Structure Over Semantics: A Controlled Re-Evaluation of Multi-Party Conversation Disentanglement

Ishan Katoch
Independent Researcher
India
ikatoch_be21@thapar.edu

Abstract

Hybrids of pre-trained language models and graph neural networks are standard for multi-party conversation disentanglement, and their gains are usually credited jointly to semantic embeddings and graph structure. We test that credit assignment on the public Kummerfeld et al. (2019) IRC corpus using one fixed GCN architecture in which only node features or edge types vary, five random seeds, and both of the oracle-K clustering protocols in common use. Node features matter in aggregate (ARI 0.401 vs. 0.062 for constant features), but which features are used does not: handcrafted position, length, question-mark, and author features show no significant difference from 1024-dimensional BERT-large embeddings across 5 seeds (0.401 vs. 0.400, $p = 0.40$), and concatenating them adds nothing. Removing gold reply-to edges costs nothing. Speaker-continuity edges alone outperform all edge types combined. Most importantly, a non-learned baseline that groups messages by author reaches ARI 0.4287, matching or exceeding every trained model here, because 63% of gold test threads contain a single message. We also correct an earlier version of this work, whose reported advantage for handcrafted features over BERT was an artifact of feature scaling. Reported gains on this benchmark reflect its evaluation setup at least as much as the models.

Keywords: Conversation Disentanglement, Graph Neural Networks (GNN), Reproducibility, Empirical Re-evaluation, Multi-party Chat Analysis, Natural Language Processing (NLP)

1 Introduction

Multi-party online discussions, such as chat rooms, forums, and customer support interactions, often include multiple concurrent conversational threads. A recent systematic review of conversational-AI datasets and techniques notes that thread disentanglement and context preservation remain open challenges in real-world deployments (Gite et al., 2024).

Consider an IRC channel where A asks how to install Python while, in the same window, C separately reports a crashing shell script: replies to A ("add the deadsnakes PPA") and replies to C ("shows a segmentation fault") interleave in one stream, and any system reading it in order will mix the two topics. This makes it hard to tell which message answers which, for humans and models alike (Elsner and Charniak, 2008; Mlakar et al., 2023; Liu et al., 2021). Applications that operate on live conversation, such as customer service and group chat moderation, need both accurate thread recognition and a judgment about which thread needs a response first (Mehri and Carenini, 2017; Tan et al., 2019; Zhang et al., 2022).

Early methods treated disentanglement as unsupervised clustering over lexical similarity measures such as term frequency–inverse document frequency (TF-IDF) and cosine similarity (Shen et al., 2006), capturing neither semantic context nor speaker-specific information. Elsner and Charniak (2008; 2010) moved the task onto annotated Internet Relay Chat (IRC) data with discourse-driven graph algorithms, an early sign that structure helps; adding speaker identities and temporal features improved on purely lexical models, though only by modest margins (Milon-Flores and Cordeiro, 2022; Mayfield et al., 2012; Chang and Ko, 2025).

Supervised deep learning then changed the field: attentive recurrent architectures such as DialogRNN (Majumder et al., 2019), reply-link predictors built on random forests and pointer networks (Mehri and Carenini, 2017; Yu and Joty, 2020), and hierarchical transformers (Vaswani et al., 2017) such as DialBERT (Li et al., 2020, 2024b).

Proceedings of the 38th Conference on Computational Linguistics and Speech Processing (ROCLING 2026), pages 360–370
November 6-7, 2026 ©2026 Association for Computational Linguistics

Graph Neural Networks (GNNs), including Graph Convolutional Networks (Kipf and Welling, 2017) and Graph Attention Networks (Veličković et al., 2018), represent conversational structure directly (Yelamarthy and Anand, 2023; Bhukar et al., 2023; Fan et al., 2023; Park et al., 2022; Mao et al., 2024), though almost all of this work stops at disentanglement without addressing prioritization (Kevin Joshua T et al., 2024; Zhang et al., 2022). Data constrained progress too: early resources such as the Ubuntu Dialogue Corpus (Lowe et al., 2015) had inconsistent annotation, until Kummerfeld et al. (2019) released a 77,000-message annotated IRC corpus that gave the field a reliable benchmark.

Combinations of a PLM with a GNN, of the kind we study here, have been applied to disentanglement before (Li et al., 2024a; Bhukar et al., 2023), usually presented as jointly crediting semantic embeddings and graph structure for reported gains. We test that credit assignment directly. We build a conversation graph with reply-to, temporal-adjacency, and speaker-continuity edges, and compare a GCN over this graph fed either handcrafted structural features or 1024-dimensional BERT-large embeddings, on the official Kummerfeld splits with five random seeds. If BERT's representations are doing the work generally credited to them, the BERT-fed variant should clearly outperform the handcrafted-feature variant; if graph structure is doing the work instead, the two should be close, and removing gold reply supervision should barely matter. We report which holds, and audit whether recently reported NMI/ARI numbers on this corpus can be compared to each other at all.

The primary contributions of this work are:

- **A matched comparison isolating features from structure.** One GCN architecture, one training schedule, five seeds, with only node features or edge types varying, reported under both clustering protocols in use on this corpus. Handcrafted features and BERT-large show no statistically significant difference under the per-file protocol (five seeds), and gold reply-to edges are unnecessary.
- **A non-learned baseline that bounds the benchmark.** Grouping messages by author, with no model at all, reaches ARI 0.4287 and matches or exceeds every trained model we evaluate. We trace this to the corpus itself, in which 63% of gold test threads are single messages, and show that the strongest graph configuration matches its ARI to four decimal places.
- **A correction and a protocol audit.** We correct a feature-scaling artifact and a protocol mismatch in an earlier version of this work, and show that cross-paper NMI/ARI figures on this corpus differ by protocol choices larger than the differences between the methods being compared (Section 6).

