	42.9	–	80.1	–	–
PointerNet	Sup	74.5	71.7	73.1	44.9	44.2	44.5	94.2	80.1	–	–
Dial-BERT	Sup	–	–	–	42.1	47.9	44.8	93.2	72.8	–	–
Co-Training	Unsup	–	–	–	–	–	–	–	18.2	54.0	45.6
Bi-CL	Unsup	–	–	–	–	–	–	–	34.5	60.7	70.4

Table 9: Comparison of Model Performance Metrics (+Assist rows use GPT-5 unless marked DS: DeepSeek V4.1 Flash), for 1) Models we applied prompt-based re-ranking on 2) Supervised Models 3) Unsupervised Models (Single Run) on the Ubuntu IRC dataset

		Link Level			Conversation (Cluster) Level						
Model	**Type**	**P**	**R**	**F1**	**P**	**R**	**F1**	**Scaled VI**	**ARI**	**NMI**	**Shen F1**
FeedForward	Sup	67.2	66.8	67.0	33.3	39.1	36.0	87.8	54.3	88.9	72.7
+ Assist @ 70%	Sup	72.8	72.4	72.6	52.8	46.7	49.6	91.5	61.5	90.2	77.1
+ Assist (DS)	Sup	73.0	72.6	72.8	52.8	46.7	49.6	91.4	61.5	90.1	76.7
Structure	Sup	64.4	64.0	64.2	31.7	35.9	33.7	90.2	76.5	90.7	79.8
+ Assist @ 90%	Sup	68.4	68.0	68.2	40.8	42.4	41.6	91.1	69.4	92.2	80.3
+ Assist (DS)	Sup	68.0	67.7	67.9	41.3	42.4	41.8	91.0	69.1	92.4	79.9
Zero-shot	Unsup	39.4	39.2	39.3	17.4	9.2	12.1	83.0	53.0	77.1	57.7
+ Assist @ 100%	Unsup	56.7	56.4	56.5	37.9	17.9	24.4	85.3	52.5	77.5	59.0
+ Assist (DS)	Unsup	56.2	55.9	56.0	34.5	16.3	22.1	84.7	52.0	76.7	57.8

Table 10: Results on the IRC New Domains dataset. DS: DeepSeek V4.1 Flash re-ranker.

		Link Level			Conversation (Cluster) Level						
Model	**Type**	**P**	**R**	**F1**	**P**	**R**	**F1**	**Scaled VI**	**ARI**	**NMI**	**Shen F1**
FeedForward	Sup	69.6	69.5	69.6	19.3	28.2	22.9	80.0	39.9	80.2	69.0
+ Assist @ 70%	Sup	77.7	77.6	77.7	30.8	30.8	30.8	84.2	47.6	80.9	67.6
+ Assist (DS)	Sup	77.3	77.2	77.3	34.3	30.8	32.4	83.9	48.5	81.0	68.8
Structure	Sup	69.0	68.9	69.0	17.1	35.9	23.1	79.1	32.7	75.9	64.2
+ Assist @ 90%	Sup	72.8	72.7	72.8	17.6	33.3	23.0	81.0	37.1	78.3	67.1
+ Assist (DS)	Sup	72.4	72.3	72.4	17.8	33.3	23.2	81.0	36.8	78.3	67.0
Zero-shot	Unsup	36.3	36.3	36.3	23.3	17.9	20.3	61.6	21.6	44.6	35.3
+ Assist @ 100%	Unsup	53.9	53.8	53.9	26.7	10.3	14.8	57.5	11.7	30.9	22.5
+ Assist (DS)	Unsup	53.2	53.1	53.2	42.9	15.4	22.6	50.1	2.2	14.4	16.7

Table 11: Results on the IRC Channel Two dataset. DS: DeepSeek V4.1 Flash re-ranker.

