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

Mohamed Achref Ben Ammar
Callem AI
Tunis, Tunisia
achraf.benammar@ieee.org

***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 solution for analyzing large-scale dialogue datasets, with practical implications for enhancing automated conversational systems.**

***Index Terms*—conversational systems, large-scale dialogue, conversational graphs, quasi-patterned dialogue, graph analysis**

I. Introduction

Several techniques for dialogue modeling have been explored, including intent extraction [5] and dialogue act modeling [7]. While dialogue act modeling classifies utterances into specific communicative functions, intent extraction 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 present underlying conversational patterns.

In this work, we present a computational approach to constructing conversational graphs, which are structured representations of dialogue flows. In a conversational graph, each node corresponds to an intent or topic identified within the conversation, and directed edges represent transitions between these intents. The weight of each edge denotes the probability or frequency of that transition occurring in the dataset.

Our approach focuses on quasi-patterned sets of conversations, defined as a collection of dialogues that, while not strictly following a predetermined or rigid structure, exhibit recurring patterns in their flow. These conversations may vary in specific details but generally share common intents, transitions, or topics. In this work, we leverage the quasi-patterned nature of customer support dialogues, where agent responses tend to follow certain themes or paths, even though the conversations themselves are not strictly scripted or formulaic.

This high-level overview involves the following key steps in order: Utterance embedding, Clustering, Intent extraction, Transition matrix construction, and Conversational graph construction. These steps will be thoroughly explained in subsequent sections of the paper.

The structure of this paper is as follows: Section II reviews related work, while Section III details the materials and methods used, including the dataset, methodology, and evaluation process. Section IV presents the results and discussion, and Section V concludes the study.

II. Related Work

The analysis of conversational systems has been a growing area of research, particularly with the proliferation of natural language processing (NLP) techniques and the development of large language models (LLMs). Various approaches have been proposed to model dialogue flows and extract meaningful patterns from conversational data. This section provides an overview of the key works related to intent extraction, dialogue act modeling and applications in customer support and automated systems highlighting the gap our framework seeks to address.

A. Intent Extraction and Dialogue Act Modeling

Intent extraction and dialogue act modeling form the foundation for understanding conversational dynamics. Intent extraction aims to identify the underlying intention or goal behind user utterances. Techniques for intent extraction have evolved from rule-based approaches to machine learning-based models, which leverage semantic embeddings and contextual features to improve accuracy [5]. Pre-trained language models, such as BERT [15], have further enhanced intent extraction capabilities by enabling models to capture rich contextual information. However, these models often focus on individual utterances without considering their place in the broader dialogue flow.

Dialogue act modeling, on the other hand, seeks to classify utterances into communicative functions such as questions,

requests, or confirmations. This line of research has produced notable results in labeling conversational data using supervised learning techniques [7], often focusing on specific domains such as customer service or task-oriented dialogue systems. While dialogue act models capture the functional role of each utterance, they tend to operate on individual conversation segments rather than holistically across sets of conversations.

Both intent extraction and dialogue act modeling are vital to our work, as they enable the identification of key components within dialogues. However, they fall short in addressing the higher-level patterns found in large-scale, quasi-patterned conversations. Our framework builds on these foundational approaches but moves beyond isolated utterances to explore the relationships and transitions between intents across conversations.

B. Applications in Customer Support and Automated Systems

The practical implications of conversational graph modeling are particularly relevant for customer support systems. Prior research has demonstrated the effectiveness of NLP techniques in automating responses and improving user experience in customer service settings [17]. Many existing systems rely on scripted dialogue flows or predefined response templates, which can limit flexibility and adaptability. By analyzing large-scale, quasi-patterned customer support conversations, our framework provides a means of improving conversational agents by identifying common paths, detecting bottlenecks, and suggesting optimizations based on real conversational data.

Our work contributes to this area by offering a solution for analyzing unstructured, real-world dialogues, thus enabling more adaptive and responsive automated systems.

III. Dataset

The dataset chosen for this study is the ABCD (Action-Based Conversations Dataset) [2], 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 recurring 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.

