A Computational Approach to Modeling Conversational Systems: Analyzing Large-Scale Quasi-Patterned Dialogue Flows

Mohamed Achref Ben Ammar
achrafbenammar@callem.ai

Abstract

The analysis of conversational dynamics has become increasingly important with the rise of large language model-based conversational systems. As these systems interact with users across diverse contexts, understanding and representing the underlying patterns in conversations are critical for ensuring consistency, reliability, and dependability. In this work, we present a novel computational framework for constructing conversational graphs that effectively capture the flow and patterns within sets of conversations that do not follow strict conversational structures but nevertheless exhibit common underlying conversational flow patterns—referred to as quasi-patterned sets of conversations. Our approach combines advanced embedding techniques, clustering, and large language models to extract intents and transitions, leading to a clear, interpretable graph representation. Through comparative analysis of various graph simplification methods, we demonstrate that our Filter&Reconnect method produces the most readable and insightful graphs, allowing for the visualization of complex conversational flows with minimal noise. This work offers a scalable and robust solution for analyzing large-scale dialogue datasets, with practical implications for enhancing automated conversational systems.

1 Introduction

Several techniques for dialogue modeling have been explored, including intent extraction (Schuster et al., 2019) and dialogue act modeling (Khanpour et al., 2016). While dialogue act modeling classifies utterances into specific communicative functions, the intent extraction method seeks to identify the underlying intents expressed in utterances. Nevertheless, these techniques are designed to analyze conversations as elements by themselves rather than as part of a set of conversations that presents underlying conversational patterns.

In this work, we present a computational approach to constructing conversational graphs that can effectively represent the intents and transitions within quasi-patterned sets of dialogues. This high-level overview involves the following key steps: 1) Utterance embedding (Reimers and Gurevych, 2019), 2) Clustering (Arthur and Vassilvitskii, 2007), 3) Outlier removal, 4) Intent extraction, 5) Transition matrix construction, and 6) Conversational graph construction. These steps will be thoroughly explained in subsequent sections of the paper.

The structure of this paper is as follows: Section 2 reviews related work, while Section 3 details the materials and methods used, including the dataset chosen for this subject, the methodology and the evaluation process. Section 4 presents the results and discussion, and Section 5 concludes the study.

2 Related Work

This section reviews key areas relevant to our research, including intent extraction, dialogue act modeling, conversational graph representations, and sentence embedding models. These topics provide the foundational techniques and methodologies that underpin our approach to analyzing quasi-patterned conversational datasets and constructing conversational graphs.

2.1 Intent Extraction

The intent extraction technique aims to identify the intent of each utterance in a dialogue. This technique has seen a recent rise in applications with the evolution of deep learning and natural language processing (NLP) in the past few years. For example, Schuster et al. (2019)

presented a bidirectional encoder representations from transformers (BERT)-based model for intent detection and slot filling, that has achieved state-of-the-art performance on multiple datasets. Neural network models, such as those proposed by Goo et al. (2018), have also shown remarkable success in joint intent detection and slot filling.

Joint intent detection and slot filling is the process by which we can extract the intent of a user's utterance and the relevant data simultaneously. (e.g., the intent : booking a flight and the relevant information 'slots': date / destination).

2.2 Dialogue Act Modeling

Dialogue act modeling is an approach to categorizing an utterance from a dialogue into a specific communicative function, such as a question , a statement , a command or other. Some notable works include Khanpour et al. (2016), who introduced deep learning models for dialogue act classification, outperforming traditional machine learning approaches.

2.3 Conversational Graphs

A conversational graph is a way to mathematically represent a conversation's dynamics by representing intents as nodes and transitions as edges, as described by (Gritta et al., 2021). This graph-based representation, known as the Conversation Graph (ConvGraph), can be leveraged for data augmentation, multi-reference training, and evaluation of non-deterministic agents. ConvGraph is particularly useful in scenarios where traditional dialogue systems struggle due to limited training data or the inherent complexity of dialogue paths. By generating novel dialogue paths, ConvGraph enhances data volume and diversity, leading to more robust dialogue systems.