2 Related Work

2.1 Disentanglement: From Lexical Clustering to Graphs

Early work treated disentanglement as unsupervised clustering over lexical similarity (Shen et al., 2006). Elsner and Charniak (2008; 2010) moved the task onto annotated IRC data with discourse-driven graph algorithms, and later work added speaker identity, timing, and topic features (Mayfield et al., 2012; Milon-Flores and Cordeiro, 2022; Tan et al., 2019), then supervised neural models: RNN/LSTM classifiers (Mehri and Carenini, 2017), pointer networks (Yu and Joty, 2020), and transformer encoders such as DialBERT (Li et al., 2020). GNN-based methods made structure explicit rather than implicit in features (Yelamarthy and Anand, 2023; Bhukar et al., 2023; Fan et al., 2023; Huang et al., 2022), with Mao et al. (2024)'s Hierarchical Graph Fusion Network and Chang and Ko (2025)'s pseudo-initiative strategy both pointing to participant initiative as a useful signal, one our speaker-continuity edges also encode. Tan et al. (2023) and Li et al. (2024a) probe prompted PLMs and discourse-graph losses respectively; recent work generally pairs semantic and structural modeling, which our framework follows while also sharing one representation across disentanglement and prioritization. Larger annotated corpora (Kummerfeld et al., 2019; Lowe et al., 2015; Chang et al., 2023; Gouravajhala et al., 2023; Shang et al., 2024) made this progress possible, though a recent review still finds gaps in multi-party annotation quality (Gite et al., 2024), and no single corpus covers every conversational format.

2.2 Response Prioritization

Prioritization gets far less attention than disentanglement itself, yet it is what decides which message a support agent or moderator actually sees first. Early approaches, such as Kevin Joshua T

et al. (2024), relied on simple heuristics like waiting time. Later work moved toward learned signals instead: model uncertainty, embedding distances, and urgency cues, combined rather than hand-tuned (Zhang et al., 2022; Kumar et al., 2023). Milon-Flores and Cordeiro (2022) show that behavioral features can flag online grooming early in a chat stream, where isolating the right thread quickly enables intervention.

3 Dataset

We use the Conversation Disentanglement Corpus of Kummerfeld et al. (2019): 77,563 IRC messages with explicit reply-to annotations and full chronological order preserved. The official splits (Section 4.4) cover its 74,963 #Ubuntu messages; the other 2,600 (#Linux) are unused. Messages skew short, and same-thread versus cross-thread cosine similarity distributions overlap substantially, motivating supervised, context-aware methods over similarity thresholding alone. This corpus is markedly larger and more consistently annotated than earlier resources (Elsner and Charniak, 2008; Mehri and Carenini, 2017; Gu et al., 2022), though the Ubuntu Dialogue Corpus (Lowe et al., 2015) and newer resources spanning dramatic scripts and multi-domain chat (Chang et al., 2023; Gouravajhala et al., 2023) offer useful, if noisier, complements.

Thread-size distribution. One property of the test split matters for every result that follows and is, to our knowledge, not stated in prior work using this corpus: the gold threads are overwhelmingly short. Of the 961 test threads, 63.1% contain a single message and 69.5% contain at most two; the median thread has one message and one speaker. Correspondingly, 69.3% of threads involve exactly one speaker and 67.5% of speakers appear in exactly one thread. Section 5.1 shows that this makes speaker identity a near-complete proxy for thread membership, which in turn governs how any result on this corpus should be read.

Synthetic data. A secondary check (Section 5.6) uses synthetic IRC-style data whose generator is deterministic given its seed and released with the code. It draws on eight hand-written technical-support topic templates (Python installation, segmentation-fault debugging, Vim configuration, Git rebase, Docker networking, SSL certificates, screen resolution, pip dependency conflicts), each a 5–7 message question-answer exchange. Each session samples five topics and assigns three of six named speakers per topic. Messages are interleaved by repeatedly sampling an unfinished thread with probability proportional to its remaining messages, with a 0.3 chance after each message of immediately emitting that thread's next message to create bursts. Fifty sessions give 1,476 messages across 250 threads. Because threads are lexically clean and non-overlapping by construction, this setting is easier than the real corpus and is used only to check that relative patterns persist.

We use the official corpus release exactly as distributed, with 67,463 training messages, 2,500 development messages, and 5,000 test messages (961 gold threads in the test split); no thread crosses a partition. All results in Section 5 use the held-out test split. The development split is not used for hyperparameter selection in this study, since our goal is a controlled comparison under one fixed configuration rather than a tuned leaderboard entry, and we state this explicitly rather than imply tuning that did not happen.

4 Methodology

We compare six model families, ranging from unsupervised clustering to an integrated hybrid combining a GNN with PLM embeddings.

