UNIVERSITÀ DEGLI STUDI
DI TRENTO

DEPARTMENT OF INFORMATION ENGINEERING AND COMPUTER SCIENCE
ICT International Doctoral School

What is the Multi-Party Trick?

Language and Interactions in Multi-Party Conversations during the Era of LLMs

Nicolò Penzo

Advisor

Sara Tonelli, PhD

Fondazione Bruno Kessler, Trento, Italy

Co-Advisors

Marco Guerini, PhD

Fondazione Bruno Kessler, Trento, Italy

Bruno Lepri, PhD

Fondazione Bruno Kessler, Trento, Italy

prof. Goran Glavaš

University of Würzburg, Würzburg, Germany

April 29th, 2026

Acknowledgements

This PhD journey has been so much more than just a cycle of study, planning, observation, writing, presentation, and then starting all over again. Coffee, long and impromptu discussions in the office, beers after deadlines, competitive games of "biliardino" and many other shared moments have characterised this trip, made meaningful by the many people who have been part of it along the way.

First, I want to thank Sara, for trusting me during all these years, for supporting my crazy (and sometimes absurd) ideas, and for complaining only about my fixation with bullet points. Thank you for the advice and the trust you gave me, even in the moments when things did not seem to work.

Thanks to Marco for hosting me for more than two years, sharing the deadline screams and delirium, moments of collective craziness, and an impressive collection of dad jokes that somehow always arrived at the "perfect" time.

Thanks to Bruno, especially for helping me at the very beginning of my period in Trento, for the chit-chats, and for letting us use the NERF as a team building activity.

I have to thank all the three of you for the ideas, the guidance, the trust in my "travels abroad" and the many side-skills which I could only have learned from you.

Thanks also to Goran for hosting and welcoming me during my Bavarian period, for helping me both in the work and in the German-style life. Thanks also for the dinners at Habaneros, and our lively discussions, sometimes in Italian.

A special thank you (and an apology) goes to the people who worked more closely with me. I hope the fun outweighed the anxiety and the craziness.

Thanks to Prof. Anne Lauscher and Prof. Luca Maria Aiello for reviewing my thesis and providing valuable feedback on how to improve it. Thanks also to Prof. Alessio Palmero Aprosio and Prof. Oscar Araque for being part of the final committee.

Beyond all the people who "officially" contributed to my PhD work, there are many colleagues and friends (old and new) who accompanied me throughout these years. Their presence made the most stressful and demanding periods lighter, turning challenges into shared experiences and everyday moments into memories I will always remember.

First of all, thanks to the members of the DH, LanD, and MobS groups for all the coffee breaks, lunch breaks, and really-every-kind of break in between. You taught me that the office is not just a workplace. I am especially grateful for welcoming me into a new city, introducing me to new people, and helping me feel at home from the very beginning (I will not answer to the question "so what's your favourite group?").

I want to thank also the WueNLP group members, who expose me to different perspectives on work and life, and put up with me complaining about the German food choices (and for the long discussions around "why fries on pizza?").

To the PhD friends & co., you have shaped my attitude and been there through it all. Over these years, we've shared countless crazy moments and unforgettable times together. Priceless moments, far richer and more fun than I could have ever imagined.

The same goes for my Würzburger friends, with whom I've shared music, dinners, countless beers, deep conversations, and little adventures exploring Germany. Your company made my time there unforgettable and full of laughter, learning, and friendship.

I also want to thank all the friends I've made along the way, scattered across the world, whom I met at conferences and similar events. It's always a joy to run into you unexpectedly in the most unusual places around the world, turning brief encounters into memorable moments. A big thank you also to the Trento life-related people, who allowed me to discover new sides of myself, pushing me to grow in ways I would have never seen myself and making this part of the journey truly transformative.

Last but not least, the friends from my hometown (or close by), even if most of you don't know what I've done in these years.

If you don't fall into any of the categories above, but we have shared laughs, advice, good times, good vibes, or you host me on your sofa (there is a long list), this acknowledgment goes to you as well: thank you for being part of this journey in your own special way.

Un ringraziamento speciale va anche ai miei genitori, Antonella e Elvis, per avermi sempre creduto e supportato (o sopportato, a vostra scelta), seppur non abbiate prove vere di ciò che abbia fatto in questi anni, e a mio fratello Daniele per aver provato a spiegar loro.

Però vi giuro che questa è l'ultima.

Abstract

Multi-Party Conversations (MPCs) are a common conversational scenario taking place in everyday life, from informal spoken discussions to large-scale debates on social media platforms. Modeling and processing such type of conversations provide important challenges for Natural Language Processing (NLP) systems, as they require to go way beyond the analysis of the single messages, by taking into account also the conversational context and the interactional structure. Despite recent advances in Large Language Models (LLMs), how such dimensions should be effectively modeled and evaluated is still an open research direction.

In this thesis, we analyze the modeling, evaluation, and generation of written multi-party conversations, i.e., conversations where interactions happened through written messages, making all the information available in textual form, without the presence of a physical environment or possible visual cues, focusing on social media data.

First, we study how conversational context can be incorporated to perform downstream classification tasks, by including textual, interactional, and temporal information. Through extensive experiments and analyses, we show that while contextual modeling can improve robustness and stability, it requires significantly higher amount of training data with respect to non-contextual solutions. Moreover, we demonstrate that macro-level evaluation metrics fail to show important performance variations across conversations with different structural complexity.

Second, we analyze the ability of LLMs to model and predict conversational components, through proxy tasks like addressee recognition and response selection, in zero-shot settings. We also assess the ability of LLMs in generating useful summaries of conversations and useful descriptions of users' behavior. We propose a diagnostic evaluation framework that analyzes model sensitivity to prompt formulation and interactional structure, revealing systematic performance gaps that are not captured by standard benchmarks.

Third, we address data scarcity and evaluation limitations by exploring the use of LLMs to generate large-scale synthetic MPC datasets under explicit structural constraints. Our results show that LLMs can produce structurally diverse and controllable conversations, especially by using a multi-step generation strategy. However, human evaluation highlights

persistent challenges in assessing conversational quality and naturalness, especially when it comes to the interactional dimension, with respect to real world MPCs, as well as risks related to subjectivity and synthetic bias.

To mitigate these limitations, we introduce a Human–AI collaborative platform that supports the creation and refinement of high-quality linear multi-party conversations from reply trees, by combining tree visualization, human supervision and LLM-assisted refinement. This approach improves controllability, transparency, and decision-making for creating conversational interactions, while offering a practical solution to scalable dataset creation.

Overall, this thesis highlights the importance of interaction-aware modeling, fair evaluation protocols, and data diversity for advancing MPC research. The findings of this work open several directions for future research at the intersection of language modeling, conversational interaction analysis, and human–AI collaboration for the creation of synthetic MPC datasets.

Keywords

Multi-Party Conversations, Large Language Models, Interaction Networks, Conversational Evaluation, Synthetic Data Generation

Contents

List of Tables

List of Figures

Chapter 1

Introduction

> *Our talk exchanges do not normally consist of a succession of disconnected remarks, and would not be rational if they did. They are characteristically, to some degree at least, cooperative efforts; and each participant recognizes in them, to some extent, a common purpose or set of purposes, or at least a mutually accepted direction.*
>
> Grice (1975)

A sentence does not carry its meaning just on its surface. Its interpretation depends on how, where, and by whom it is used. As Wittgenstein (1953) writes, "for a large class of cases (though not for all) in which we employ the word *meaning* it can be defined thus: the meaning of a word is its use in the language". When we converse, we do not only exchange grammatically well-formed strings of words; we participate in what Wittgenstein calls language-games, in which utterances are used as jokes, complaints, invitations, or questions, depending entirely on the context. Similarly, a conversational turn is effective not only because it is syntactically correct, but mostly because it is understood inside a specific background of intentions, norms, and mutual expectations. Moreover, conversation is not just a mechanical transmission of sentences, but rather a cooperative activity.

Early philosophical reflections on the nature of conversation focused almost entirely on human-to-human communication. With the technological advances of the second half of the twentieth century, a new interlocutor entered the scene: the machine. If meaning arises through our participation in language-games, what does it mean for an artificial system (one that does not share our form of life and physical environment) to "converse"? Can an artificial system "play" with the language? And if so, which aspects of conversational skills and information are required for such participation?