2.4 Sentence Embedding Models

Sentence embedding models, such as Sentence-BERT (SBERT) (Reimers and Gurevych, 2019), have significantly enhanced the capacity to capture and encode semantic information from textual data. SBERT, a modification of the pretrained BERT architecture, was specifically designed to address the computational challenges of deriving meaningful sentence representations for tasks requiring semantic similarity comparisons. By leveraging siamese and triplet network structures.

The introduction of SBERT marked a considerable improvement in both speed and accuracy over previous models such as BERT and RoBERTa for tasks like semantic textual similarity, clustering, and paraphrase identification.

By encoding sentences into a high-dimensional semantic space, SBERT allows for effective semantic analysis, enabling downstream NLP applications to function more robustly across various domains.

3 Materials and Methods

This section details the research methodology, starting with an overview of the ABCD dataset and its key statistics, followed by the techniques used to analyze intent flow patterns, and an evaluation of these techniques.

3.1 Dataset

The dataset chosen for this study is the ABCD (Action-Based Conversations Dataset) (Chen et al., 2021), which contains customer support conversations between agents and customers. This dataset is particularly well-suited for our research because customer support interactions often loosely follow guided paths. Customers generally have recurring types of inquiries, leading to a reacuring patterns in the agent's flow of responses. While these conversations do not follow a strict dialogue structure, the patterns observed in the agents' actions and responses across multiple conversations provide sufficient regularity to classify this dataset as a quasi-patterned set of conversations. This makes it an ideal dataset for constructing and analyzing conversational graphs that aim to capture the quasi-patterned nature of such sets of interactions.

In this context, several key terminologies are used throughout the dataset:

- **Agent**: Refers to the customer support agent handling the interaction.
- **Customer**: Refers to the customer making the inquiry or seeking support.
- **Action**: Represents a task or operation performed by the agent, such as accessing a database, searching through an FAQ page, or retrieving customer information.

These actions are an integral part of the dialogue and are often triggered by specific customer requests or queries.

To provide a quantitative overview of the dataset, the general statistics and interaction-specific metrics are summarized in Tables 1, 2, and 3.

General Dataset Statistics	Value
Average dialogue length (iterations)	22
Average dialogue length (characters)	904
Average dialogue length (words)	175.17
Maximum dialogue length (words)	632
Minimum dialogue length (words)	35
Median dialogue length (words)	166.00
Variance of dialogue length (words)	3683.11
Standard deviation of dialogue length (words)	60.69

Table 1: General Dataset Statistics

Agent Interactions	Value
Average interactions	9.47
Maximum interactions	31
Minimum interactions	1
Median interactions	9.00
Variance of interactions	10.86
Standard deviation of interactions	3.30

Table 2: Agent Interactions

Customer Interactions	Value
Average interactions	8.90
Maximum interactions	39
Minimum interactions	1
Median interactions	8.00
Variance of interactions	16.67
Standard deviation of interactions	4.08

Table 3: Customer Interactions

Action Interactions	Value
Average interactions	3.63
Maximum interactions	24
Minimum interactions	1
Median interactions	3.00
Variance of interactions	1.94
Standard deviation of interactions	1.39

Table 4: Action Interactions

3.2 Methodology

The proposed methodology for constructing conversational graphs involves several steps: Initially, we sample N_d dialogues from the ABCD dataset (Chen et al., 2021), denoted as $\mathcal{D} = \{d_1, d_2, \ldots, d_{N_d}\}$. Each utterance u_{ij} in dialogue d_i is embedded using the all-MiniLM-L12-v2 model (Reimers and Gurevych, 2019), resulting in a vector $v_{ij} = \text{Embed}(u_{ij})$.

These embeddings are then initially clustered into N_c clusters using the K-means++ algorithm (Arthur and Vassilvitskii, 2007), with the optimal number of clusters determined by the elbow method. The objective is to minimize the within-cluster sum of squares:

$$\arg\min_{k} \sum_{i=1}^{N_d} \sum_{j=1}^{|d_i|} ||v_{ij} - \mu_c||^2 \quad (1)$$

where μ_c is the centroid of cluster c.