The distribution of utterance lengths in the dataset is shown in Figure 1. As seen from the histogram, most utterances are relatively short, containing fewer than 15 words, with a smaller number extending to 30 or more words. This distribution aligns with typical customer support interactions, where shorter, more direct exchanges are common.

Furthermore, the utterance length distribution by role (Figure 2) highlights that agents tend to use shorter utterances, with most containing fewer than 5 words. Customers, on the other hand, tend to have slightly longer utterances, which more often exceed 10 words. This distinction between agent and customer utterances reflects the typical nature of customer service dialogues, where agents provide concise responses and customers may explain their issues in more detail.

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 metrics are summarized in Table I.

IV. Methodology

In this section, we present the methodology of our approach. Figure 3 presents a general overview of our pipeline. The process begins by extracting utterances from the dataset, then each utterance is embedded by a text-embedding model [9], thus transforming them into vectors that represent the semantic meaning of each utterance. Then, these vectors get clustered to identify the key utterance intents contained in the quasi-patterned set of conversations using a Large Language Model (LLM).After that, we construct a Markov Chain, which will be filtered and processed to discard the noise from the graph, irrelevant intent transitions, and thus create a conversational graph that can be analyzed to extract the general paths that conversations follow in our quasi-patterned set of conversations.

Fig. 1: Distribution of utterance lengths (in words) in the dataset.

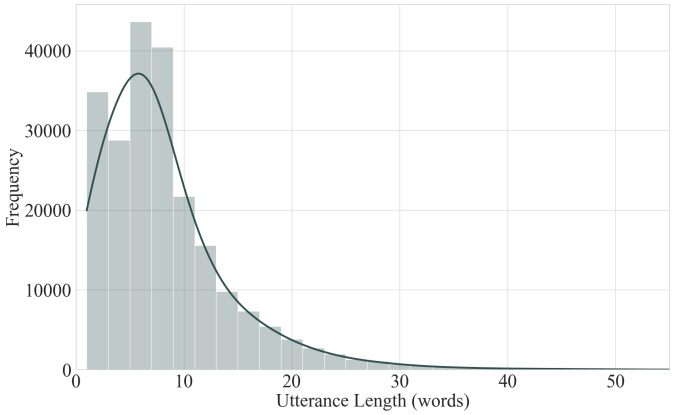


Fig. 2: Distribution of utterance lengths (in words) by role.

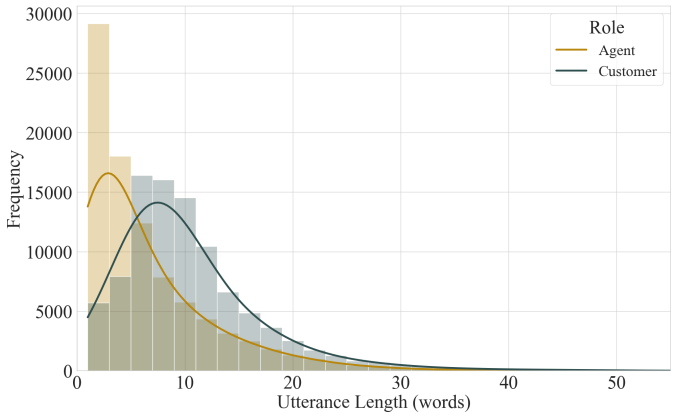


TABLE I: General Dataset Statistics

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

A. Vector Embedding Generation

The first step involves generating vector embeddings for each utterance using a pre-trained text embedding model. For this work, we utilized the all-MiniLM-L12-v2 model [9], which converts textual data into high-dimensional vectors that represent the semantic meaning of the utterances. These embeddings form the foundation for clustering, as they encapsulate both syntactic and semantic similarities [12]. By using embeddings that capture the contextual meaning of utterances, we ensure that the clusters formed represent coherent thematic groups.

B. Clustering and Intent Extraction

Once the vector embeddings of the utterances are generated, we use K-means++ [10] for clustering. K-means++ improves upon traditional K-means by strategically initializing centroids to be well-spread, leading to faster convergence and more accurate clusters. This reduces the likelihood of poor clustering due to random initialization, while maintaining the simplicity and scalability of K-means. The algorithm is efficient for large datasets, making it ideal for grouping utterances into meaningful thematic categories with minimal computational cost.