Human–machine conversation became a practical reality with the release of ELIZA

by Weizenbaum (1966). Following this milestone, research interest in Natural Language Processing (NLP), particularly in conversational contexts, grew rapidly. As noted by Jurafsky and Martin (2025), new models emerged at an increasing pace, exploring specific subfields such as: frame-based paradigms for conversation modeling (Young et al., 2013); hierarchical structures of dialogue (Lu et al., 2022); plan-based approaches to dialogue management (Santos Teixeira and Dragoni, 2022). During the 1990s, machine learning techniques began to be applied to tasks such as slot filling and dialogue act detection, leading to significant advances in dialogue state tracking (Lee et al., 2021; Jacqmin et al., 2022). In the early 2000s, researchers also began to explore reinforcement learning as a framework for optimizing conversational strategies (Schatzmann et al., 2006; Chawla et al., 2023).

One of the most important aspects in NLP systems, including the ones employed for dialogues, is how words are represented in a vectorial space and how the system performs tasks on sentences. These two aspects are called, respectively, *text embedding* (Apidianaki, 2023) and *language modeling* (Chang and Bergen, 2024). These two aspects share several underlying techniques nowadays. Indeed, after the advent of transformers (Vaswani et al., 2017), model architectures started to learn embeddings and language modeling objectives jointly, by encoding lexical, syntactic, and semantic information within the same representation space, using it also to predict missing tokens, generate coherent continuations, or classify entire sequences (Bai et al., 2023).

With the release of transformer-based language models like `BERT` (Devlin et al., 2019), `GPT2` (Radford et al., 2019), and their successors, downstream classification and generation tasks became easily accessible through simple fine-tuning on relatively small, task-specific datasets. `GPT2` also popularized the idea of zero-shot learning in NLP, performing tasks based only on a natural-language description, although early results were far from being competitive. Moreover, the introduction of `GPT3` (Brown et al., 2020) demonstrated that few-shot learning, i.e., providing only a few in-context examples, could lead to strong performance across many NLP tasks. At the "`GPT3` stage", models reached scales of hundreds of billions of parameters. Building on top of `GPT3`, OpenAI released ChatGPT [1], which put large language models into mainstream public opinion. Following its release, numerous companies accelerated the development of increasingly larger models and began training them on instruction-tuned datasets, giving rise to the line of instruction-based or instruction-following models (Lou et al., 2024).

According to such developments, in this thesis we refer to pre-`GPT3` models such as `BERT` and `GPT2` as Pretrained Language Models (PLMs), and to models emerging after `GPT3` (i.e., characterized by larger training data size, larger number of parameters, in-context

[1] `https://openai.com/index/chatgpt/`

learning capabilities and eventual instruction tuning) as Large Language Models (LLMs). These advances in NLP technologies have also led to significant improvements in dialogue systems, particularly for human–AI interactions in one-to-one settings (Mendonça et al., 2024). As a result, people increasingly rely on AI assistants across a wide range of scenarios today, due to their growing reliability and effectiveness in performing a wide range of tasks (Hu et al., 2025).

Still, *conversations involving multiple participants remain a challenging, and extremely important, scenario to address.* In such settings, the meaning of the messages is deeply shaped by the interactional dynamics that unfold among participants, influencing both the interpretation and the pragmatics of each sentence. A large portion of these conversations takes place on social media platforms, a context that is particularly suitable for LLMs due to the absence of physical environment cues and speech cues (Laskowski, 2010; Shu et al., 2023).

For humans, reading such online conversations naturally allows them to infer what messages mean, what intentions they convey, and what underlying attitudes or sentiments they express. This understanding arises not only from what is written, but also from who wrote it and to whom it is addressed. This is crucial, as the same message may carry entirely different meanings depending on the surrounding conversational and social context. However, developing computational models capable of achieving human-level understanding of such interactions remains extremely challenging (Hovy and Yang, 2021). Even more complex, and often debated, is the question of how to evaluate these models in a fair and meaningful way.

The term Multi-Party Conversations (MPCs) is used to categorize this type of social interaction. Such conversations exhibit two defining characteristics (Traum, 2003):

I. *multi-party*, because they involve several participants (typically more than two) forming a diverse and interactive group;

II. *multi-turn*, because numerous responses unfold from a single initial message, giving rise to ongoing and dynamic exchanges among participants.

Effectively modeling multi-party conversations goes beyond understanding the meaning of individual sentences. Conversational context is fundamental for properly identifying and mitigating harmful phenomena such as hate speech, misinformation, and disinformation (Shu et al., 2020; Yu et al., 2022; Yang et al., 2023), as well as for designing effective counter-narratives and intervention strategies (Baez Santamaria et al., 2024; Chen et al., 2025). Beyond content moderation, modeling correctly the conversational context is also fundamental for effectively simulate social dynamics and forecasting user behavior under specific conversational configurations (Zhou et al., 2024). Recent advances in LLM capa-

bilities have further encouraged researchers to employ these models to simulate complex social scenarios, including settings that incorporate physical or environmental aspects (Park et al., 2023).

At the same time, the increasing deployment of LLM-based agents as active participants in online discussions, particularly on social media platforms, raises significant ethical challenges. These systems can be used to manipulate conversations by steering discourse or reinforcing specific viewpoints Breum et al. (2024), intentionally or not, often by amplifying harmful content or polarizing narratives (Ferrara et al., 2016; Shao et al., 2018; Caldarelli et al., 2020; Piot and Parapar, 2025). Addressing such ethical concerns requires a deeper understanding of LLMs behaviors in multi-party conversational dynamics with robust evaluation frameworks.

1.1 Context and Research Questions

We began our research journey in November 2022, shortly before the release of ChatGPT and the rapid rise of Large Language Models (LLMs). At that time, the dominant NLP systems were based on `BERT` and `GPT2`–style architectures. Throughout 2023, the NLP field witnessed an unprecedented acceleration in the release of LLMs by universities and companies, deeply reshaping both the state of the art and the scientific debate (Fan et al., 2024). This trend continued through 2024 and 2025 (Xiao et al., 2025).

The evolution of this thesis reflects such developments. The first stage of our work exploits a `BERT`-based pretrained language model, `RoBERTa` (Liu et al., 2019), relying on task-specific fine-tuning (Chapter 3). The second stage is based on `Llama2-13b-chat` (Touvron et al., 2023), which was among the most promising LLMs at the time, though it has been surpassed by newer generations since then (Chapter 4). In the third stage, we adopt more recent instruction-based models such as `Llama3.1` (Dubey et al., 2024) and `Qwen2.5` (Yang et al., 2024), shortly before the emergence of the first reasoning LLMs (Chapter 5). Finally, in the last stage of the research, we employ `Llama4-Maverick` (Meta, 2025), which incorporates mixture-of-experts mechanisms (Chapter 6).

Moreover, following what was outlined above, it becomes evident that, at the beginning of our research journey, the literature still presented numerous open questions concerning multi-party conversations, textual information, and interactional structures. Although some prior work had already attempted to incorporate interactional aspects, often through Graph Neural Networks (Sun et al., 2021; Gu et al., 2022a) or other structured models (Rahimi and Litman, 2020; Gu et al., 2021), very few studies had directly examined the actual usefulness of contextual information. Among those that did, most reported negative or inconclusive results (Menini et al., 2021; Anuchitanukul et al., 2022), leaving

a substantial gap in our understanding of when and why conversational context matter for downstream tasks. For these reasons, as a first step we tried to answer the following question:

Research Question 1 (RQ1). To what extent can transformer-based language models capture the interactional structure of multi-party conversations, and under which conditions does incorporating interactional information lead to significant improvements on downstream tasks?

Our findings (Chapter 3) show that, for MPC models to meaningfully leverage interactional information, a substantially large amount of training data is required, far exceeding the size of typical datasets used for standard downstream tasks. At the same time, we demonstrated that transformer-based models are capable of encoding interactional features, achieving more stable and robust performance across MPCs of varying structural complexity.

Building on these insights, we extended our investigation to LLMs in a zero-shot setting, trying to understand how to properly evaluate LLM performance on MPCs. In particular, we examined how results vary across different prompt formulations and across MPCs that differ in their interactional complexity. Indeed, in the second step of this research journey we tried to answer the following question:

Research Question 2 (RQ2). How do LLMs perform in zero-shot classification of multi-party conversations, and how does their performance vary based on the textual and structural information provided, the formulation of prompts, and the structural complexity of the conversations themselves?