After the initial clustering, outliers are identified and removed based on their Euclidean distances from the cluster centroids, defined by a threshold τ. Specifically, vectors with distances greater than the P_{75} percentile from their respective cluster centroids are considered outliers (see Appendix A for details). The remaining data is then reclustered using the K-means++ algorithm to produce more refined and spherical clusters.

For each cluster c, we then extract the k closest vectors to the cluster centroid:

$$\{r_{c_1}, r_{c_2}, \ldots, r_{c_k}\} = \arg\min_{v_{ij} \in c} ||v_{ij} - \mu_c|| \quad (2)$$

The utterances corresponding to these k closest vectors are then passed through large language models (LLMs) such as Gemma2 (Team et al., 2024) or LLaMA3 (Dubey et al., 2024) with a prompt designed to extract the common intent from these utterances. The extracted intent is then assigned as the label for that cluster. Subsequently, every utterance u_{ij} in the sampled dialogues is labeled with the identified intent of its cluster (I_c).

We then construct a transition matrix (M) where each element ($M[i][j]$) represents the probability of transitioning from intent (I_i) to (I_j). Let (T_{ij}) denote the number of transitions from intent (I_i) to intent (I_j) in the sampled dialogues. The transition matrix (M) is defined as follows:

$$M[i][j] = \frac{T_{ij}}{\sum_{k=1}^{N_c} T_{ik}}$$

Finally, the conversational graph (G) is built by merging all labeled conversational flows, where nodes represent intents, and edges represent transition probabilities derived from the transition matrix. Various filtering techniques, such as threshold filtering (see Appendix B.1), top-K filtering (see Appendix B.2), and the novel Filter and Reconnect method (methods are explained in the next sub-sections), are explored to enhance the graph's readability and accuracy.

3.2.1 Threshold Technique

The Threshold Technique is a filtering method used to simplify the constructed conversational graph by setting a minimum weight threshold for the edges. In this context, the "weight" refers to the transition probability between two intents in the graph.

The process begins by examining all possible edges between nodes (intents) and filtering out any edge with a weight below the specified threshold (τ). The key goal of this method is to reduce noise by retaining only significant transitions in the graph, thereby improving readability and interpretability.

The threshold (τ) is chosen based on the desired balance between noise reduction and coverage of conversational paths. Setting a high threshold results in a sparse graph with fewer edges, leading to a loss in coverage but enhanced clarity. On the other hand, a low threshold increases coverage but may introduce noise by retaining less significant transitions.

Formally, the filtering condition is:

$$\text{Keep edge } (I_i \rightarrow I_j) \text{ if } M[i][j] \geq \tau$$

where $M[i][j]$ is the transition probability from intent I_i to intent I_j.

3.2.2 Top-K Filtering Technique

The Top-K Filtering Technique is another approach used to simplify the constructed conversational graph. It combines filtering based on a minimum weight threshold with a selection process that retains only the top K highest-weighted edges for each node.

The method involves two steps: Threshold Filtering First, a minimum weight threshold (τ) is applied to filter out edges with weights below a certain value, similar to the Threshold Technique described above. Top-K Selection: For each node in the graph, only the top K edges with the highest weights are retained, based on the transition probabilities. This process ensures that each node preserves only the most likely transitions while discarding less probable ones.

This technique provides a focused representation of the conversational graph by emphasizing the most common conversational paths, while still maintaining a degree of complexity by allowing multiple transitions per node. The parameter K controls the level of simplification, where a low value of K (e.g., $K = 1$) results in a highly simplified and readable graph, whereas a higher K value allows for more comprehensive coverage of the dialogue paths.

3.2.3 Filter&Reconnect Method

The novel Filter&Reconnect method (Algorithm 1) constructs and refines a directed graph based on the transition matrix. This method ensures that the graph is acyclic, providing a clear and readable representation of the conversational flow.

This algorithm begins by filtering edges based on a minimum weight threshold (τ) and excluding self-transitions. After filtering, edges between intents are added according to the transition probabilities, but only the top-k incoming edges for each node are retained. The algorithm then applies cycle elimination and reconnects any detached subgraphs, producing a clean and readable graph.