		Link Level			Conversation (Cluster) Level						
Model	**Type**	**P**	**R**	**F1**	**P**	**R**	**F1**	**Scaled VI**	**ARI**	**NMI**	**Shen F1**
FeedForward	Sup	89.7	89.7	89.7	32.0	14.0	19.5	87.5	53.1	82.8	60.1
+ Assist @ 70%	Sup	89.5	89.5	89.5	32.9	14.0	19.6	87.1	51.2	82.2	58.8
+ Assist (DS)	Sup	89.4	89.4	89.4	31.9	13.4	18.9	87.0	50.9	82.1	58.5
Structure	Sup	87.7	87.7	87.7	26.5	19.6	22.6	89.2	58.5	88.8	69.0
+ Assist @ 90%	Sup	88.1	88.1	88.1	27.4	19.1	22.5	89.1	57.9	88.0	68.1
+ Assist (DS)	Sup	88.0	88.0	88.0	26.9	18.8	22.1	89.1	57.9	88.0	68.0
Zero-shot	Unsup	39.3	39.3	39.3	19.7	11.6	14.6	85.8	50.9	84.0	59.7
+ Assist @ 100%	Unsup	59.9	59.9	59.9	31.7	13.7	19.1	87.2	51.4	82.4	59.0
+ Assist (DS)	Unsup	58.2	58.2	58.2	31.2	13.4	18.8	87.0	51.1	82.2	58.7

Table 12: Results on the Dramatic Dialogue dataset. DS: DeepSeek V4.1 Flash re-ranker.

A.14 Model Characteristics

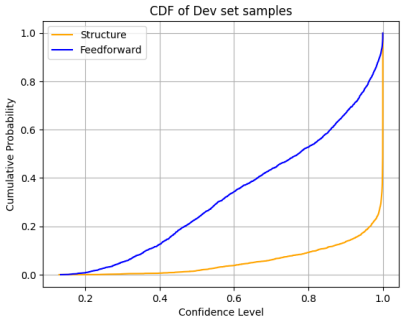


Figure 2: Cumulative Distribution Function of development set samples over FF and ST confidence levels (Ubuntu IRC)

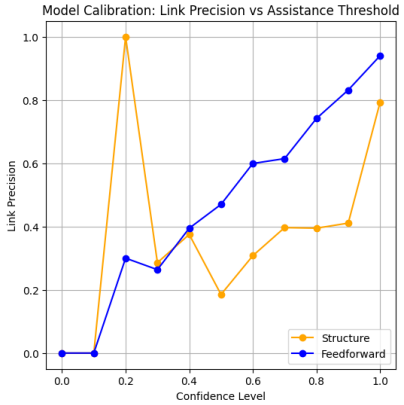
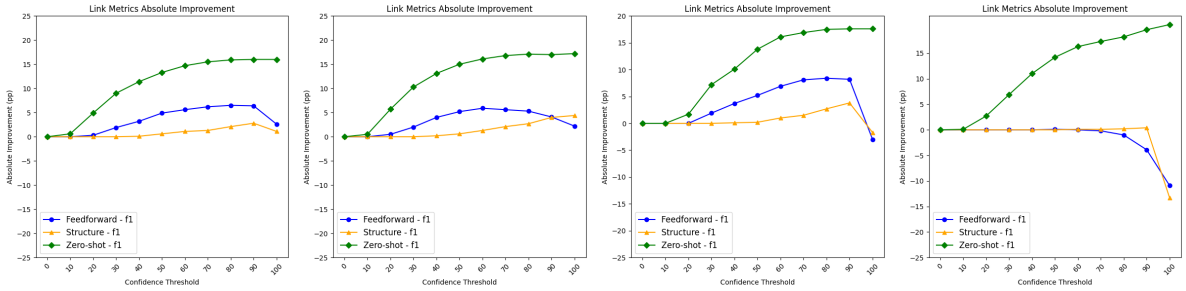
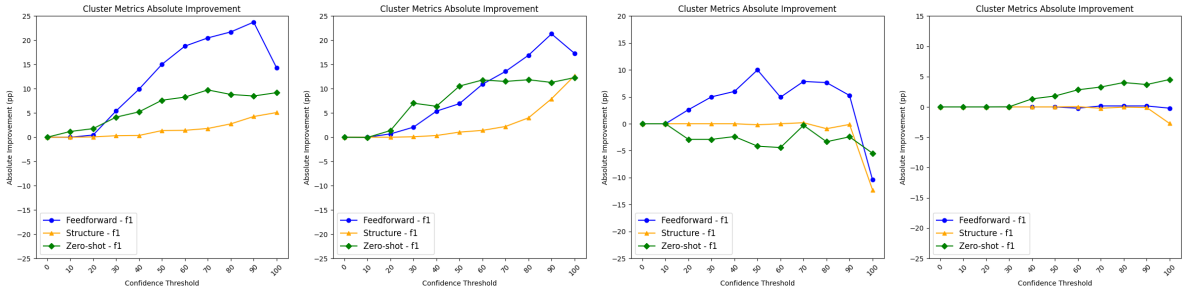