Through this work (Chapter 4), we highlighted several limitations in current MPC research, especially the reliance on macro-level evaluation metrics that fail to capture how model performance varies across conversations with different interactional complexities. Together, the findings from RQ1 and RQ2 point out two major issues: the need for substantially larger training data to fully exploit interactional aspects, and the lack of datasets that adequately represent the wide variety of possible conversational structures and speaker-interaction patterns. These concerns raise doubts about the different scenarios' representativity in existing resources and emphasize the need for scalable methods to obtain large quantities of high-quality, structurally diverse MPC data. Motivated by these insights, in the third step we tried to answer the following question:

Research Question 3 (RQ3). How can LLMs be effectively used to generate large, high-quality synthetic multi-party conversation datasets that satisfy predefined structural and stance constraints, what generation strategies lead to the most reliable results, and how can we robustly evaluate the diversity and quality of the produced MPCs?

Our work (Chapter 5) shows that some LLMs can effectively generate high-quality synthetic MPCs with substantial structural variety, particularly when using a generation strategy with multiple substeps. Nonetheless, the resulting conversations are still far from being indistinguishable from natural human dialogues, and evaluating their quality remains challenging, since human assessments proved to be highly subjective and difficult to make more objective. Motivated by these findings, as a final step we explore the potential of exploiting a Human–AI collaborative approach for generating synthetic MPCs from reply trees, implemented through a dedicated platform we developed, LLMberjack (Chapter 6). This final step concludes our research journey, laying the groundwork for future developments and investigations.

1.2 Contribution

The research activities conducted in this thesis in response to the three research questions introduced above have resulted in three peer-reviewed conference publications.

- The first work addresses RQ1 by investigating the impact of training data size and contextual information on conversational context modeling tasks:

  Putting Context in Context: the Impact of Discussion Structure on Text Classification. Nicolò Penzo, Antonio Longa, Bruno Lepri, Sara Tonelli, and Marco Guerini. 2024. In Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1793–1811, St. Julian's, Malta. Association for Computational Linguistics.

- The second work addresses RQ2 by proposing a diagnostic framework for evaluating conversational component modeling tasks, explicitly accounting for interactional aspects of multi-party conversations:

  Do LLMs suffer from Multi-Party Hangover? A Diagnostic Approach to Addressee Recognition and Response Selection in Conversations. Nicolò Penzo, Maryam Sajedinia, Bruno Lepri, Sara Tonelli, and Marco Guerini. 2024. In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 11210–11233, Miami, Florida, USA. Association for Computational Linguistics.

- The third work addresses RQ3 by analyzing different LLMs and generation strategies for producing large-scale synthetic MPC datasets under predefined constraints:

  Don't Stop the Multi-Party! On Generating Synthetic Written Multi-Party Conversations with Constraints. Nicolò Penzo, Marco Guerini, Bruno Lepri, Goran Glavaš and Sara Tonelli, 2026. In Proceedings of the AAAI Conference on Artificial Intelligence, 40(39), pages 32701-32709, Singapore. Association for the Advancement of Artificial Intelligence.

In addition to these works, we developed a Human–AI collaborative platform that extends the contributions of the third study by supporting the semi-automatic creation and refinement of multi-party conversations. The platform enables guided interaction between human annotators and LLMs, facilitating the construction of high-quality MPCs starting from reply trees and addressing limitations observed in fully automated generation. This work is currently publicly available as a preprint:

- *LLMberjack: Guided Trimming of Debate Trees for Multi-Party Conversation Creation.* Leonardo Bottona, Nicolò Penzo, Bruno Lepri, Marco Guerini and Sara Tonelli, 2025. Preprint, arXiv:2601.04135.

1.3 Structure of the Thesis

Since this thesis is primarily intended for readers within the NLP research community, we assume prior knowledge of the core literature on LLMs and their underlying mechanisms. For these reasons, we do not provide an overview of basic LLM concepts. At the same time, we provide a brief introduction to concepts from network analysis and explain how we exploit such tools in the context of multi-party conversation research. Given these premises, this thesis follows the structure described below.

Chapter 2 — Background: What, How, and Where. In Chapter 2, we introduce the theoretical foundations underlying multi-party conversations, with a particular emphasis on their interactional structure. We present a formal network-based representation of these interactions and describe the network metrics that will be used throughout the subsequent chapters.

Chapter 3 — Modeling the Conversational Context. In this Chapter, we address the *Research Question 1* by fine-tuning a pretrained language model on multiple input configurations, each incorporating different contextual aspects of multi-party conversations. We evaluate how effectively the model leverages such context(s) by analyzing its performance

across varying training set sizes (learning curve analysis) and across conversations that differ in interactional complexity, length, and number of speakers.

Chapter 4 — Modeling the Conversational Components. In this chapter, we address the *Research Question 2* by developing a diagnostic evaluation pipeline for two key conversational component tasks, i.e., addressee recognition and response selection. Our analysis systematically examines model performance under different prompt formulations and across conversations exhibiting diverse structural characteristics.

Chapter 5 — Generating Conversations. In this chapter, we address the *Research Question 3* by examining four LLMs under two generation strategies, and evaluating the resulting synthetic MPC datasets in terms of constraint compliance, linguistic variability, emergent interaction structures, and human-perceived quality.

Chapter 6 — LLMberjack: Human-AI MPC creation. In this chapter, we introduce LLMberjack, our Human–AI collaborative platform designed to create linearized MPCs starting from reply trees. The system provides an interactive tree visualization to support annotators in understanding the conversational structure, and integrates LLM-assisted refinement for both messages and speaker profiles. In addition, we conduct an evaluation to assess the usefulness of the visualization component and the impact of LLM assistance on the annotation workflow.

Chapter 7 — Conclusion. In the final chapter, we summarize the main findings of our research journey, outlining the answers to the research questions provided throughout the thesis. We also discuss the new challenges and research directions that emerge from our work, highlighting the broader scenarios opened by our contributions.

Chapter 2

Background: What, How and Where

In this chapter, we introduce the theoretical bases about multi-party conversation research and the computational approaches developed to model and analyze them, which we will collectively refer to as Multi-Party Conversation Processing tasks. To this end, we first define the fundamental components of a multi-party conversation, then we discuss the main categories of processing tasks, and finally outline the sources and types of datasets from which multi-party conversations can be obtained. Throughout this chapter, we highlight the specific aspects addressed in this thesis and discuss the motivations behind such research choices.

2.1 What are Multi-Party Conversations?

Multi-Party Conversations (MPCs) are multi-turn conversations involving more than two participants (Traum, 2003; Branigan, 2006), which have been studied across multiple disciplines. Research in conversational analysis and linguistics has focused on modeling interaction dynamics (Sacks et al., 1974; Wilson et al., 1984), identifying participant roles (Malouf, 1995), or capturing emergent structural patterns in discourse (Gibson, 2003). These studies highlight both the complexity and the diversity among real-world MPCs, where elements like turn-taking succession, speaker alignment, and social context shape the flow of conversation.

Multi-party conversations are everywhere in everyday life, for example occurring as spoken interactions, such as those in online meetings or face-to-face group discussions, or as written interactions, for instance in social platforms (we will further develop this distinction in Section 2.3). These conversations can occur in a wide range of contexts, such as:

- unstructured exchanges in informal settings, characterized by spontaneous turn-

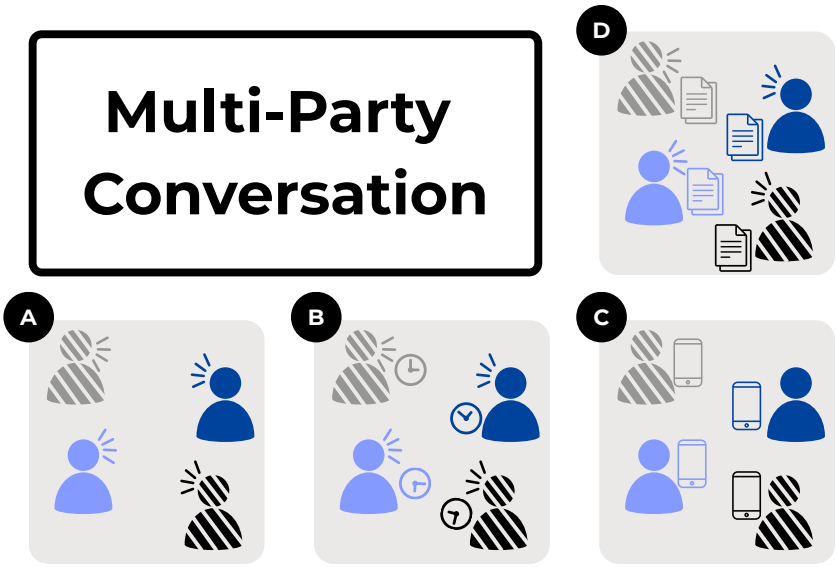

Figure 2.1: some examples of the different types of MPCs: spoken (A), structured debates (B), online-conversations (C), scripted (D).

taking, shared context and overlapping turns (Janin et al., 2003; Carletta et al., 2005);

- formal debates or structured discussions, where participants follow explicit rules and roles (Liang et al., 2024);

- scripted dialogues involving multiple characters in films, or media productions, reflecting designed conversational dynamics (Serban et al., 2016);

- large-scale online discussions on social media platforms such as Reddit or Twitter, where interaction is asynchronous, public, and often multi-threaded (Zhang et al., 2018a; Chang and Danescu-Niculescu-Mizil, 2019; Ouchi and Tsuboi, 2016).