After clustering, we proceed with intent extraction from the clusters. To achieve this, we select the top_k vectors, with $k = 15$, closest to the centroid of each cluster and extract the common intent from the corresponding utterances. For intent extraction, we leverage the gemini-1.5-flash model [11] to summarize the core intent of each cluster, enabling us to label the groups with meaningful thematic descriptions.

C. Markov Chain Construction

Following clustering and intent extraction, we build a Markov chain [14] to model the transition probabilities between clusters based on the conversations' flow. A Markov chain is a stochastic model that describes transitions from one state to another according to defined probabilities. In this context, the states correspond to the different clusters, and the transitions between them represent the flow of intents throughout the conversations.

Let $S = \{s_1, s_2, \ldots, s_n\}$ denote the sequence of cluster IDs assigned to the n utterances in the order they appear in the conversations. We construct a transition matrix $T \in \mathbb{R}^{k \times k}$, where k is the number of unique clusters. The matrix element $T_{i,j}$ is defined as:

$$T_{i,j} = \frac{\text{count}(s_t = i, s_{t+1} = j)}{\sum_{j=1}^{k} \text{count}(s_t = i, s_{t+1} = j)}$$

Here, $\text{count}(s_t = i, s_{t+1} = j)$ represents the number of times a transition occurs from cluster i to cluster j in the sequence S. Each row in the transition matrix T sums to 1, indicating a probability distribution of transitions from one cluster to the next. Therefore, $T_{i,j}$ expresses the probability of moving from cluster i to cluster j.

The transition matrix T encapsulates the probabilistic flow of intents between clusters, representing the natural progression of themes and topics in the dialogues. This Markov chain model effectively maps how conversational elements evolve, providing insights into the underlying structure of the conversations and it will be considered as our base conversational graph for the next steps.

D. Conversational Graph Processing

After constructing the Markov chain, we refine the conversational graph using three different processing techniques to best fit the analysis of quasi-patterned conversations. Each technique simplifies the graph in different ways, aiming to improve readability and interpretability while maintaining meaningful conversational paths. These techniques are: Threshold Filtering, Top-K Filtering, and Filter&Reconnect.

Each of these techniques will be evaluated in the Results and Discussion section to determine which produces the most readable and analyzable conversational graph for our purposes.

1) Threshold Technique: The Threshold Technique simplifies the graph by removing edges with weights below a certain threshold (τ). The weight represents the transition probability between two intents. This technique attempts to reduce noise and retain only significant transitions. A higher threshold results in a sparser graph, while a lower threshold retains more transitions but may introduce noise.

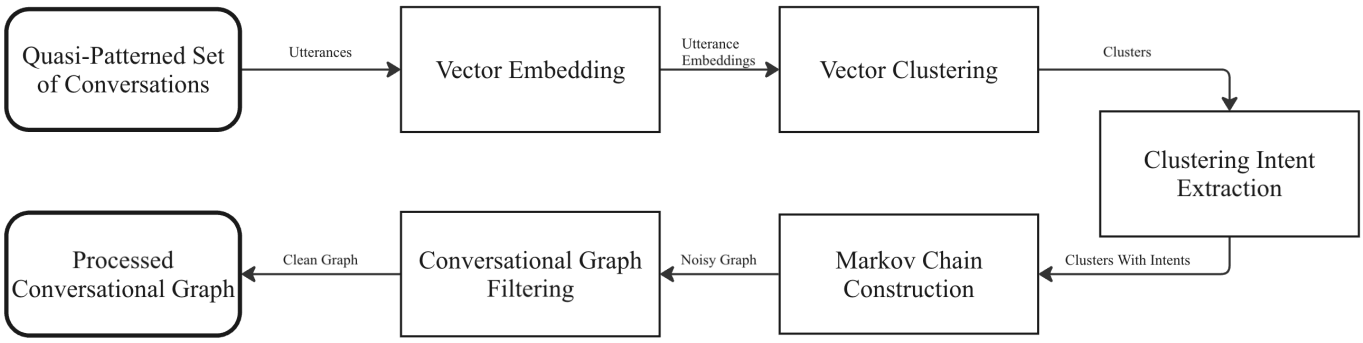


Fig. 3: Flowchart showing the process of conversation clustering and graph construction. Conversations are embedded into vectors, clustered, and used to build a conversational graph. Various filtering and processing techniques refine the graph, resulting in a processed conversational graph for evaluation.