4.1 Clustering Baselines, Siamese, and Triplet Models

The simplest benchmarks are K-Means over two feature types, TF-IDF vectors and BERT (Devlin et al., 2019) [CLS] embeddings (`bert-base-uncased` and `bert-large-uncased`), with K fixed to the true thread count. A Siamese network goes further. For a message pair (m_i, m_j), it asks a single question: do these two share a thread? Both messages pass through a frozen BERT-large encoder, and the resulting similarity score is squashed through a sigmoid, $y_{ij} = \sigma(\text{sim}(f(m_i), f(m_j)))$, and trained with binary cross-entropy. At inference time, an edge connects any pair whose predicted probability clears a threshold, and connected components become the predicted threads. The triplet-loss model takes a different route entirely: rather than classifying pairs, it learns an embedding space where same-thread messages sit closer together than different-thread ones, starting from a BERT-base (768-dimensional) initialization and optimizing the margin loss $\mathcal{L}_{\text{triplet}} =$

$\sum_{(a,p,n)} \max\{d(m_a, m_p) - d(m_a, m_n) + \alpha, 0\}$, where d is Euclidean distance and α the margin. K-Means clustering on the resulting embeddings gives the final thread assignment.

4.2 Graph Neural Network Model

We want conversational structure to be visible to the model directly, not inferred from features, so we build an interaction graph $G = (V, E)$ per conversation with three edge types: reply-to (from the corpus annotation), temporal adjacency (consecutive messages in a session), and speaker continuity (same author, same session, at any distance apart). Edges are made undirected and symmetrically normalized.

One architecture throughout. Every graph-based result in this paper uses the *same* two-layer GCN: a linear projection to 256 hidden units, symmetric-normalized neighbourhood aggregation, ReLU, dropout 0.3, and a linear output layer, trained with Adam at 1×10^{-3} for 50 epochs. Model variants differ *only* in which node features enter the first layer and which edge types are present in the graph. Holding the architecture, optimizer, and hidden width fixed is what licenses the feature and edge comparisons in Section 5; no attention-based variant is trained or reported.

The training objective, and its limits. The network is trained as a node classifier over thread slots, where slots are numbered per file by order of first appearance. This objective is weak by construction: slot 3 denotes a different conversation in every file, so a single shared output layer is asked to predict labels whose meaning is not consistent across the training set. We report its accuracy for completeness but do not interpret it, and we take all disentanglement metrics from clustering the hidden layer rather than from the classifier head. A contrastive or pairwise objective would be better matched to the task and is the most direct extension of this work.

4.3 Hybrid GNN+PLM (BERT) Approach

The hybrid model is the architecture of Section 4.2 with one substitution: each node is initialized with a 1024-dimensional BERT-large [CLS] embedding in place of handcrafted features. Graph construction, hidden width, optimizer, schedule, and evaluation are identical, so any difference between the two is attributable to the node features alone.

The hybrid approach proceeds through the following stages:

1. Graph and Node Features: Each message is a node $v \in V$ in the per-conversation graph $G = (V, E)$ described above, initialized with its BERT-large [CLS] embedding (1024-dimensional).
2. Graph Neural Network Processing: Node embeddings are propagated through a two-layer GCN architecture. At each GCN layer, the embedding for a node v at a layer $l + 1$ aggregates embeddings from neighboring nodes $u \in N(v)$ and itself: $h_v^{(l+1)} = \sigma(\sum_{u \in \tilde{N}(v)} \frac{1}{\sqrt{\tilde{d}_v \cdot \tilde{d}_u}} W^{(l)} h_u^{(l)})$, where $\tilde{N}(v) = N(v) \cup \{v\}$ is the neighborhood of v augmented with a self-loop, $\tilde{d}_v = |\tilde{N}(v)|$ the corresponding degree, $\sigma(\cdot)$ ReLU, and $W^{(l)}$ the layer-specific learnable weight matrix. This symmetric normalization follows Kipf and Welling (2017) and ensures stable aggregation across nodes of varying degree.
3. Training Objective: The model is trained end-to-end to classify nodes into their conversation threads with the standard cross-entropy loss $\mathcal{L} = -\sum_{v \in V} \sum_{t=1}^{T} y_{v,t} \log(\hat{y}_{v,t})$, where $y_{v,t}$ is the true label for node membership in thread t and $\hat{y}_{v,t}$ the GNN's predicted probability, using Adam at learning rate 1×10^{-3} for 50 epochs.

The overall hybrid architecture is visualized in Figure 1. Hyperparameters: 2 GCN layers, ReLU activation, dropout 0.3, Adam at lr 1×10^{-3} for 50 epochs, batch size 32, mean-pooling aggregation, one message-passing step per layer, on the 67,463/2,500/5,000-message train/dev/test split (Section 3).

The refined node embeddings after graph propagation are used for three downstream applications:

- **Reply Link Prediction:** Determining reply relationships between pairs of messages by classifying edges in the graph; we do not separately report edge-classification accuracy for this hybrid model, and the only pairwise link-prediction number in this paper (Section 5.5) belongs to a separate, non-graph Siamese baseline.
- **Thread Disentanglement:** Assigning each message to its appropriate conversation thread via direct classification or clustering techniques applied on refined embeddings.
- **Response Prioritization:** Each thread t gets a priority score $s(t) = w_1 q(t) + w_2 c(t) +$

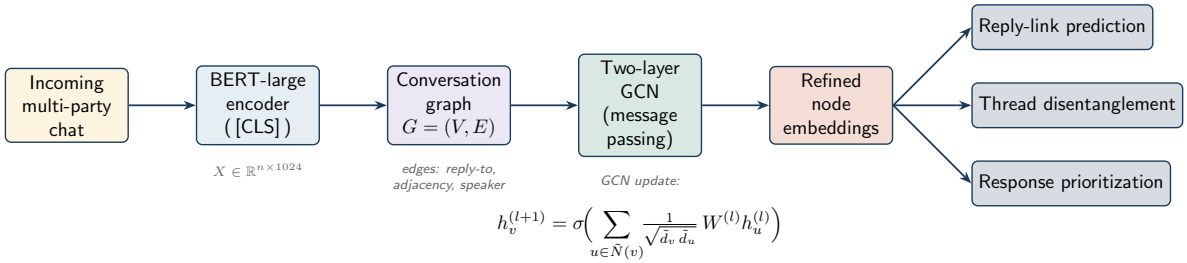


Figure 1: End-to-end architecture of the proposed hybrid GNN+PLM (BERT) framework. Incoming multi-party messages are encoded by BERT-large ([CLS] token) into an embedding matrix $X \in \mathbb{R}^{n\times 1024}$; a conversation graph is built with reply-to, adjacency, and speaker edges; a two-layer GCN propagates relational context; and the refined node embeddings feed three downstream tasks: reply-link prediction, thread disentanglement, and response prioritization.