The "Remove Cycles" algorithm (Algorithm 2) is a crucial step that ensures the graph remains acyclic. It identifies cycles and systematically removes the weakest (lowest weight) edges within them until no cycles are left. This process is essential for maintaining the logical flow of conversations in the graph, avoiding confusing loops that could obscure the true sequence of intents. The "Reconnect Subgraphs" algorithm (Algorithm 3) ensures that any subgraphs created by removing cycles or filtering edges are reconnected to each other to form the final conversational graph.

Algorithm 1 Filter&Reconnect

Input: Transition matrix M, intent by cluster C, top-k k, min_weight threshold τ
Output: Directed graph G
$G \leftarrow$ Initialize directed graph
for each intent i in C **do**
  for each intent j in C **do**
    if $i \neq j$ and $M[i][j] > \tau$ **then** ▷ Exclude self-transitions and filter by min_weight
      Add edge from $C[i]$ to $C[j]$ with weight $M[i][j]$ to G
    end if
  end for
end for
for each node v in G **do**
  incoming_edges $\leftarrow$ Get incoming edges for v
  Sort incoming_edges by weight in descending order
  Keep top k edges
  Remove all other incoming edges
end for
Remove cycles by iteratively removing the weakest edge in each cycle (see Algorithm 2)
Reconnect detached subgraphs using the transition matrix (see Algorithm 3)
Return G

By finding the strongest transition between nodes in smaller subgraphs and the main subgraph, the algorithm re-establishes connectivity while preserving the most probable transitions. This process guarantees that the final graph represents a unified conversational flow.

3.2.4 Coverage Metric

The Coverage Metric is designed to measure how effectively the conversational graph, represented as a directed graph $G = (V, E)$, is able to capture the actual conversations in the dataset. The graph G models the intents (represented by the set of vertices V) and the probabilities of transitions between these intents (represented by the edges E). The key idea behind the Coverage Metric is to assess how well the graph G can "cover" or represent the real-world conversation flows in the dataset F, where each flow $f \in F$ is a sequence of intents $f = [i_1, i_2, \ldots, i_n]$ from one conversation from the quasi-patterned set of conversations. The Transition Alignment Score for a given conversation flow f is defined as the ratio of the number of transitions in f that are present in the graph G to the total number of transitions in f. This captures how much of the flow can be "aligned" or represented by the edges in the graph. Formally, the Transition Alignment Score for a flow f is calculated as:

$$TAS(f) = \frac{\sum_{j=1}^{n-1} \mathbf{1}(i_j, i_{j+1}) \in E}{n-1} \tag{3}$$

where $\mathbf{1}(i_j, i_{j+1}) \in E$ is an indicator function that is 1 if the edge (i_j, i_{j+1}) exists in the graph G, and 0 otherwise. The Coverage Metric C is then defined as the average Transition Alignment Score across all the conversation flows in the dataset F:

$$C = \frac{1}{|\mathcal{F}|} \sum_{f \in \mathcal{F}} TAS(f) \tag{4}$$

This Coverage Metric provides a quantitative way to evaluate how well the conversational graph G is able to capture and represent the actual dialogue flows in the dataset. A higher Coverage Metric indicates that the graph can more accurately model the real-world conversations.

4 Results and Discussions

This section provides observations based on the different graph simplification techniques visualized through generated graphs and coverage metrics. The figures showcasing the visualizations for the filtering techniques (Threshold Filtering, Top-K Filtering, and Filter and Reconnect) are presented in the appendix (Figures 3, 4, and 5).

4.1 Threshold Filtering

The conversational graph generated with a minimum weight threshold of $\tau = 0.1$,Figures 3, illustrates the limitations of basic filtering methods. While threshold filtering can reduce some noise, the resulting graph remains overly complex, with numerous low-significance edges and a lack of discernible structure. This makes it challenging to derive meaningful insights, especially when analyzing large sets of conversations. Consequently, this method is not well-suited for revealing the underlying patterns within quasi-patterned conversation sets.

4.2 Top-K Filtering

The conversational graph generated with top-K=1,Figure 4, retains only the most likely transition for each node, providing a highly simplified view of conversational dynamics. While this simplification improves clarity, it may result in an underrepresentation of diverse conversational paths, reducing the coverage of actual dialogues represented by the graph.