2) Top-K Filtering Technique: The Top-K Filtering Technique retains the top K highest-weighted edges for each node while also applying a minimum weight threshold (τ) to remove insignificant edges. This technique involves two steps. First comes threshold filtering, edges below the weight threshold τ are removed and then Top-K Selection, For each node, only the top K edges with the highest weights are kept. This ensures that the most likely transitions are retained, providing a focused representation of the conversational graph.

3) Filter&Reconnect Method: The Filter&Reconnect method constructs an acyclic conversational graph by filtering and simplifying transitions. First, edges with weights below a threshold (τ) and self-transitions are removed. For each node, only the top-k strongest incoming edges are retained to emphasize significant transitions. Next, any cycles in the graph are identified, and the weakest edges within them are removed to ensure the graph remains acyclic. Finally, any subgraphs that were disconnected during this process are reconnected to the main graph—the subgraph with the most number of nodes, by restoring the strongest transition between them.

E. Evaluation

We evaluate our approach by determining the readability and interpretability of the final graphs produced by our approach. For readability, we evaluate it by determining the number of cycles in that graph; the more cycles a conversational graph has, the less readable it becomes and the harder it is to understand our quasi-patterned set of conversations. And for interpretability, we evaluate the graph by determining how much that graph looks like a tree. That is because quasi-patterned sets of conversations generally follow a tree structure, with branches being determined by the different paths conversations can take as they unfold. For this, we calculate:

- **Branching Factor**: The average number of edges per vertex in the graph.
- **δ-hyperbolicity**: Measures how close a graph is to being Gromov-hyperbolic, meaning it quantifies how much the graph's metric resembles a hyperbolic space [13] by analyzing the slimness of triangles formed by shortest paths in the graph, which correlates with tree-like structures because trees are the simplest examples of hyperbolic spaces. A graph with low delta-hyperbolicity has properties similar to a tree, where the distance between nodes can often be efficiently described by paths that are close to each other, much like the branching nature of trees.

V. Results and Discussions

In this section, we present the outcomes of applying three distinct graph simplification techniques—Threshold Filtering, Top-K Filtering, and Filter&Reconnect—on the conversational dataset. Each method is evaluated based on key performance metrics: delta-hyperbolicity, branching factor, and the number of cycles in the resulting graphs. We also provide visual representations of the generated conversational graphs to facilitate comparative analysis and insight.

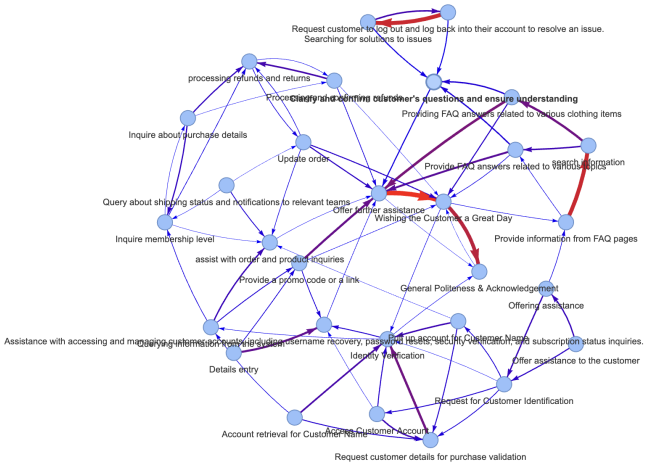
A. Threshold Filtering Results

The Threshold Filtering method, while effective at removing low-probability transitions, does not perform well in generating interpretable conversational graphs. The primary issue is the significant presence of noise, which results in a high number of cycles (56 cycles). As shown in Figure 4, the threshold filtering approach performs poorly in terms of the delta-hyperbolicity metric (2.50), indicating that the resulting graph deviates substantially from a tree-like structure. This renders the graph less readable and interpretable for understanding the conversational flow.