$w_3 u(t)$, where $q(t)$ flags an unanswered question, $c(t)$ is mean GNN thread-assignment confidence, and $u(t)$ is a keyword-based urgency score (e.g., *urgent*, *ASAP*, *error*); weights are fixed by design, not tuned, and threads are ranked by descending $s(t)$. Section 6.3 discusses evaluating this module rigorously.

4.4 Reproducibility Protocol

Because this study's central claim rests on comparing model variants fairly, every choice that could otherwise be left implicit is fixed and reported here.

- **Corpus and splits:** the official Kummerfeld et al. irc-disentanglement release, used as distributed, with no re-splitting. Train-split thread labels are derived from the annotated reply-to links via union-find, since the official train release does not ship pre-computed cluster IDs; dev and test splits use the corpus's own gold cluster files directly.
- **Embedding freezing:** BERT-base and BERT-large are used as frozen encoders throughout (no fine-tuning), so any performance difference between the BERT-fed and simple-feature GCN variants is attributable to the features themselves, not to differential training.
- **Graph construction:** reply-to edges come from the corpus annotations; temporal-adjacency edges connect each message to its immediate predecessor and successor within a channel session; speaker-continuity edges connect all message pairs by the same author within a session. All edges are made symmetric and normalized following Kipf and Welling (2017) (Section 4.3).
- **Cluster count (K) and clustering scope.** Two oracle-K protocols are possible on this corpus and they are not interchangeable. Under the *global* protocol, all 5,000 test messages are clustered at once into the corpus-wide thread count (961). Under the *per-file* protocol, each of the ten test files is clustered separately into that file's own thread count, which hands the model the file partition for free and is therefore substantially easier. Both use scikit-learn K-Means (10 initialisations, seeded with the training seed) on the GCN's 256-dimensional hidden layer, unrescaled. An earlier version of this work mixed these two protocols in a single table, confounding the comparison; every model here is now reported under *both* (Table 2), and cross-model claims are made only within a protocol.
- **Seeds:** every learned model is trained and evaluated under five random seeds (42–46); we report the mean and sample standard deviation across seeds for every metric in Section 5, rather than a single run.
- **Gold-edge ablation:** because reply-to edges are themselves part of what a real disentanglement system would need to predict, we report the hybrid GCN both with gold reply edges present in the graph (the literal setting used in Sections 4.2–4.3) and with them removed, leaving only adjacency and speaker edges, to check how much of the result depends on supervision that would not be available at deployment time.

5 Experiment Results

All models are evaluated on the official test split. We report Adjusted Rand Index (ARI) for every model, and additionally Normalized Mutual Information (NMI) for the non-learned baselines (Table 1); learned models are evaluated under

Baseline (no model)	ARI	NMI
Group by author, within file	0.4287	0.8655
Group by author, across files	0.4163	0.8589
Whole file as one cluster	0.1264	0.5732
Contiguous blocks (oracle K)	0.0927	0.7785
Random into oracle K	0.0189	0.6680
Every message its own cluster	0.0000	0.8045

Table 1: Non-learned baselines on the official test split. Grouping messages by author alone matches or exceeds every trained model in Table 2. These are deterministic, so no variance is reported.

both clustering protocols defined in Section 4.4. Learned models are run under five random seeds (42–46) and reported as mean ± standard deviation; non-learned baselines are deterministic.

5.1 Non-Learned Baselines

We begin with baselines that involve no model, training, or features, because they bound everything that follows (Table 1). Assigning every message to a cluster according to *who wrote it* reaches ARI 0.4287 within files and 0.4163 across them, matching or exceeding every trained model we evaluate, including the BERT-based hybrid (Section 5.2).

The explanation lies in the corpus statistics reported in Section 3: 63.1% of gold test threads contain a single message, 69.3% involve a single speaker, and 67.5% of speakers appear in exactly one thread. Under these conditions authorship is very nearly a complete description of thread membership, and a clustering metric computed against oracle K rewards recovering it. This is a property of the evaluation, not a claim that disentanglement is trivial in general: real deployments do not know K, and the interesting threads are the long multi-speaker ones making up the remaining third of the data.

5.2 Matched Comparison of Node Features

Table 2 varies node features while holding the architecture, graph, optimizer, and evaluation fixed, which is the comparison Reviewer feedback on an earlier version of this work correctly identified as missing.