Despite their pervasiveness, multi-party conversations differ substantially depending on their nature, organization, and underlying social dynamics. Indeed, each conversational setting imposes distinct structural and communicative constraints, affecting how participants interact, respond, and coordinate within the conversation (Mahajan and Shaikh, 2021; Pilan et al., 2024). In Figure 2.1 we report some examples of multi-party conversations. In this thesis, we will focus on Written MPCs, especially the ones obtained from online social media platforms.

2.2 Categories of Tasks on Multi-Party Conversations

This section is based primarily on the taxonomy proposed by Gu et al. (2022b), which we will leverage throughout the whole thesis as the main reference for categorizing the tasks involving multi-party conversations. Designing computational systems for MPC processing is inherently challenging, not only because the textual dimension is the result of multiple turns, but also because of the need to capture the underlying interactional aspects, e.g., the relationships that define who speaks to whom and how information flows across participants. Determining how these two dimensions, textual and interactional, should be effectively integrated for MPC processing, and assessing the role that large language models (LLMs) can play in this integration, remain open research questions. Tasks related to MPC processing can be broadly divided into two main categories: (I.) Conversational Context Modeling and (II.) Conversational Component Modeling.

Conversational Context Modeling. This category involves tasks that aim to capture high-level relational and structural patterns within a multi-party conversation, for example, by understanding how utterances, topics or emotions are connected throughout the conversation. The goal here is to model effectively the full conversation, capturing the contextual dependencies among utterances and understanding how the meaning evolves across turns and participants.

Examples of tasks in this category include *dialogue discourse parsing*, where the model must go beyond simple reply-to links and label the relations between utterances according to some criteria, like understanding the pragmatic of an utterance (e.g., elaboration, acknowledgment, comment; Chi and Rudnicky, 2022; Fan et al., 2023), and *information flow modeling*, which aims to track the propagation of content, emotions, or topics throughout the conversation, often as an intermediate step for performing other downstream tasks, such as stance detection or emotion recognition (Poria et al., 2019b; Li et al., 2021; Shen et al., 2021).

Conversational Component Modeling. This category includes all tasks that focus on modeling one specific component of the conversational triple that together define a turn in a multi-party conversation, i.e., (I.) speaker, (II.) addressee, (III.) utterance/message (we will further explain this in Section 2.4). Accordingly, these tasks can be grouped into three main subcategories: (I.) *speaker modeling*, i.e., tasks that aim to capture speaker-related properties, such as predicting who will take the next turn in the conversation (Laskowski, 2010; Enomoto et al., 2020; Castillo-López et al., 2025) or identifying the speaker of a given message based on linguistic or contextual cues (Gu et al., 2021); (II.) *addressee modeling*, i.e., tasks where the aim is to determine who a message is directed

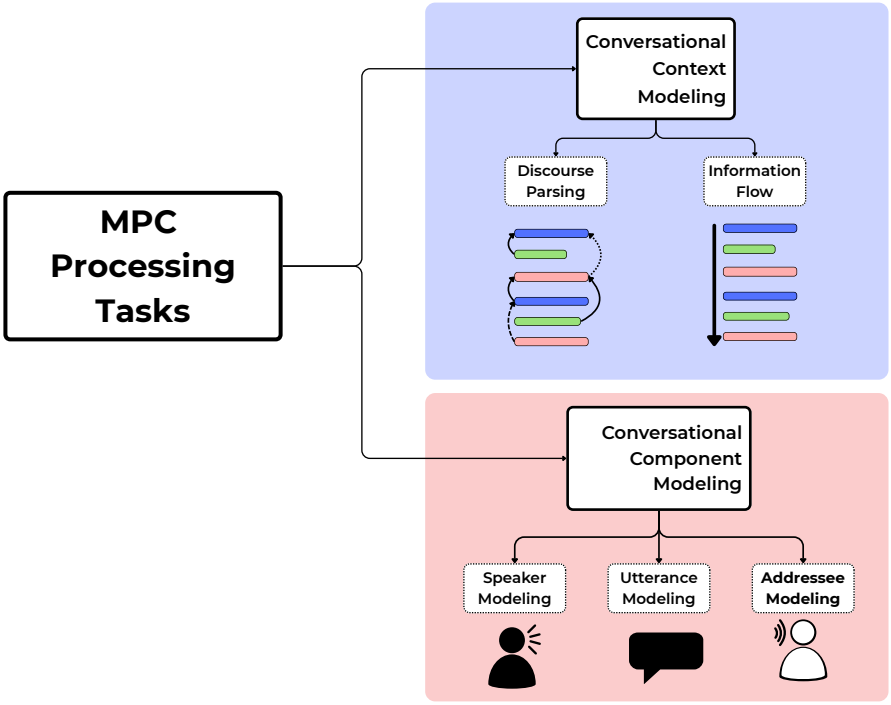

Figure 2.2: scheme of the MPC Processing tasks categorization followed in this thesis.

to, like explicit addressee recognition (Ouchi and Tsuboi, 2016; Gu et al., 2021; Zhu et al., 2023), or dialogue disentanglement (i.e., separating overlapping conversations within a single thread; Kummerfeld et al., 2019; Ma et al., 2022); (III.) *utterance modeling*, i.e., tasks focusing on the message component itself, such as response generation (Gu et al., 2022a; Mahajan and Shaikh, 2024), or next response selection, where the goal is to generate or identify (among some candidates) the most contextually appropriate continuation of the conversation (Ouchi and Tsuboi, 2016; Wang et al., 2020).

In this thesis, we will approach both categories of tasks. The first step will be into the Conversational Context Modeling (Chapter 3), by testing if pretrained language models like `RoBERTa` (Liu et al., 2019) can potentially exploit contextual information for performing downstream tasks like stance detection. Then, we test if LLMs (`Llama2-13b-chat` in our case, by Touvron et al., 2023) are able to model the context itself by predicting a component of the next turn, like the next addressee (addressee recognition task) or the next response (response selection task), given the next speaker (Chapter 4). We put particular emphasis on the evaluation side, especially for what concerns the correlation with structural metrics. In the past, researchers have debated around effective MPC evaluation methods, often emphasizing linguistic aspects or using candidate rankings (Mahajan

et al., 2022) or just looking at conversation length and number of users (Gu et al., 2023). However, there is a gap in evaluating MPC systems based on interactional aspects and structural metrics. In Chapter 3 we provide a first approach to structural analysis, by distinguishing complex and simple interaction graphs, where in simple ones each user contribute just one time and in complex ones at least one user provides multiple contributions. Then in Chapter 4 we go more in depth with the structural analysis, by: (I.) creating diagnostic datasets with a good structural variety, always in terms of interaction graphs, and (II.) by analysing the performance according to precise node-level network metrics (i.e., *degree centrality* and *average outgoing weight* of the speaker node).

Interactional analyses of social communication networks have primarily focused on interaction patterns across multiple conversations (Panzarasa et al., 2009; Coletto et al., 2017b; Garimella et al., 2018; Felmlee et al., 2021), confirming the relevance of such interactional structures in studying conversation dynamics. However, our focus is on interactions emerging within a single conversation rather than across multiple discussions, applying the same structural analysis techniques. Despite this shift in scope, the same structural analysis techniques can be effectively applied at the local conversation level.

2.3 Source of data for Multi-Party Conversation

Historically, the first MPC corpora were collected from in-person group meetings, like the ones presented in Janin et al. (2003) and Carletta et al. (2005). Over time, the focus has shifted strongly toward social media platforms (Mahajan and Shaikh, 2021), where large-scale conversational data are more easily accessible. This shift, however, has introduced several confounding factors. On many social media platforms, conversations are often constrained by a one-to-one reply structure, which overlooks implicit addressees and simplifies the natural dynamics of interaction. In contrast, real-world conversations frequently involve turns directed to multiple participants, with a flexible and dynamic conversational interaction that reflects more complex patterns of engagement (Gu et al., 2022b).