4.3 Filter&Reconnect

The novel Filter&Reconnect method proves to be the most effective approach for extracting conversational patterns within quasi-patterned sets of conversations. Unlike more basic filtering techniques, the graph generated by this method is highly readable, with minimal to no noisy edges. By systematically filtering and reconnecting the edges, the method eliminates unnecessary transitions, resulting in a clean, structured graph.

One of the key strengths of the Filter&Reconnect method is the clear and visible branching structure it produces. This structure effectively captures the typical flow of conversations, particularly in the ABCD dataset, where most dialogues begin with a common intent (e.g., agent : "offer assistance to the customer") and then recursively branch out into distinct paths as the conversation progresses. This branching pattern highlights the varying paths customers and agents might take based on the context and nature of the interaction.

Overall, the Filter&Reconnect method not only preserves essential transitions but also enhances the clarity of the graph by focusing on significant paths, making it the most suitable approach for analyzing and visualizing the complex conversational dynamics present in quasi-patterned conversation sets.

4.4 Coverage vs. Minimum Weight

The coverage metric as a function of the minimum weight threshold helps to understand the trade-offs between readability and coverage efficiency (Figure 1).

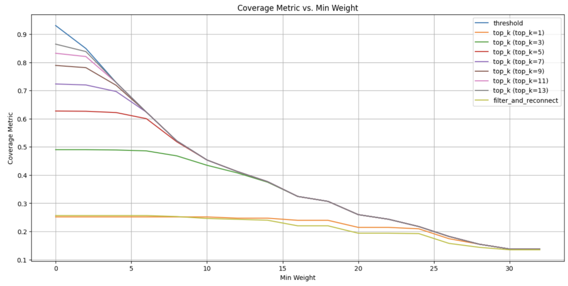

Figure 1: Coverage metric versus minimum weight threshold for different graph simplification techniques: Top-K filtering, Filter and Reconnect, and Threshold filtering. The graph shows how varying the minimum weight affects the coverage across these methods.

As the threshold increases, coverage typically decreases for all techniques. While Filter and Reconnect provides more balanced coverage across a range of thresholds, Top-K filtering offers higher coverage for low thresholds for higher k values , and a quasi-identical coverage for k=1. Threshold filtering shows a sharp decline in coverage as the minimum weight threshold increases, highlighting the trade-off between reducing noise and maintaining conversational flow representation.

5 Conclusion

This research presents a novel and effective methodology for extracting and constructing conversational graphs from customer service interactions within quasi-patterned sets of conversations. By leveraging the ABCD dataset, we captured and analyzed real-world customer support interactions, focusing on identifying underlying patterns in agent responses. The methodology involved embedding utterances using sentence-transformers, clustering these embeddings, removing outliers and reclustering the data to refine the clustering quality. Utterances corresponding to the representative vectors of each cluster were then processed using large language models (LLMs) to extract common intents, which were used to label clusters and construct a transition matrix for generating the final conversational graph.

Among the techniques explored, the novel Filter&Reconnect method proved to be the most effective for graph simplification. This method not only minimized noise but also preserved the essential branching structures inherent in quasi-patterned conversation sets. The resulting graphs were acyclic, and offered a clear representation of conversational flows, making them highly interpretable and insightful for understanding this set of dialogues.

The proposed computational approach is adaptable to any quasi-patterned set of conversations. These insights could be used to monitor conversational (AI) systems, improve the training of customer service models, leading to more reliable conversational systems.

In summary, this research introduces a robust approach to constructing conversational graphs that reveal the hidden structures within quasi-patterned conversations. Future work could explore extending this methodology to different domains or enhancing the algorithm to accommodate more diverse conversational data.

Acknowledgments

We would like to thank the contributors and maintainers of the ABCD dataset for their invaluable resource. Additionally, we extend our gratitude to our colleagues and peers who provided insightful feedback throughout the course of this research.

References

David Arthur and Sergei Vassilvitskii. 2007. k-means++: the advantages of careful seeding. In *SODA '07: Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035, Philadelphia, PA, USA. Society for Industrial and Applied Mathematics.