B. Top-K Filtering Results

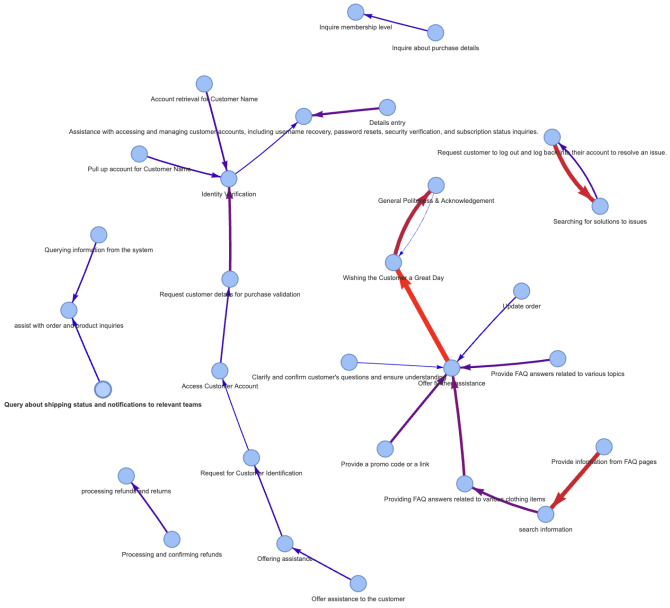
The Top-K Filtering method significantly reduces the complexity of the graph by retaining only the most important transitions. It excels in terms of delta-hyperbolicity, achieving a perfect score of 0, which indicates that the graph closely resembles a tree. However, this method also introduces a major limitation: it results in disconnected subgraphs. As shown in Figure 5, although the number of cycles is reduced to 2, the disconnected nature of the graph hinders its ability to

Fig. 4: Conversational graphs generated using threshold filtering for (τ = 0.1).



effectively capture the natural flow of conversation between intents and topics.

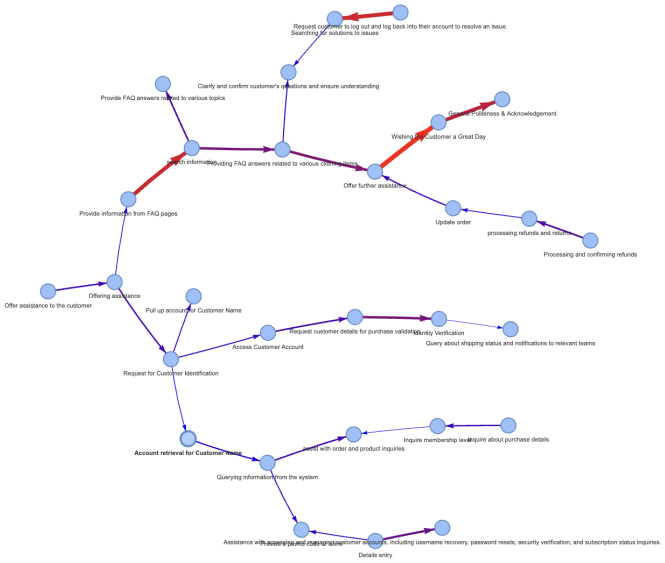
Fig. 5: Conversational graphs generated using Top-K filtering for (top-k = 1 and τ = 0.1).



C. Filter&Reconnect Results

The Filter&Reconnect method produces the most interpretable and coherent conversational graph. As shown in Figure 6, this method ensures a tree-like structure by eliminating all cycles (0 cycles) while preserving meaningful branching of intents. The resulting graph showcases a clear conversational flow, making it highly readable and interpretable. In addition, the delta-hyperbolicity metric remains at 0, indicating that the

Fig. 6: Conversational graph generated using the Filter&Reconnect method for (top-k = 1 and τ = 0.1).



graph closely approximates a tree structure, which is ideal for representing the flowing nature of conversation intents.