Three results follow. First, node features matter enormously in aggregate: replacing informative features with a constant collapses ARI from 0.401 to 0.062 (paired t over seeds, $p < 10^{-4}$). A graph with uninformative nodes recovers almost nothing. Second, and in direct contrast, *which* informative features are used makes no detectable difference over five seeds. Handcrafted features (position, length, question-mark presence, hashed author) reach 0.401 ± 0.003 and 1024-dimensional BERT-large embeddings reach 0.400 ± 0.003; the difference is $+0.0017$ with $p = 0.40$. Concatenating both adds $+0.0008$ ($p = 0.63$). An earlier version of this work reported handcrafted features *outperforming* BERT-large (0.411 vs. 0.363); that gap was an artifact of feeding unstandardised BERT embeddings to the first linear layer, and it disappears once both feature types are standardised with train-split statistics. Third, removing gold reply-to edges at test time costs nothing: $+0.0007$ for handcrafted features ($p = 0.80$) and -0.0001 for BERT ($p = 0.97$). The label-leakage concern that motivated this experiment does not materialise.

5.3 Which Edges Carry the Signal

Table 3 holds features fixed and varies the edge set. Speaker-continuity edges alone (0.428) outperform all three edge types together (0.401); temporal adjacency alone reaches 0.125 and gold reply-to edges alone 0.275. Adding edges beyond speaker continuity does not help and slightly hurts.

The speaker-only configuration deserves closer attention, because it does not learn anything. Speaker-continuity edges make each author's messages a clique; symmetric-normalised aggregation therefore assigns every message by a given author in a given file an identical representation, so K-Means can separate authors but never split one author's messages. The global-protocol ARI is consistent with the graph reproducing the author grouping, though it does not establish identical assignments: the speaker-only model scores ARI 0.4287 ± 0.0000 under the global protocol, identical to four decimal places to the within-file non-learned author partition in Table 1, with zero variance across five seeds; under the per-file protocol, where K need not equal the number of authors, it scores 0.428 ± 0.000 (Table 3). Under the global protocol, the best graph configuration here behaves as a deterministic re-derivation of `group-by-author`.

5.4 Effect of Conversation Shape

We also asked whether performance depends on how much speakers repeat within a session. Splitting the ten test files at the median (6.06 messages per unique speaker) gives mean ARI 0.355 for low-repetition files and 0.378 for high-repetition files, with Spearman $\rho = +0.088$ ($p = 0.55$) across 50 file–seed pairs, in the expected direction but not statistically reliable at this sample size, so we report

	Per-file oracle K		Global oracle K	
Node features	With gold	No gold	With gold	No gold
Constant (no features)	0.062 ± 0.000	0.079 ± 0.002	0.015 ± 0.000	0.034 ± 0.002
Handcrafted	0.401 ± 0.003	0.401 ± 0.008	0.377 ± 0.005	0.383 ± 0.006
BERT-large	0.400 ± 0.003	0.400 ± 0.002	0.392 ± 0.005	0.402 ± 0.004
Handcrafted + BERT-large	0.402 ± 0.002	0.403 ± 0.003	0.402 ± 0.001	0.406 ± 0.005
No graph (K-Means on frozen features)				
K-Means, BERT-large	0.044 ± 0.001		0.007 ± 0.001	
K-Means, TF-IDF	0.116 ± 0.006		0.001 ± 0.001	

Table 2: Matched comparison: one GCN architecture, identical training, features standardised with train-split statistics. *With gold* uses reply-to + adjacency + speaker edges; *No gold* drops reply-to edges. Mean $\pm$ std over 5 seeds. The K-Means rows use no graph structure, features only.

Edges present	**ARI**
Speaker continuity only	0.428 ± 0.000
Temporal adjacency only	0.125 ± 0.004
Gold reply-to only	0.275 ± 0.005
Temporal + speaker	0.401 ± 0.008
All three	0.401 ± 0.003

Table 3: Edge ablation, handcrafted features held fixed. Speaker-continuity edges alone account for the entire effect. ARI, mean $\pm$ std over 5 seeds, per-file oracle K.

it as a null result rather than a finding.

5.5 Other Architectures

For completeness, two non-graph supervised models were also evaluated. A Siamese pairwise link predictor over frozen BERT-large reaches $76.3\% \pm 0.8$ pair accuracy and, after reconstructing threads by connected components, NMI 0.657 ± 0.003 and ARI 0.146 ± 0.001. A triplet-loss metric learner reaches ARI 0.014 ± 0.001. Both fall well below the author baseline.

5.6 Secondary Check on Synthetic Data

We repeated the feature comparison on synthetic IRC-style data with unambiguous ground truth (generator described in Section 3; 1,476 messages, 250 threads, 50 sessions). Five embedding backbones spanning BERT-base to E5-small all cluster within a one-point NMI band of each other (0.446–0.453) without graph structure (Figure 2a), and adding a single message-passing step raises NMI to 0.661 (Figure 2b). This NMI pattern matches the real corpus: embedding choice barely moves the result, while adding structure moves it substantially. ARI on this synthetic split, however, stays near zero in both conditions, plausibly because the generator's thread sizes are more balanced than the real corpus's heavy singleton-thread skew (Section 3); we flag this rather than omit it. We report this only as a secondary consistency check, not independent evidence; the real-corpus results in Table 2 are what the paper's claims rest on.

6 Comparative Evaluations

Can our numbers be compared to prior work at all? This section argues that, for the most part, they cannot be compared as directly as a single table implies, and shows why: published results on this task differ not just in method but in dataset split, cluster-count strategy, and scoring granularity, and these protocol choices move NMI and ARI by margins as large as the differences between methods.