Mahajan and Shaikh (2021) proposed a taxonomy of MPC corpora distinguishing between *Spoken* and *Written* categories. The *Spoken* corpora are also further subdivided into *Scripted* and *Unscripted* types, reflecting whether the conversations are pre-planned or naturally occurring.

Written. Written multi-party conversations are those in which all interactions occur through text-based communication. These are typical of online platforms, where users exchange messages in a digital environment. The nature of the platform and its commu-

nity norms can lead to substantial differences in the interactional structure and dynamics of the conversations, as platform-specific constraints strongly influence how interactions are organized (Aragón et al., 2017). For example, platforms such as Reddit or Twitter/X implement reply-tree structures, where each message can start multiple branches of discussion, allowing any user to potentially engage in different subthreads. In contrast, Telegram or WhatsApp group chats involve a restricted set of participants communicating within a continuously evolving linear thread, where all users share the same conversational space, with possible parallel subconversations inside the same thread. These structural differences significantly impact the interaction flow, turn-taking behavior, and information diffusion within written MPCs.

Spoken. Spoken multi-party conversations are those in which interactions occur verbally, and the available data typically consist of transcriptions of the spoken dialogues. However, the textual transcript captures only a partial representation of the communicative event. Indeed, spoken conversations inherently involve a rich set of environmental and non-verbal factors that contribute to meaning and interactional structure. These may include physical or virtual settings (e.g., in-person meetings or online conferences), synchronous turn-taking, overlapping turns, and a variety of paralinguistic cues such as tone of voice, gaze direction, pauses, and gestures (Poria et al., 2019a; Zhou et al., 2021; Zheng et al., 2023; Jovanovic and op den Akker, 2004). This multimodal nature plays a crucial role in shaping the flow, coherence, and social dynamics of conversation, influencing how participants interpret, coordinate, and reply to one another. For these reasons, while transcripts provide valuable linguistic data, they often omit essential non-verbal dimensions that are important for understanding spoken interaction.

Inside this category, *Spoken Scripted MPCs* include planned and pre-written conversations designed to resemble natural interactions, such as movie or television scripts. Compared to their unscripted counterparts, these conversations tend to be less noisy and exhibit clearer turn boundaries, as speaker roles and turn-taking sequences are predefined. Moreover, the degree to which they reflect authentic conversational dynamics is debatable, since scripted dialogue may lack the spontaneity and variability typical of natural discourse. Well-known datasets of this type include the Cornell Movie-Dialogue Corpus (Danescu-Niculescu-Mizil and Lee, 2011), the TVD Corpus (Roy et al., 2014), and MELD (Poria et al., 2019a). *Spoken Unscripted MPCs*, in contrast, consist in natural, non-planned conversations, involving a wide range of situations, from informal dialogues and spontaneous discussions to formal meetings and panel interactions. These can occur either in real life or through virtual meeting platforms, preserving the spontaneity, interruptions, and coordination patterns typical of genuine spoken exchanges (Mlakar et al.,

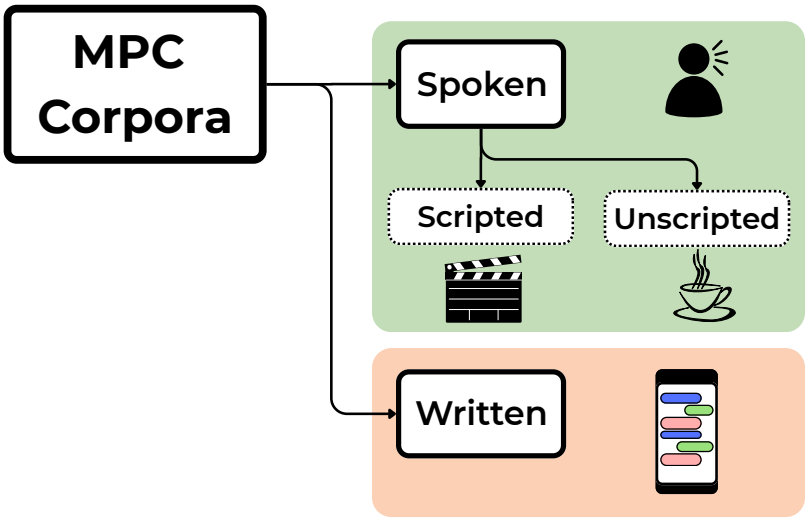

Figure 2.3: Scheme of the MPC Corpora categorization followed in this thesis

2023).

Since the objective of this thesis is to analyze conversations where all relevant information can be represent within a textual transcript, *the focus will be exclusively on Written Multi-Party Conversations*, which we will call generally multi-party conversations from now on. The primary goal is to model and capture the interaction dynamics as they unfold through text. Nevertheless, written multi-party conversations can also serve as a testbed for developing methods and representations that could later be extended to spoken multi-party conversation, where additional modalities, such as gaze, gestures or voice cues, would enrich the scenario and the complexity of the scene.

As previously said, most of the written MPC corpora come from social media platforms (SMPs). However, these platforms present several confounding factors. Indeed, SMPs often enforcing a one-to-one reply structure, overlooking implicit addressees and simplifying interaction dynamics; in natural conversations, in contrast, a turn is often directed to multiple participants and the conversational interactions are more dynamic (Branigan, 2006; Gu et al., 2022b). As a result, MPC corpora derived from social media platforms often lack structural diversity, which limits their utility in analyzing real-world conversational phenomena (Wei et al., 2023). Despite these limitations, online conversations remain a highly relevant subset of MPCs, as they constitute one of the main channels through which fundamental social and communicative phenomena occur. These include the spread of misinformation, fake news, rumors, and hate speech (Sheth et al., 2022), the expression of political leanings (Garimella et al., 2018), and even the disclosure of personal

or health-related information (Guntuku et al., 2017). For these reasons, designing systems capable of effectively modeling and analyzing this specific type of MPC is of substantial practical and ethical importance. Such efforts must not only address the structural constraints inherent to SMPs but also carefully consider privacy protection, bias mitigation, and responsible data use, given the sensitive nature of online conversational data (Dinan et al., 2021).

The wide retrieval of MPC data from social media platforms and their limitations can potentially affect generation capabilities of current LLMs. For LLMs trained on conversations from social media and mostly optimized for two-party interactions (i.e. human-assistant use-cases), multi-party conversations represent a distributional shift, resulting in their underwhelming performance in common MPC contexts (Tan et al., 2023). The next generation of LLMs is, however, expected to engage in multi-party settings and have success in tasks like identifying the appropriate speaker to respond to (Wei et al., 2023), summarizing meetings (Kirstein et al., 2024) or even managing multi-agent scenarios (Wu et al., 2023). Recent studies have explored their performance in social contexts (Ziems et al., 2024; Chang et al., 2024), emphasizing the need for large, representative datasets to train this novel generation of LLMs and to ensure robustness across diverse and less frequent interaction patterns (Lee et al., 2024).

Moreover, while social media allow for rapid collection of large-scale written MPCs, these datasets often come with incomplete interaction metadata. As affirmed above, many datasets record only explicit reply-to relationships, neglecting implicit addressees and richer conversational dynamics (Ouchi and Tsuboi, 2016; Zhang et al., 2018a; Chang and Danescu-Niculescu-Mizil, 2019). As a result, Wei et al. (2023) point out that well-known MPC corpora (Ritter et al., 2010; Baumgartner et al., 2020; Lowe et al., 2015) are useful for response generation, but not for more interactive tasks. Only most recent efforts are starting to focus on capturing conversational dynamics, going beyond text content (Hua et al., 2024).

2.4 Formalization of MPC structure and dynamics

Multi-party conversations can appear in different conversational structures in everyday interactions, particularly in online contexts. Depending on the social platform, conversations may be organized as linear threads, tree-like structures, or allow branching only at specific levels. In this thesis, we focus exclusively on MPCs structured as linear threads of ordered turns. It is important to note that each turn is not necessarily a direct reply to the preceding one, except in Chapter 3. From such conversational thread, it is possible to retrieve an interaction graph representing the relationships among the participating

speakers. In this Section we introduce a set of network-based metrics and a mathematical formalization of the interaction graphs, which will serve as the theoretical and analytical foundation for the experiments conducted throughout the thesis.

Formalization. Let a conversation be represented as $C = (M, U)$, where $M = \{m_1, m_2, ..., m_n\}$ denotes the set of chronologically ordered messages (m_i occurs before m_j if $i < j$) and $U = \{u_1, u_2, ..., u_p\}$ denotes the set of users participating in C. Each message m_i is associated with an ordered pair $(u_j, \bar{u}_j)$, where u_j is the speaker of m_i and $\bar{u}_j$ is the set of addressees of m_i, i.e., $u_j = S(m_i)$ and $\bar{u}_j = A(m_i)$.