Figure 3: Link Precision on development set over FF and ST confidence levels (assistance threshold) (Ubuntu IRC)



(a) Ubuntu IRC (b) New Domains IRC (c) Channel Two IRC (d) Dramatic Dialogue

Figure 4: Comparison of link F1 improvements vs Confidence across datasets



(a) Ubuntu IRC (b) New Domains IRC (c) Channel Two IRC (d) Dramatic Dialogue

Figure 5: Comparison of cluster F1 improvements vs Confidence across datasets

A.15 Prompts

The prompts used in our experiments are shown below, including the direct zero-shot prompt used for prompting in isolation (Fig. 11).

Role

You are a linguist annotating some data that come from a Ubuntu IRC Chat, which involve lots of chats about IT related questions.

Instructions

You have the following query message

[02:27] <L1nuxRules> like I said it would need scripting to do waht you want i.e restrict it to packages, you can only restrict by repo at PM level unless things have changed in new distros Im old skool

Your job is to determine the index of the most relevant message it would be replying to as part of a conversation. Think through your response step by step, thinking about the motivations of query message in relation to each possible parent message.

Remember, you must provide the index.

Choices of parents

0: [02:26] <hammoommah> apparently I have 2 screenasaver deamons running.... lubuntu default and gnome deamon
1: [02:26] <bonhoffer> lsof -i :3000 worked too
2: [02:26] <bonhoffer> works
3: [02:26] <bonhoffer> L1nuxRules, thanks!
4: [02:26] <Tony> I only want the NGINX packages from that repository. how do I do that?

Original Conversation Context

[*Original conversation in chronological order. Length of context varies depending on the experiment variant*]

Figure 6: Prompts: Top 5 option chooser for re-ranker (Zeroshot Prompt)

ReAct Prompt

Instructions

You are an linguistic expert on conversational analysis who understands Ubuntu IRC chat jargon, technical terms, and abbreviations. You will be given a query message and some messages sent before the query. Your task is to determine the most likely message the query is responding to. You will be given the last {context_size} messages before the query message. In your answer you will consider:

1. The users' intentions
2. Technical commands and their meaning
3. Patterns of online communication in a tech support channel

If the query message is a system message, consider it as replying to itself as it is not related to any conversation.

Think through you answer step by step. Giving your thoughts, actions, observations at each step of the way like so:

Thought: ...
Action: ...
Observation: ...
Thought: ...
...

As the last line of your response after your reasoning, quote the parent message and its index

Formats

Normal Messages appear like:

[07:52] <jushur> OlofL: that way you dont mess up your current config.

where 07:52 is the time the message was sent, jushur is the user sending the message, O1ofL is the user the user is replying to.

System messages appear with "===" at the start of the message:

=== Guest58747 is now known as Captain_Wind_X

Context

{context}

Query

{query}

Figure 7: Prompts: ReAct prompt

Demo 1

Choices of Parents

0: [14:04] <TekGuy> Try 'sudo systemctl restart NetworkManager'\\
1: [14:04] <Jonsie> yes, Ubuntu 20.04\\
2: [14:04] <TekGuy> are you using Ubuntu 16.04 or newer?\\
3: [14:04] <TekGuy> let me know if anything goes wrong\\
4: [14:04] <Jonsie> ok, trying now\\