D. Comparative Analysis

Table II summarizes the performance of the three graph simplification methods across key metrics: δ-hyperbolicity, branching factor, and the number of cycles. As the table demonstrates, the Filter&Reconnect method outperforms the other techniques, yielding a graph with no cycles, a reasonable branching factor of 0.97, and a delta-hyperbolicity of 0, indicating that it most closely approximates an ideal tree structure. In contrast, the Threshold Filtering method suffers from excessive noise, resulting in a high number of cycles and poor hyperbolicity, while the Top-K Filtering method, despite achieving perfect hyperbolicity, produces a graph that is disconnected and less informative, as seen in Figures 4, 5, and 6.

Overall, the Filter&Reconnect method is the most effective for producing readable and interpretable conversational graphs, making it the recommended approach for this type of dataset. While the Top-K Filtering method has certain strengths, such as perfect hyperbolicity, its disconnected structure limits its utility. Finally, the Threshold Filtering method, though simple, suffers from noise and high cyclicity, making it less suited for generating interpretable conversational graphs.

VI. Conclusion

In this study, we presented a novel computational approach to modeling quasi-patterned conversational flows using conversational graphs. By applying advanced text embedding techniques and clustering methods, we effectively extracted conversational intents and transitions to build interpretable graph representations. Through the comparative analysis of

TABLE II: Best-Performing Combination for Each Method

Method	δ-hyperbolicity	Branching Factor	Number Of Cycles
Threshold Filtering	2.50	**2.41**	56
Top-K Filtering	**0.00**	0.86	2
Filter&Reconnect	**0.00**	0.97	**0**

The table contains the best-performing combination out of our search space for each method. The search space consists of $k = \{1, 2, 3, 4\}$ and $\tau = \left\{ \frac{0}{100}, \frac{1}{100}, \ldots, \frac{49}{100} \right\}$.

multiple graph simplification methods—Threshold Filtering, Top-K Filtering, and Filter&Reconnect—we demonstrated that the Filter&Reconnect method yielded the most readable and insightful graphs. This method preserves meaningful transitions between conversational intents while eliminating noise, resulting in a tree-like structure that aligns well with the quasi-patterned nature of the dataset.

Our approach offers a solution for analyzing large-scale dialogue datasets, providing valuable insights into conversational dynamics that can be leveraged to enhance automated systems, such as customer support agents and dialogue management systems. The successful application of our method to the ABCD dataset highlights its potential to uncover underlying conversational structures in other large-scale, loosely structured datasets, opening the door to further developments in conversational system optimization.

VII. Limitations

A. Parameter Sensitivity in Filtering Methods

While the Filter&Reconnect method proves effective in reducing noise, it requires careful tuning of parameters such as the threshold and top-k edges, which might not generalize across all datasets without manual intervention.

B. Assumption of Quasi-patterned Conversations

Another limitation is the assumption that conversations follow quasi-patterned flows, which might not be true for more free-form or highly dynamic dialogues. For such cases, more adaptive models that can account for greater variability in conversational paths may be necessary.

Acknowledgment

Achref thanks Callem AI and specifically Hamza Ghouili, CEO of Callem AI, for providing the resources and support necessary to produce this research. Gratitude is also extended to INSAT for the academic environment that fostered the development of this work.

References

[1] F. Murtagh and P. Legendre, "Ward's Hierarchical Agglomerative Clustering Method: Which Algorithms Implement Ward's Criterion?" *Journal of Classification*, vol. 31, no. 3, pp. 274–295, 2014. doi: 10.1007/s00357-014-9161-z.

[2] D. Chen, H. Chen, Y. Yang, A. Lin, and Z. Yu, "Action-Based Conversations Dataset: A Corpus for Building More In-Depth Task-Oriented Dialogue Systems," in *Proc. of the 2021 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Online, pp. 3002–3017, 2021. doi: 10.18653/v1/2021.naacl-main.239.