The above definition provides a general framework that addresses all the cases appearing in this thesis. In subsequent chapters, we focus on specific subsets of this framework, providing more precise formalizations and, where appropriate, introducing additional labels.

From this formalization, an interaction graph can be retrieved from each conversation, where users correspond to nodes and messages correspond to edges. There is no unique representation of such a graph; the choice of representation can vary depending on whether exact timestamps, edge directionality, message frequency, or different levels of granularity are needed.

Throughout this thesis, no prior information about users is exploited, and original usernames are anonymized for all data collected online. This approach avoids introducing bias related to individual users and ensures fully privacy-preserving and profiling-preserving solutions. In addition, we replace each username with "local MPC IDs", which means that a user is assigned a new ID for each discussion they participate in. As a consequence, if a user is active in several discussions, this information is not available and user profiling at global network level (i.e., with information from different conversations of the same dataset) is not possible, thus enforcing privacy preservation. In this way, the models developed rely solely on the interactional dynamics of the conversation itself, without incorporating information about users' behavior in other conversations or any potentially biasing speaker-specific metadata.

2.4.1 MPC dynamics as interaction graphs

From a multi-party conversation $C = (M, U)$, an interaction graph $G(C)$ can be constructed to represent the interactional structure and the ongoing dynamics of the conversation. In this graph, each user $u_j \in U$ is represented as a node, and messages represent the directed edges connecting these nodes. Formally, for each message $m_i \in M$, let $u_j = S(m_i)$ denote the speaker and $\bar{u}_j = A(m_i)$ denote the set of addressees. Then, for

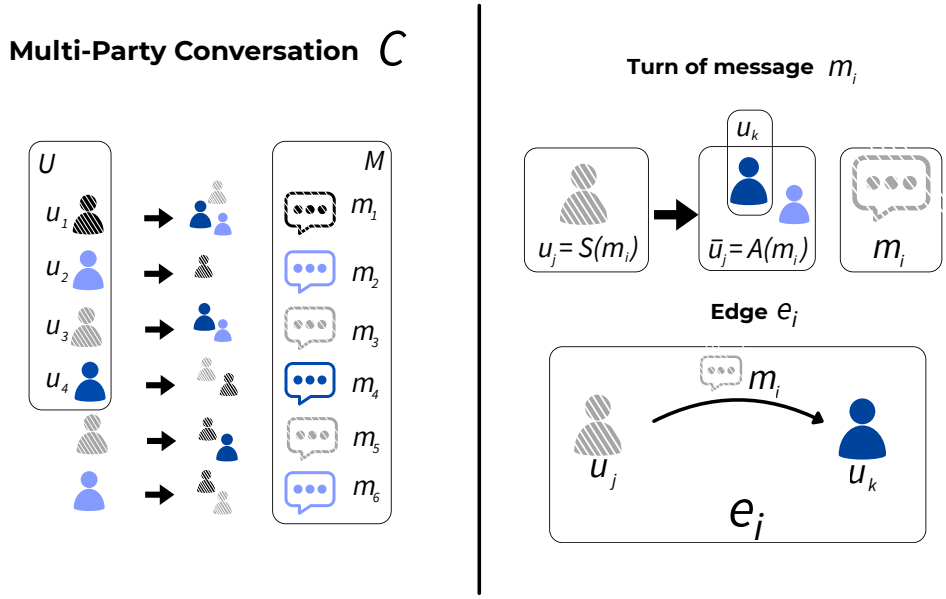

Figure 2.4: MPC representation and details about how one turn is mathematically formalized, and its representation in terms of interaction.

every addressee $u_k \in \bar{u}_j$, a directed edge $e_i = (u_j, u_k)$ is added to the graph, representing the interaction from u_j to u_k. Thus,

$$G(C) = (U, E) \quad \text{where} \quad E = \{(u_j, u_k) \mid \forall m_i \in M, u_j = S(m_i), u_k \in A(m_i)\}.$$

Multi-Edge directed graph. In its most general form, $G(C)$ can be modeled as a multi-edge directed graph $G_{me}(C) = (U, E_{me})$, where multiple edges between the same pair of nodes are allowed. Each edge corresponds to a distinct message, preserving the chronological order of communication between two users. Formally, if multiple messages are exchanged from u_j to u_k, the corresponding edges $\{e_{i_1}, e_{i_2}, ..., e_{i_t}\}$ are ordered according to their temporal sequence, i.e., if message m_{i_a} precedes m_{i_b}, then e_{i_a} precedes e_{i_b}.

Each edge can be associated with a set of attributes or labels that encode additional information about the message, such as the *timestamp* $t(m_i)$, i.e., the exact time at which the message was sent, or generic *message content features* $f(m_i)$ like stance, sentiment, or dialogue act type.

This representation does not only capture *who interacts with whom*, but also preserves the *temporal* and *interactional dynamics* of the conversation, enabling analyses of conversational flow, speaker influence, and interaction patterns over time (Hu et al., 2019;

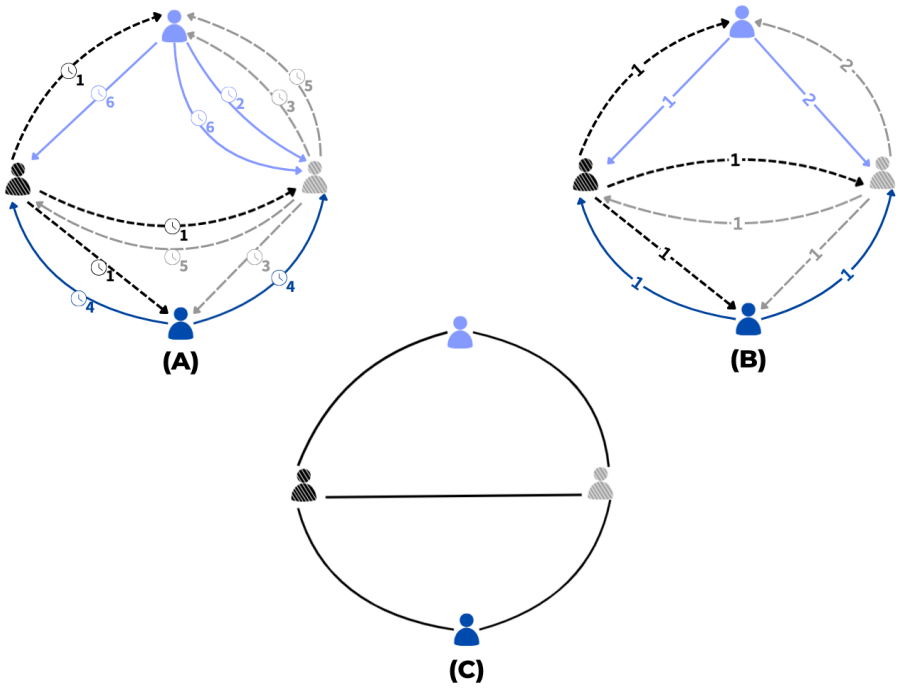

Figure 2.5: Examples of the different versions of interaction graphs discussed – (A) multi-edge directed graph, (B) weighted directed graph, (C) unweighted undirected graph.

Gu et al., 2022a). At the same time, it is harder to perform intuitive large-scale analysis because of the high level of detail. Indeed, the multi-edge graph can be further aggregated into a *weighted directed graph*, where edge weights encode the frequency of communication between pairs of users.

Weighted and Directed Graph. A *weighted and directed graph* can be derived from the multi-edge directed version by aggregating multiple edges between the same pair of users into a single directed edge. Formally, given the multi-edge directed graph $G_{me} = (U, E_{me})$, we define the corresponding weighted directed graph $G_d^w(C) = (U, E_d, W)$, where E_d is the set of unique directed edges between users, and $W : E_d \to \mathbb{R}^+$ assigns a positive weight to each edge.

Each edge $(u_j, u_k) \in E_d$ represents the overall interactions from user u_j (the speaker) to user u_k (the addressee), with its weight w_{jk} quantifying the amount of messages exchange between them in that specific direction, i.e.:

$$w_{jk} = |\{m_i \in M : S(m_i) = u_j, u_k \in A(m_i)\}|$$

This representation captures both *directionality* and *interactional intensity* among

speakers, offering a compact yet expressive abstraction of the underlying interaction graph. Weighted directed graphs are particularly useful for studying influence dynamics, centrality, and information flow within a multi-party conversation, as they enable the application of network analysis techniques to conversational data (Rahimi and Litman, 2020).