Derek Chen, Howard Chen, Yi Yang, Alexander Lin, and Zhou Yu. 2021. Action-based conversations dataset: A corpus for building more in-depth task-oriented dialogue systems. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3002–3017, Online. Association for Computational Linguistics.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaoqing Ellen Tan, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen

Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aaron Grattafiori, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alex Vaughan, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Franco, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, Danny Wyatt, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Firat Ozgenel, Francesco Caggioni, Francisco Guzmán, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Govind Thattai, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Karthik Prasad, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kun Huang, Kunal Chawla, Kushal Lakhotia, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Maria Tsimpoukelli, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikolay Pavlovich Laptev, Ning Dong, Ning Zhang, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Rohan Maheswari, Russ Howes, Ruty Rinott, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Kohler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vítor Albiero, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaofang Wang, Xiaojian Wu, Xiaolan Wang, Xide Xia, Xilun Wu, Xinbo Gao, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yuchen Hao, Yundi Qian, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, and Zhiwei Zhao. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Chih-Wen Goo, Guang Gao, Yun-Kai Hsu, Chih-Li Huo, Tsung-Chieh Chen, Keng-Wei Hsu, and Yun-Nung Chen. 2018. Slot-gated modeling for joint slot filling and intent prediction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 753–757, New Orleans, Louisiana. Association for Computational Linguistics.

Milan Gritta, Gerasimos Lampouras, and Ignacio Iacobacci. 2021. Conversation Graph: Data Augmentation, Training, and Evaluation for Non-Deterministic Dialogue Management. *Transactions of the Association for Computational Linguistics*, 9:36–52.

Hamed Khanpour, Nishitha Guntakandla, and Rodney Nielsen. 2016. Dialogue act classification in domain-independent conversations using a deep recurrent neural network. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2012–2021, Osaka, Japan. The COLING 2016 Organizing Committee.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *CoRR*, abs/1908.10084.

Sebastian Schuster, Sonal Gupta, Rushin Shah, and Mike Lewis. 2019. Cross-lingual transfer learning for multilingual task oriented dialog. 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 3795–3805, Minneapolis, Minnesota. Association for Computational Linguistics.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, Nino Vieillard, Piotr Stanczyk, Sertan Girgin, Nikola Momchev, Matt Hoffman, Shantanu Thakoor, Jean-Bastien Grill, Behnam Neyshabur, Olivier Bachem, Alanna Walton, Aliaksei Severyn, Alicia Parrish, Aliya Ahmad, Allen Hutchison, Alvin Abdagic, Amanda Carl, Amy Shen, Andy Brock, Andy Coenen, Anthony Laforge, Antonia Paterson, Ben Bastian, Bilal Piot, Bo Wu, Brandon Royal, Charlie Chen, Chintu Kumar, Chris Perry, Chris Welty, Christopher A. Choquette-Choo, Danila Sinopalnikov, David Weinberger, Dimple Vijaykumar, Dominika Rogozińska, Dustin Herbison, Elisa Bandy, Emma Wang, Eric Noland, Erica Moreira, Evan Senter, Evgenii Eltyshev, Francesco Visin, Gabriel Rasskin, Gary Wei, Glenn Cameron, Gus Martins, Hadi Hashemi, Hanna Klimczak-Plucińska, Harleen Batra, Harsh Dhand, Ivan Nardini, Jacinda Mein, Jack Zhou, James Svensson, Jeff Stanway, Jetha Chan, Jin Peng Zhou, Joana Carrasqueira, Joana Iljazi, Jocelyn Becker, Joe Fernandez, Joost van Amersfoort, Josh Gordon, Josh Lipschultz, Josh Newlan, Ju yeong Ji, Kareem Mohamed, Kartikeya Badola, Kat Black, Katie Millican, Keelin McDonell, Kelvin Nguyen, Kiranbir Sodhia, Kish Greene, Lars Lowe Sjoesund, Lauren Usui, Laurent Sifre, Lena Heuermann, Leticia Lago, Lilly McNealus, Livio Baldini Soares, Logan Kilpatrick, Lucas Dixon, Luciano Martins, Machel Reid, Manvinder Singh, Mark Iverson, Martin Görner, Mat Velloso, Mateo Wirth, Matt Davidow, Matt Miller, Matthew Rahtz, Matthew Watson, Meg Risdal, Mehran Kazemi, Michael Moynihan, Ming Zhang, Minsuk Kahng, Minwoo Park, Mofi Rahman, Mohit Khatwani, Natalie Dao, Nenshad Bardoliwalla, Nesh Devanathan, Neta Dumai, Nilay Chauhan, Oscar Wahltinez, Pankil Botarda, Parker Barnes, Paul Barham, Paul Michel, Pengchong Jin, Petko Georgiev, Phil Culliton, Pradeep Kuppala, Ramona Comanescu, Ramona Merhej, Reena Jana, Reza Ardeshir Rokni, Rishabh Agarwal, Ryan Mullins, Samaneh Saadat, Sara Mc Carthy, Sarah Perrin, Sébastien M. R. Arnold, Sebastian Krause, Shengyang Dai, Shruti Garg, Shruti Sheth, Sue Ronstrom, Susan Chan, Timothy Jordan, Ting Yu, Tom Eccles, Tom Hennigan, Tomas Kocisky, Tulsee Doshi, Vihan Jain, Vikas Yadav, Vilobh Meshram, Vishal Dharmadhikari, Warren Barkley, Wei Wei, Wenming Ye, Woohyun Han, Woosuk Kwon, Xiang Xu, Zhe Shen, Zhitao Gong, Zichuan Wei, Victor Cotruta, Phoebe Kirk, Anand Rao, Minh Giang, Ludovic Peran, Tris Warkentin, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, D. Sculley, Jeanine Banks, Anca Dragan, Slav Petrov, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Sebastian Borgeaud, Noah Fiedel, Armand Joulin, Kathleen Kenealy, Robert Dadashi, and Alek Andreev. 2024. Gemma 2: Improving open language models at a practical size. *Preprint*, arXiv:2408.00118.