Original conversation context

[14::03] <Lara> how do I restart the network service?
[14::03] <TekGuy> sudo systemctl restart networking
[14::04] <Jonsie> it says 'Failed to restart networking.service: Unit networking.service not found.'
[14::04] <TekGuy> are you using Ubuntu 16.04 or newer?
[14::04] <Jonsie> yes, Ubuntu 20.04
[14::04] <kyle_b> perrytech: have you checked the official Ubuntu documentation on systemd? ...
=== NovaUser [n=jane@203-45-76-59.dyn.iinet.net.au] has joined \#ubuntu
[14::04] <TekGuy> Try 'sudo systemctl restart NetworkManager'
[14::04] <TekGuy> let me know if anything goes wrong
[14::04] <Ella> kurt, for disk management, gparted is also user-friendly if you prefer a GUI.
[14::04] <Jonsie> ok

Analysis

To determine the most relevant parent message for the query message "[14:04] <Jonsie> ok, trying now" we need to analyze the context of the conversation and the content of potential parent messages.

Here's a step-by-step analysis:

1. **Message 0: [14:04] <TekGuy> Try 'sudo systemctl restart NetworkManager'** – TekGuy suggests a command to restart the network service using a different service name, directly addressing the problem Jonsie encountered earlier when the command failed.
2. **Message 1: [14:04] <Jonsie> yes, Ubuntu 20.04** – This is a reply to TekGuy's earlier query about the Ubuntu version. This is not the correct link as their conversation has moved past the point of agreement about the ubuntu version to the command.
3. **Message 2: [14:04] <TekGuy> are you using Ubuntu 16.04 or newer ?** – This is an earlier question from TekGuy asking about the Ubuntu version, which Jonsie answers.
4. **Message 3: [14:04] <TekGuy> let me know if anything goes wrong** – This message from TekGuy follows his suggestion to try a new command. It is an instruction to report back if there are issues.
5. **Message 4: [14:04] <Jonsie> ok** – This is the message in question.

Given the context of the conversation:
- **Jonsie** is asking TekGuy how to fix his network service
- **kyle_b** seems to be engaged in another conversation with perrytech which is unrelated to Jonsie
- **Ella** is chatting with kurt about a separate disk management issue, unrelated to Jonsie

Analyzing the flow of conversation, TekGuy gives a command to try in message 0. Right after this, he instructs Jonsie in message 3 to report any issues. The response from Jonsie, "ok" follows logically as an acknowledgment and action step in response both of Tekguy 's messages. Since Tekguy has elaborated on 'Try 'sudo systemctl restart NetworkManager" with 'let me know if anything goes wrong', and Jonsie is not mentioning anything specific to the command suggested to fix his situation, therefore we would prefer the most recent message from tekguy '<TekGuy> let me know if anything goes wrong' in this case.

Therefore, the most likely parent message to Jonsie's "ok" is:

Index 3: [14:04] <TekGuy> let me know if anything goes wrong

Figure 8: Prompts: Few-shot re-ranker - Example 1

Demo 2

Choices of Parents

0: [08:10] <gamma_ray> has anyone optimized Firefox to use less memory?
1: [08:11] <gamma_ray> wow, lots of people using this channel!
2: === J_Davidson [n=james@192.168.1.100] has joined #ubuntu
3: [08:10] <PyDev> looking into swap adjustments, thanks
4: [08:10] <TechBot> !memory

Original conversation context

=== LunaTech [n=sarah@connect.lunavpn.org] has joined #ubuntu
[08:10] <CodeMaster> why does my computer freeze when I open too many tabs?
[08:10] <TechBot> !memory
[08:10] <ubotu> Sorry, I don't know anything about memory – try searching on http://bots.ubuntulinux.nl/factoids.cgi
[08:10] <TechBot> !swap
[08:10] <ubotu> Swap space is used when the RAM is full. For information on adjusting swap space or managing system memory, visit: http://help.ubuntu.com/community/SwapFaq
=== J_Davidson [n=james@192.168.1.100] has joined #ubuntu
[08:10] <gamma_ray> has anyone optimized Firefox to use less memory?
[08:10] <PyDev> looking into swap adjustments, thanks!
=== starfall [n=ella@star.networks.org] has joined #ubuntu
[08:11] <gamma_ray> wow, lots of people using this channel!

Analysis

To determine the most relevant parent message for the query message "[08:11] <gamma_ray> wow, lots of people using this channel!" we need to analyze the context of the conversation and the content of potential parent messages.