Unweighted and Undirected Graph. An *unweighted and undirected graph* provides a simplified abstraction of the interaction underlying a multi-party conversation, where only the existence of an interaction between users is considered, irrespective of its direction or frequency. This representation focuses solely on the presence of communication links among participants, thereby modeling the connectivity structure of the conversation.

Formally, given the set of users U and the message set M, the corresponding unweighted and undirected graph is defined as $G^{uw}_{ud}(C) = (U, E_{ud})$, where an undirected edge $(u_j, u_k) \in E_{ud}$ exists if and only if at least one message was exchanged between u_j and u_k, that is:

$$\{u_j, u_k\} \in E_{ud} \iff \exists m_i \in M$$

$$\text{such that } (u_j = S(m_i) \wedge u_k \in A(m_i)) \ \vee \ (u_k = S(m_i) \wedge u_j \in A(m_i)).$$

In this formulation, the edge $e_{jk} = (u_j, u_k)$ represents a nondirectional relationship, indicating that communication occurred between the two users, without specifying who initiated it or how frequently it happened.

This abstraction is particularly useful when the analysis aims to capture the overall social or structural connectivity among participants, rather than the detailed conversational flow. For instance, unweighted and undirected graphs are well suited for studying group cohesion or overall participation patterns within a multi-party conversation.

Although this representation ignores message ordering, temporal dynamics, and directionality, it offers a computationally efficient and conceptually clean view of the interaction graph.

2.4.2 Network Metrics and Their Meaning

Multi-party conversations exhibit diverse interactional patterns depending on their complexity and the degree of participant engagement (Cogan et al., 2012; Coletto et al., 2017a). To analyze how interactional complexity influences model performance in classification tasks, we employ network-based metrics derived from the interaction graphs defined in the previous section. While prior studies have investigated correlations between model performance and factors such as the number of speakers or conversation length (Gu et al.,

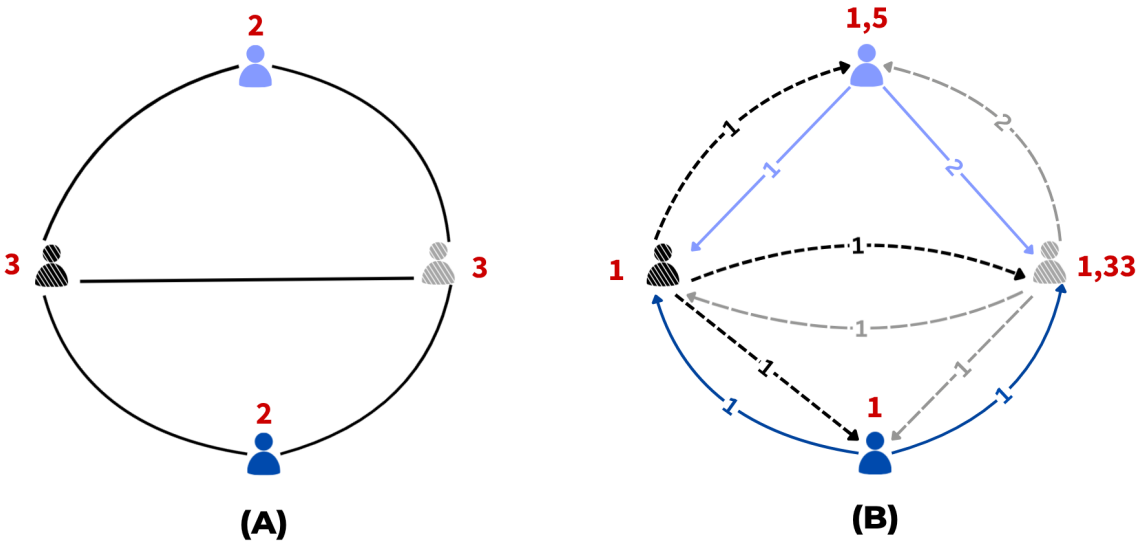

Figure 2.6: examples for node-level metrics (values reported for each node) – (A) degree centrality, (B) average outgoing weight.

2023), to our knowledge, network-theoretic measures have little been applied to quantify conversation complexity within this framework.

Node-Level Network Metrics

From the above interaction graphs, we derive two node-level network metrics that describe the conversational activity of each participant: the **Degree Centrality** and the **Average Outgoing Weight**. We report in Figure 2.6 a graphical representation of the metrics.

Degree Centrality. Given an unweighted and undirected graph $G^{uw}_{ud}(C) = (U, E_{ud})$, where U is the set of nodes and E_{ud} is the set of edges, the degree centrality $deg(u_i)$ of a node $u_i \in U$ is the number of edges $e_k \in E_{ud}$ incident in u_i, formally:

$$deg(u_i) = |\{u_j \in U \mid e_k = (u_i, u_j) \in E_{ud}\}|$$

This metric represents the number of distinct participants with whom user u_i has interacted. In the context of MPCs, it provides an estimate of how socially connected or central a participant is within the interaction graph.

Average Outgoing Weight. Given a weighted directed graph $G^{w}_{d}(C) = (U, E_d, W)$, where U is the set of nodes, E_d is the set of edges, and $W : E_d \to \mathbb{R}^+$ the weighting

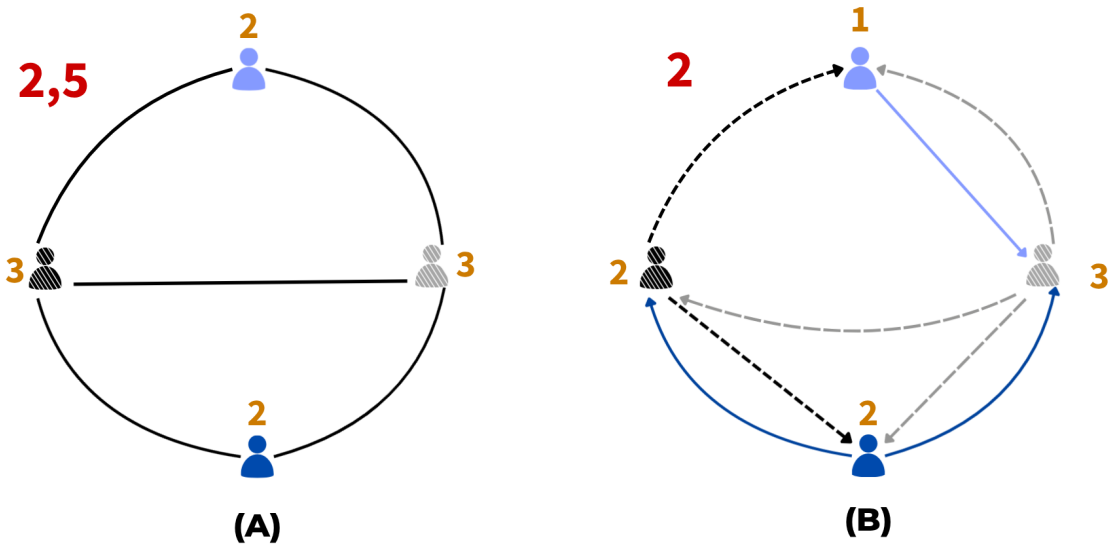

Figure 2.7: examples for global network metrics – (A) average degree centrality, (B) average outgoing degree.

function, the the out-degree centrality $outdeg(u)$ of a node $u_i \in U$ is the number of outgoing edges $e \in E_d$ for which u_i is the originating node/speaker, formally:

$$outdeg(u_i) = |\{u_j \mid (u_i, u_j) \in E_d\}|$$

Instead, the weighted out-degree $outdeg_w(u_i)$ is the sum of the weights on such edges:

$$outdeg_w(u_i) = \sum_{(u_i,u_j)\in E_d} w_{u_i u_j},$$

So the average outgoing weight $w^o_{avg}(u_i)$ is the average weight of the edges $e \in E$ for which u_i is the originating node/speaker:.

$$w^o_{avg}(u_i) = \frac{outdeg_w(u_i)}{outdeg(u_i)},$$

This metric captures the average number of messages sent by user u_i, on average, to each interlocutor they addressed, reflecting the intensity of the speaker's communication.

Global Network Metrics

To further characterize the interactional complexity of multi-party conversations, we consider global properties of the interaction graphs to assess higher-order patterns of interaction.