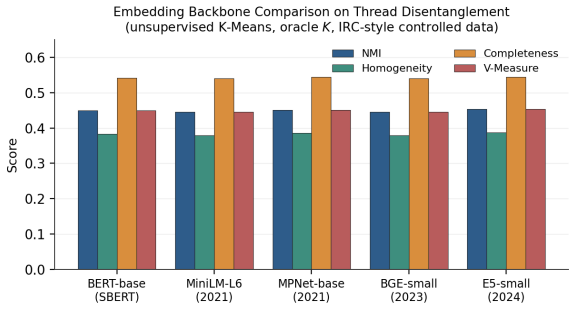
6.1 Dataset and Evaluation Differences

Elsner and Charniak (2008) reported approximately 70% thread-level F1 on a corpus of roughly 2,500 messages, and Mehri and Carenini (2017) evaluated on approximately 5,000 annotated messages; scale alone, however, does not make metrics comparable across papers. Kummerfeld et al. (2019) evaluate with variation of information, one-to-one overlap and conversation F1 rather than NMI or ARI, so their results do not sit on our scale. Our Siamese thread-reconstruction result (NMI = 0.657 $\pm$ 0.003, ARI = 0.146 $\pm$ 0.001, Section 5) is low on ARI, most plausibly because our thread reconstruction uses a fixed-window connected-components heuristic rather than a tuned clustering step, not because the underlying link predictor is weaker.

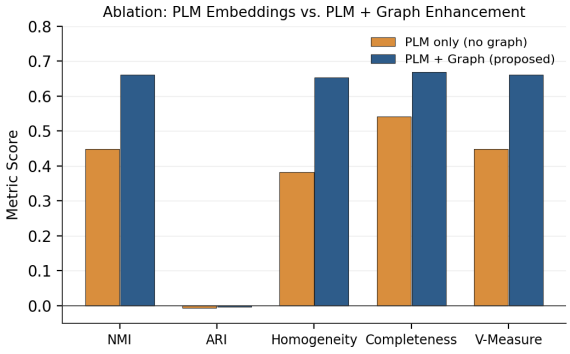
6.2 Result Comparison

The proposed hybrid GNN+PLM (BERT) approach is compared with some existing approaches and is presented in Table 4.

Table 4 lays out the field's published record; it reads as a leaderboard but argues against one. StructBERT (Ma et al., 2022) and Bi-CL (Huang et al., 2022) report higher NMI than any oracle-K result here, but on a different Ubuntu IRC split with unstated preprocessing (‡). Lam & Yang's (2025) ARI uses a link-based variant, not corpus-level clustering ARI. Takada & Mori's (2026) Gemini-2.5-pro DLA+SC figures come from a single zero-shot run, with the scoring granularity unstated. Protocol



(a) Embedding backbone comparison. Without graph structure (oracle K K-Means), all five backbones cluster within a narrow NMI band ($\approx$0.44–0.45); see (b) for the effect of adding structure on this same data.



(b) Effect of graph neighborhood aggregation. Adding one message-passing step to MPNet embeddings yields a 47% relative NMI improvement (0.449 $\rightarrow$ 0.661); ARI stays near zero in both conditions (see text).

Figure 2: Secondary check on synthetic data. Embedding choice barely moves clustering quality (a), while adding graph structure raises NMI substantially (b); this NMI pattern matches the real corpus, though ARI on this synthetic split does not, as discussed in the text.

effects of this kind are visible within this paper alone: moving the same handcrafted-feature GCN from per-file to global oracle K changes ARI by 0.024 (Table 2), more than ten times the 0.0017 gap between handcrafted and BERT-large features under per-file K. Most rows in Table 4 are consequently not answerable by "which method wins": they are not measuring the same thing.

Our matched comparison in Table 2 uses identical graphs, training, and clustering settings under both protocols. The near-tie tested for significance ($p = 0.40$, Section 5.2) is under the per-file protocol; under the global protocol the gap is larger and favors BERT-large (0.377 vs. 0.392 with gold), untested for significance and not claimed to be negligible. This is not a claim of beating the field (several published figures are higher under their own protocols); it is evidence that Table 4 cannot support the comparisons it is conventionally used for. Future state-of-the-art claims on this corpus should specify K-selection strategy and scoring granularity as routinely as a test set.

Early evidence suggested prompted LLMs lag supervised systems (Li et al., 2024a; Inoue et al., 2025); the 2026 results above reverse that picture under their own protocol. Whether frontier LLMs also lead under the corpus-level protocol of Table 2 is untested by either paper and is a natural next step. The practical case for a compact supervised model is orthogonal to this ranking question: a frozen encoder with a two-layer GCN runs locally at fixed cost, without per-message calls to a proprietary model.

6.3 Response Prioritization

Section 4.3 defines a response-prioritization score combining an unanswered-question indicator, GCN thread-assignment confidence, and a keyword-based urgency signal. It runs on the output of any disentanglement model here, but rigorously evaluating it requires human-annotated urgency ground truth that does not exist for this corpus. We therefore report it as a design contribution without an accuracy figure; a manually annotated evaluation set is left to future work (Section 7).

7 Future Works

Several directions remain open. First, Section 6 shows prompted LLMs (Takada and Mori, 2026) and our pipeline are evaluated under incompatible protocols; re-scoring published LLM predictions, or running a zero-shot LLM baseline ourselves, under the corpus-level oracle-K protocol of Table 2 is the direct next step. Second, the prioritization scoring function (Section 6.3) needs a human-annotated evaluation set with annotator agreement statistics. Further directions include real-time streaming adaptation (Bhukar et al., 2023), multilingual and cross-domain extension, and user studies measuring whether disentanglement and prioritization reduce operator workload, since intrinsic clustering metrics alone do not establish that.