[3] Gemma Team et al., "Gemma 2: Improving Open Language Models at a Practical Size," arXiv, 2024. Available: https://arxiv.org/abs/2408.00118.

[4] A. Dubey et al., "The Llama 3 Herd of Models," arXiv, 2024. Available: https://arxiv.org/abs/2407.21783.

[5] S. Schuster, S. Gupta, R. Shah, and M. Lewis, "Cross-lingual Transfer Learning for Multilingual Task Oriented Dialog," in *Proc. of the 2019 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Minneapolis, MN, pp. 3795–3805, 2019. doi: 10.18653/v1/N19-1380.

[6] C.-W. Goo, G. Gao, Y.-K. Hsu, C.-L. Huo, T.-C. Chen, K.-W. Hsu, and Y.-N. Chen, "Slot-Gated Modeling for Joint Slot Filling and Intent Prediction," in *Proc. of the 2018 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, New Orleans, LA, pp. 753–757, 2018. doi: 10.18653/v1/N18-2118.

[7] H. Khanpour, N. Guntakandla, and R. Nielsen, "Dialogue Act Classification in Domain-Independent Conversations Using a Deep Recurrent Neural Network," in *Proc. of COLING 2016, the 26th International Conf. on Computational Linguistics: Technical Papers*, Osaka, Japan, pp. 2012–2021, 2016. Available: https://aclanthology.org/C16-1189.

[8] M. Gritta, G. Lampouras, and I. Iacobacci, "Conversation Graph: Data Augmentation, Training, and Evaluation for Non-Deterministic Dialogue Management," *Transactions of the Association for Computational Linguistics*, vol. 9, pp. 36–52, 2021. doi: 10.1162/tacl_a_00352.

[9] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings Using Siamese BERT-Networks," arXiv, 2019. Available: http://arxiv.org/abs/1908.10084.

[10] D. Arthur and S. Vassilvitskii, "K-Means++: The Advantages of Careful Seeding," in *Proc. of the Annual ACM-SIAM Symp. on Discrete Algorithms*, pp. 1027–1035, 2007. doi: 10.1145/1283383.1283494.

[11] M. Reid, N. Savinov, D. Teplyashin, D. Lepikhin, T. Lillicrap, J.-B. Alayrac, R. Soricut, A. Lazaridou, O. Firat, J. Schrittwieser, et al., "Gemini 1.5: Unlocking Multimodal Understanding Across Millions of Tokens of Context," arXiv, 2024. Available: https://arxiv.org/abs/2403.05530.

[12] S. Padmasundari and S. Bangalore, "Intent Discovery Through Unsupervised Semantic Text Clustering," in *Proc. of Interspeech*, vol. 2018, pp. 606–610, 2018.

[13] W. Chen, W. Fang, G. Hu, and M. W. Mahoney, "On the Hyperbolicity of Small-World and Treelike Random Graphs," *Internet Mathematics*, vol. 9, no. 4, pp. 434–491, 2013. doi: 10.1080/15427951.2013.828336. Available: https://doi.org/10.1080/15427951.2013.828336.

[14] J. R. Norris, *Markov Chains*, 2nd ed. Cambridge, UK: Cambridge University Press, 1998.

[15] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "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*, Minneapolis, MN, USA, 2019, pp. 4171–4186. doi: 10.18653/v1/N19-1423.

[16] R. A. Rossi, A. Zhou, and N. K. Ahmed, "Graph Representation Learning: A Survey," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 8, pp. 1793-1813, Aug. 2020. doi: 10.1109/TPAMI.2019.2921755.

[17] A. Ram, R. Prasad, C. Khatri, A. Venkatesh, P. G. Nagaraju, and J. W. F. Liu, "Conversational AI: The Science Behind the Alexa Prize," in *ArXiv Preprint arXiv:1801.03604*, 2018. Available: https://arxiv.org/abs/1801.03604.

[18] Z. Liu, P. Xu, A. Fung, M. Lewis, and P. Hase, "Dialogue Policy Learning with Action Embeddings," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Punta Cana, Dominican Republic, 2021, pp. 2474–2484. doi: 10.18653/v1/2021.emnlp-main.199.