Here's a step-by-step analysis:

1. **Message 0: [08:10] <gamma_ray> has anyone optimized Firefox to use less memory?** – This message by gamma_ray asks the community if anyone has experience or tips on reducing the memory usage of Firefox. This is likely a follow-up or related query to ongoing discussions about system performance or specific software optimization.
2. **Message 1: [08:11] <gamma_ray> wow, lots of people using this channel!** – Gamma_ray makes an observation about the number of participants or activity level on the channel. This message serves more of a social function, possibly to express surprise or to initiate small talk with the community.
3. **Message 2: === J_Davidson [n=james@192.168.1.100] has joined #ubuntu** – A system or server-generated message indicating that a user, J_Davidson, has joined the Ubuntu channel.
4. **Message 3: [08:10] <PyDev> looking into swap adjustments, thanks** – PyDev acknowledges the information received (likely related to managing system memory via swap space) and indicates they are going to explore it further.
5. **Message 4: [08:10] <TechBot> !memory** – TechBot triggers a command within the IRC channel, likely designed to fetch and display automated responses or links about memory management.

Given the context of the conversation:
- **PyDev** has likely asked an earlier question to Techbot and techbot has helped them by triggering the ubotu bot
- **Techbot** seems to be a helpful answerer, helping both CodeMaster and PyDev by triggering ubotu commands
- **gamma_ray** is asking questions on how to improve the memory usage of their browser, which is related the other conversations between pydev and techbot

Analyzing the flow of conversation, gamma_ray asks a question in message 0, and acknowledges the large inflow of users in message 2. The intent of second message serves mainly a social purpose and is unrelated to
the main technical topic discussed on optimising memory earlier. Therefore '[08:11] <gamma_ray> wow, lots of people using this channel!' is a start of a new conversation rather than a follow up to '[08:10] <gamma_ray> has anyone optimized Firefox to use less memory?'
If a message is the start of a new conversation, then it is linked back to itself.

Therefore, the most likely parent message to "[08:11] <gamma_ray> wow, lots of people using this channel!" is:

Index 1: [08:11] <gamma_ray> wow, lots of people using this channel!

Figure 9: Prompts: Few-shot re-ranker - Example 2

Chain-of-thought Prompt

Instructions

You are an linguistic expert on conversational analysis who understands Ubuntu IRC chat jargon, technical terms, and abbreviations. You will be given a query message and some messages sent before the query. Your task is to determine the most likely message the query is responding to. You will be given the last {context_size} messages before the query message. In your answer you will consider:

1. The users' intentions
2. Technical commands and their meaning
3. Patterns of online communication in a tech support channel

If the query message is a system message, consider it as replying to itself as it is not related to any conversation.

As the last line of your response after your reasoning, quote the parent message and its index

Think through you answer step by step.

Formats

Normal Messages appear like:

[07:52] <jushur> OlofL: that way you dont mess up your current config.

where 07:52 is the time the message was sent, jushur is the user sending the message, O1ofL is the user the user is replying to.

System messages appear with "===" at the start of the message:

=== Guest58747 is now known as Captain_Wind_X

Context

{context}

Query

{query}

Figure 10: Prompts: Chain-of-thought prompt

LLM solo-zeroshot Prompt

Instructions

You are an linguistic expert on conversational analysis who understands Ubuntu IRC chat jargon, technical terms, and abbreviations. You will be given a query message and some messages sent before the query. Your task is to determine the most likely message the query is responding to. You will be given the last {context_size} messages before the query message. In your answer you will consider:

1. The users' intentions
2. Technical commands and their meaning
3. Patterns of online communication in a tech support channel

If the query message is a system message, consider it as replying to itself as it is not related to any conversation.

As the last line of your response, quote the parent message and its index

Give your answer concisely.

Formats

Normal Messages appear like:

[07:52] <jushur> OlofL: that way you dont mess up your current config.

where 07:52 is the time the message was sent, jushur is the user sending the message, O1ofL is the user the user is replying to.

System messages appear with "===" at the start of the message:

=== Guest58747 is now known as Captain_Wind_X

Context

{context}

Query

{query}

Figure 11: Prompts: LLM Solo Zeroshot prompt