A Appendix ; Outlier Removal Effect

This appendix demonstrates the impact of outlier removal on clustering. Figure 2 presents a t-SNE visualization comparing clustering results before and after outlier removal and reclustering. In the right plot, vectors with distances greater than the P_{75} percentile from their respective cluster centroids were removed, resulting in tighter and more coherent clusters.

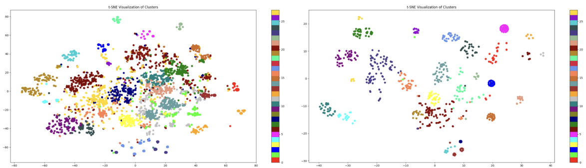

Figure 2: Comparison of clustering results before (left) and after (right) outlier removal and reclustering. Removing outliers enhances cluster coherence and improves the quality of the resulting conversational graphs.

B Appendix : Graph Visualization for Filtering Techniques

This appendix presents the graphical results of the different graph simplification techniques discussed in the Results and Discussions section.

B.1 Threshold Filtering

Threshold filtering applies a minimum weight threshold to edges in the graph. Any edge with a weight below the threshold is removed, ensuring that only significant transitions remain in the final graph. The primary benefit is a reduction in noise, leading to a clearer and more focused representation of the conversational flow. However, increasing the threshold too much can lead to a sharp decline in coverage, as fewer edges remain. The figure below illustrates this technique.

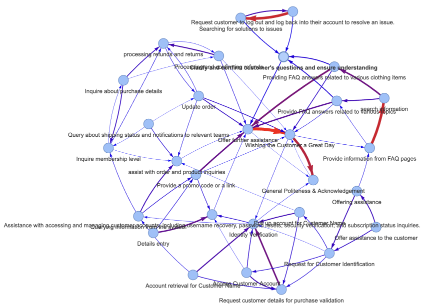

Figure 3: Conversational graph generated using a minimum weight threshold $\tau = 0.1$. This approach prioritizes readability by filtering out less significant transitions, though this comes at the cost of reduced coverage. As the threshold increases, the graph becomes sparser, focusing only on the most prominent transitions.