8 Conclusion

We set out to test, rather than assume, what drives performance in PLM+GNN hybrids for conversation disentanglement. Holding one GCN architec-

References	Core Method/Model	Dataset	NMI	ARI	Other Metrics
Kummerfeld et al. (2019)	FFNN + greedy clustering	IRC Disentanglement Corpus (77k msgs)	–	–	VI, 1-1, F1 (no NMI/ARI reported)
Ma et al. (2022)‡	StructBERT (speaker + reference structure)	Ubuntu IRC corpus	0.678	0.371	Shen-F = 0.675
Huang et al. (2022)‡	Bi-CL (contrastive learning)	Ubuntu IRC corpus	0.624	0.360	Shen-F = 0.707
Li et al. (2024)‡	DiHRL (heterogeneous GNN + hierarchical ranking)	Ubuntu IRC corpus	–	–	+7.5 ARI over their own Struct baseline (0.736 → 0.811)
Lam & Yang (2025)§	DeBERTa link prediction + clustering	IRC Disentanglement Corpus (77k msgs)	–	0.664	Link F1 = 0.723
Takada & Mori (2026)†	Gemini-2.5-pro, zero-shot DLA+SC	IRC Disentanglement Corpus, test split	0.98	0.92	VI 0.97, 1-1 0.91; single run
This work (handcrafted)	**Position, length, author (gold edges)**	**Official test split (5k msgs, 961 threads)**	**–**	**0.401 ± 0.003**	**5-seed mean $\pm$ std (per-file K)**
This work (BERT-large)	GCN + BERT-large embeddings (gold edges)	Official test split (5k msgs, 961 threads)	–	0.400 ± 0.003	no significant difference ($p = 0.40$)

Table 4: Comparative survey of conversation disentanglement methods on IRC-style data, spanning 2006–2026. This table surveys the published record; it is not a leaderboard, since rows differ in splits, cluster-count strategy, and scoring granularity (Section 6). ‡ Ubuntu IRC subset with different preprocessing from our splits. § Link-based ARI from predicted reply links (10-run average). † DLA+SC: dialogue-level assignment with subsequent context; single run; scoring granularity not stated.

ture fixed and varying only node features or edge types, across five seeds and both clustering protocols, we find that informative node features are necessary (ARI 0.401 vs. 0.062 with constant features) but that which informative features are used makes no significant difference at this sample size ($n = 5$ seeds): cheap handcrafted features score within 0.002 ARI of BERT-large, and combining them adds nothing. Gold reply-to edges can be removed at no cost, and of the three edge types speaker continuity alone accounts for the entire effect.

The result that most constrains how any of this should be read is the non-learned one. Grouping messages by author, with no model, reaches ARI 0.4287, and nothing we trained beats it; the best graph configuration matches that partition's ARI to four decimal places, with zero seed variance. This follows from the corpus, not the models: most gold test threads are single messages by a single speaker, so under oracle-K clustering metrics authorship nearly determines the answer. We therefore do not read our numbers, or comparable published ones, as evidence about how well these architectures disentangle conversation, but as evidence about what this benchmark, scored this way, actually measures. Two errors in our own earlier reporting are corrected here: a handcrafted-over-BERT advantage that was an artifact of unstandardised scaling, and a table that mixed clustering protocols. This is precisely the class of error the paper argues is widespread.

Several limitations remain. All evaluations use the IRC corpus, so cross-domain generalisation is untested. The per-file node-classification objective is weak by construction (Section 4.2); a contrastive or pairwise objective would be better matched, and we did not test whether it changes the feature comparison. Our conclusions are specific to ARI and NMI under oracle K; one-to-one overlap, Shen-F and variation of information weight short threads differently and may not show the same pattern. Our handcrafted features include author identity, the same signal speaker-continuity edges encode; a matched ablation stripping it (Section 5.3) would separate the two more cleanly. Most importantly, the trivial-baseline result invites a constructive follow-up: evaluating on the subset of long, genuinely multi-speaker threads, where authorship is not a sufficient statistic, would say considerably more about model quality than full-split numbers.

The full reproducibility protocol, including every split, seed, and hyperparameter used, is specified in Section 4.4.

References

Karan Bhukar, Harshit Kumar, Dinesh Raghu, and Ajay Gupta. 2023. End-to-end deep reinforcement learning for conversation disentanglement. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 12571–12579.

Kent K 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.

Youngjae Chang and Youngjoong Ko. 2025. Soft engagement with pseudo initiatives for multi-party dialogue generation. *Pattern Recognition Letters*, 191:103–109.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186.

Micha Elsner and Eugene Charniak. 2008. You talking to me? a corpus and algorithm for conversation disentanglement. In *Proceedings of ACL-08: HLT*, pages 834–842.

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

Yaxin Fan, Feng Jiang, Peifeng Li, Fang Kong, and Qiaoming Zhu. 2023. Improving dialogue discourse parsing via reply-to structures of addressee recognition. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8484–8495.

Shilpa Gite, Urvi Rawat, Sameer Kumar, Bunny Saini, Anant Bhatt, Ketan Kotecha, and Nithesh Naik. 2024. Unfolding conversational artificial intelligence: A systematic review of datasets, techniques and challenges in developments. *Engineered Science*, 31:1210.

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.

Jia-Chen Gu, Chongyang Tao, and Zhen-Hua Ling. 2022. Who says what to whom: A survey of multi-party conversations. In *IJCAI*, pages 5486–5493.

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. Association for Computational Linguistics.

Koji Inoue, Divesh Lala, Mikey Elmers, Keiko Ochi, and Tatsuya Kawahara. 2025. An llm benchmark for addressee recognition in multi-modal multi-party dialogue. In *Proceedings of the 15th International Workshop on Spoken Dialogue Systems Technology (IWSDS)*, pages 330–334.