B.2 Top-K Filtering

Top-K filtering involves retaining only the top K transitions (based on weight) for each node in the graph. This technique ensures that only the most likely transitions are represented, simplifying the graph while preserving key conversational flows. While this method can improve readability, it may underrepresent diverse conversational paths, leading to a less comprehensive coverage. The figure below shows the effect of applying top-K filtering with $K = 1, \tau = 0.1$.

Figure 4: Conversational graph generated using a top-K filtering technique with $K = 1$. This technique prioritizes clarity by retaining only the most likely transitions for each node. However, this simplification might reduce the coverage of actual dialogue paths, resulting in a concise but potentially oversimplified graph.

B.3 Filter and Reconnect

The Filter and Reconnect technique balances retaining significant transitions and ensuring the graph remains acyclic . Although the coverage metric is generally lower than the threshold and top-K techniques, the resulting graphs are more readable with a clearer branching structure , representing how conversations start with the same intent ('offer assistance to the customer' for the ABCD dataset) and then recursively split into different paths.

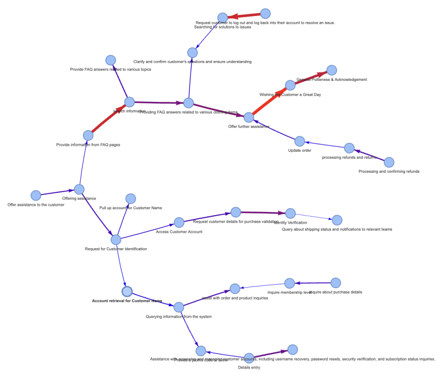

Figure 5: Conversational graph generated using the Filter and Reconnect method with $\tau = 0.1$. This approach ensures the graph is acyclic while also retaining key transitions. The graph exhibits a clear branching structure with fewer cycles, making it more readable and easier to trace the conversational flow while maintaining significant coverage.

C Appendix : Algorithmic Details for Filter and Reconnect Method

This appendix details the algorithms used for constructing and refining the conversational graph using the Filter and Reconnect method.

C.1 Filter&Reconnect Algorithm

The Filter&Reconnect algorithm constructs a directed graph based on the transition matrix and intent clusters, ensuring the graph is acyclic. The process starts by filtering edges based on a minimum weight threshold and excludes self-transitions. After filtering, the top-k incoming edges for each node are retained, cycles are removed, and any detached subgraphs are reconnected.

C.2 Cycle Removal Algorithm

The Cycle Removal algorithm ensures that the graph remains acyclic by identifying cycles and systematically removing the weakest edges within them until no cycles remain.

Algorithm 2 Remove Cycles

Input: Directed graph G
Output: Acyclic directed graph G
cycles $\leftarrow$ Find all cycles in G
while cycles $\neq \emptyset$ **do**
  for each cycle c in cycles **do**
    weakest_edge $\leftarrow$ Find the edge with the minimum weight in c
    Remove weakest_edge from G
  end for
  cycles $\leftarrow$ Find all cycles in G
end while
Return G

C.3 Subgraph Reconnection Algorithm

The Subgraph Reconnection algorithm reconnects any subgraphs created after cycle removal or edge filtering. It finds the strongest edge that can connect a node in a smaller subgraph to the largest main subgraph, thereby merging the disconnected subgraphs back into the main graph.

Algorithm 3 Reconnect Subgraphs

Input: Directed graph G, transition matrix M, intent by cluster C
Output: directed graph G
subgraphs $\leftarrow$ Find all weakly connected components in G
main_subgraph $\leftarrow$ Largest component in subgraphs
for each subgraph s in subgraphs **do**
  if $s \neq$ main_subgraph **then**
    max_weight $\leftarrow -\infty$
    best_edge $\leftarrow \emptyset$
    for each node u in s **do**
      for each node v in main_subgraph **do**
        weight $\leftarrow M[C^{-1}[u]][C^{-1}[v]]$
        if weight $>$ max_weight **then**
          max_weight $\leftarrow$ weight
          best_edge $\leftarrow (u, v)$
        end if
      end for
    end for
    if best_edge $\neq \emptyset$ **then**
      Add best_edge to G with weight max_weight
    end if
  end if
end for
Return G