Kevin Joshua T, Arnav Agarwal, Shriya Sanjay, Yash Sarda, John Sahaya Rani Alex, Saurav Gupta, Sushant Kumar, and Vishwanath Kamath. 2024. Thread detection and response generation using transformers with prompt optimisation. In *2024 International Conference on Signal Processing and Communications (SPCOM)*, pages 1–5. IEEE.

Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*.

Shivani Kumar, Sumit Bhatia, Milan Aggarwal, and Tanmoy Chakraborty. 2023. Dialogue agents 101: a beginner's guide to critical ingredients for designing effective conversational systems. *Natural Language Processing*, pages 1–39.

Jonathan K Kummerfeld, Sai R Gouravajhala, Joseph J Peper, Vignesh Athreya, Chulaka Gunasekara, Jatin Ganhotra, Siva Sankalp Patel, Lazaros C Polymenakos, and Walter S 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. Association for Computational Linguistics.

Tung-Thien Lam and Cheng-Zen Yang. 2025. Revisiting pre-trained language models for conversation disentanglement. In *Proceedings of the 37th Conference on Computational Linguistics and Speech Processing (ROCLING 2025)*, pages 296–302. Association for Computational Linguistics.

Bobo Li, Hao Fei, Fei Li, Shengqiong Wu, Lizi Liao, Yinwei Wei, Tat-Seng Chua, and Donghong Ji. 2024a. Revisiting conversation discourse for dialogue disentanglement. *ACM Transactions on Information Systems*, 43(1):1–34.

Jingyang Li, Shengli Song, Yixin Li, Hanxiao Zhang, and Guangneng Hu. 2024b. Chatmdg: A discourse parsing graph fusion based approach for multi-party dialogue generation. *Information Fusion*, 110:102469.

Tianda Li, Jia-Chen Gu, Xiaodan Zhu, Quan Liu, Zhen-Hua Ling, Zhiming Su, and Si Wei. 2020. Dialbert: A hierarchical pre-trained model for conversation disentanglement. *arXiv preprint arXiv:2004.03760*.

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.

Ryan Lowe, Nissan Pow, Iulian Serban, and Joelle Pineau. 2015. The Ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. In *Proceedings of the 16th Annual*

Meeting of the Special Interest Group on Discourse and Dialogue, pages 285–294, Prague, Czech 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 265–278.

Navonil Majumder, Soujanya Poria, Devamanyu Hazarika, Rada Mihalcea, Alexander Gelbukh, and Erik Cambria. 2019. Dialoguernn: An attentive rnn for emotion detection in conversations. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 6818–6825.

Tiezheng Mao, Tianyong Hao, Jialing Fu, and Osamu Yoshie. 2024. Hierarchical graph fusion network and a new argumentative dataset for multiparty dialogue discourse parsing. *Information Processing & Management*, 61(2):103613.

Elijah Mayfield, David Adamson, and Carolyn Rose. 2012. Hierarchical conversation structure prediction in multi-party chat. In *Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 60–69.

Shikib Mehri and Giuseppe Carenini. 2017. Chat disentanglement: Identifying semantic reply relationships with random forests and recurrent neural networks. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 615–623.

Daniela F Milon-Flores and Robson LF Cordeiro. 2022. How to take advantage of behavioral features for the early detection of grooming in online conversations. *Knowledge-Based Systems*, 240:108017.

Izidor Mlakar, Darinka Verdonik, Simona Majhenič, and Matej Rojc. 2023. Understanding conversational interaction in multiparty conversations: the eva corpus. *Language Resources and Evaluation*, 57(2):641–671.

Kyungjin Park, Hyunwoo Sohn, Wookhee Min, Bradford Mott, Krista Glazewski, Cindy E Hmelo-Silver, and James Lester. 2022. Disruptive talk detection in multi-party dialogue within collaborative learning environments with a regularized user-aware network. In *Proceedings of the 23rd annual meeting of the special interest group on discourse and dialogue*, pages 490–499.

Xiuwei Shang, Shuai Zhang, Yitong Zhang, Shikai Guo, Yulong Li, Rong Chen, Hui Li, Xiaochen Li, and He Jiang. 2024. Analyzing and detecting information types of developer live chat threads. *ACM Transactions on Software Engineering and Methodology*, 33(5):1–32.

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*, pages 35–42.

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 2026), co-located with AAAI 2026*, volume 4178 of *CEUR Workshop Proceedings*, pages 14–37, Singapore.

Chao-Hong Tan, Jia-Chen Gu, and Zhen-Hua Ling. 2023. Is ChatGPT a good multi-party conversation solver? In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 4905–4915, Singapore. Association for Computational Linguistics.

Ming Tan, Dakuo Wang, Yupeng Gao, Haoyu Wang, Saloni Potdar, Xiaoxiao Guo, Shiyu Chang, and Mo Yu. 2019. Context-aware conversation thread detection in multi-party chat. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 6456–6461.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *International Conference on Learning Representations*.

Nikhil Yelamarthy and Oshin Anand. 2023. Context-aware information extraction from multi-thread business conversations. In *2023 5th International Conference on Natural Language Processing (ICNLP)*, pages 274–283. IEEE.

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. Association for Computational Linguistics.

Wentao Zhang, Shuang Xu, and Haoran Huang. 2022. Two-level supervised contrastive learning for response selection in multi-turn dialogue. *arXiv preprint arXiv:2203.00793*.