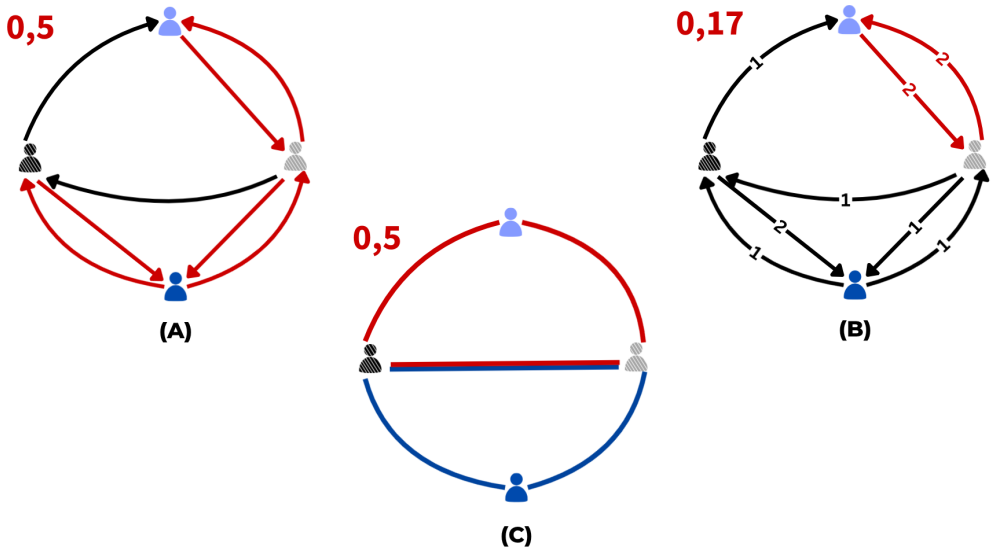

Figure 2.8: examples for dyadic and tryasic metrics – (A) reciprocity, (B) consistent reciprocity, (C) transitivity.

Average Degree Centrality. For the unweighted and undirected graph $G_{ud}^{uw}(C)$, the average degree centrality is:

$$deg_{avg}(G_{ud}^{uw}(C)) = \frac{1}{|U|} \sum_{u_i \in U} \frac{deg(u_i)}{|U| - 1}.$$

It represents the average number of distinct users each participant interacts with, normalized by the maximum possible degree. Higher values indicate broader participation and more interconnected conversations.

Average Outgoing Degree. For the weighted directed graph $G_d^w(C)$, the average outgoing degree is:

$$outdeg_{avg}(G_d^w(C)) = \frac{1}{|U|} \sum_{u_i \in U} \frac{outdeg(u_i)}{|U| - 1}.$$

This metric quantifies the average number of directed interactions per participant, thus capturing the directional flow of communication within the MPC.

Dyadic and Triadic Metrics

When two speakers, s_1 and s_2, reply to each other, they form a *cycle* (Coletto et al., 2017b), represented by a directed edge e_1 from s_1 to s_2 and a reciprocal edge e_2 from s_2 to s_1. If this back-and-forth exchange continues multiple times, the edge weights $w(e_1)$ and

$w(e_2)$ will both become greater than 1. We refer to such recurring exchange as *consistent cycles*.

To capture the presence of reciprocal and group interactions, we analyze dyadic (pairs of users) and triadic (triplets of users) structures within $G_d^w(C)$ and $G_{ud}^{uw}(C)$.

Reciprocity. Given the weighted directed graph $G_d^w(C)$, the reciprocity $R(G_d^w(C))$ is defined as the proportion of dyads $\{u_i, u_j\}$ such that both directed edges (u_i, u_j) and (u_j, u_i) exist:

$$R(G_d^w(C)) = \frac{|\{(u_i, u_j) \mid (u_i, u_j) \in E_d \wedge (u_j, u_i) \in E_d\}|}{|E_d|}$$

This measure reflects how often communication is bidirectional rather than one-sided. High reciprocity tells us that participants tend to respond to one another, engaging in back-and-forth exchanges rather than posting isolated messages.

Consistent Reciprocity. The consistent reciprocity $R^w(G_d^w(C))$ extends this notion by incorporating edge weights, capturing recurring bidirectional exchanges where $w_{ij} > 1$ and $w_{ji} > 1$. In other words, consistent reciprocity does not merely indicate that two users have exchanged messages in both directions at least once, but rather that they have done so repeatedly throughout the conversation, maintaining a focused interaction thread.

Transitivity. To quantify triadic connectivity, we compute the transitivity $T(G_{ud}^{uw}(C))$ of the undirected graph:

$$T(G_{ud}^{uw}) = \frac{3 \times \text{number of triangles}}{\text{number of connected triplets of nodes}}.$$

This metric measures the likelihood that two users who both interact with a third user also interact with each other. High transitivity values indicate that participants tend to form tightly connected clusters of interaction, where multiple users actively exchange messages with one another, reflecting collective engagement. In literature this is known also as clustering coefficient.

2.5 Outline of the thesis and challenges approached

As anticipated above, this thesis focuses on written multi-party conversations derived from social media platforms. In Chapter 3, the primary objective is to investigate how

and under which conditions pre-trained language models can effectively leverage conversational context when performing downstream classification tasks, specifically stance detection. The analysis begins with comparatively simple tasks, providing a controlled framework to assess the extent to which contextual, structural/interactional, and temporal cues influence model performance. Subsequently, Chapter 4 extends this investigation to conversational component prediction, focusing on tasks such as next-response selection and addressee recognition given the preceding conversational context. For both research steps, previous works have revealed a gap in evaluation practices, as most studies have overlooked the complexity of conversational interactions, thereby limiting the generalizability of their results. In both chapters, therefore, a central contribution of this thesis lies in analyzing how model performance varies in function of the interaction graph underlying a multi-party conversation.

As discussed in Section 2.3, most of the MPC datasets exhibit limited diversity in interactional patterns, often reflecting only a narrow range of social or conversational configurations. This limitation limits a fair evaluation of the ability of pretrained language models and large language models to generalize across less frequent (yet equally important) conversational scenarios, which should not be ignored only because they are underrepresented in available corpora. To address this issue, Chapter 5 investigates whether LLMs can be exploited to generate large quantities of synthetic MPCs, exploring a variety of generation strategies with constraints (given through natural language statements) and analyzing the interactional patterns that emerge. The study introduces the synthetic generation of turns with multiple addressees and incorporates both automatic analyses and human evaluations to ensure the quality and coherence of the generated conversations. At the same time, this approach raises important methodological and ethical questions concerning the subjectivity involved in evaluating conversational realism, even when assessments are performed along multiple, carefully defined dimensions.

Chapter 3

Modeling the conversational context

Online social media platforms host conversations that are nowadays fundamental for understanding human behavior and social phenomena. To monitor and analyze these phenomena, NLP systems must go beyond simple text classification, but most existing approaches still focus only on individual messages or, at most, pairs of comments, as in stance detection tasks, while largely ignoring the broader conversational context.

Yet, conversations are more than isolated messages. Knowing who interacts with whom, how often, and in what order can reveal patterns of influence, escalation, or consensus. Temporal dynamics highlight waves of activity, often pointing to triggering events, as shown in studies of online toxicity (Saveski et al., 2021) and the spread of misinformation (Vosoughi et al., 2018). These patterns provide a richer, wider picture of how dialogues unfold over time.

Before attempting to model these interactions, however, we must first ask a first basic question: does such contextual information actually help? This is especially true for multi-party conversations, where the structure of interactions (who addresses whom, and in which sequence) provides crucial contextual and pragmatic cues that can shift the meaning of messages.

Therefore, our first step is to evaluate the value of the interactional context. We aim to determine not just whether it improves downstream task performance, but also how it contributes and under what conditions it is most effective. By systematically answering these questions, we can quantify the importance of interactional structure and start working on models that actually integrate context, moving closer to capturing the richness of real-world multi-party communication.

In this chapter, we break down the *Research Question 1* presented in Section 1.1, which can be further defined in three sub-questions:

RQ(1.1): To what extent can transformer-based language models capture and exploit

contextual features in multi-party conversations?

RQ(1.2): How does the size of the training dataset affect a model's ability to effectively leverage contextual information?

RQ(1.3): How does the structural complexity of conversational interactions affect model performance when interactional information is available?

We addressed these questions by presenting a series of experiments aimed at modeling conversational context, embedding its *textual*, *temporal* and *interactional dimensions*, within a unified architecture.

To address **RQ(1.1)**, we design a framework where all contextual information is expressed in natural language and processed by a pretrained language model, specifically, `RoBERTa` (Liu et al., 2019), to perform classification tasks on pairs of messages, without explicitly modeling the latent structural dependencies of the interactions. In this setting, the conversational context is not strictly required to accomplish the task, permitting to identify a powerful and trustful baseline, but can potentially lead to improved performance by providing additional contextual cues. However, the benefit of this integration is not given for granted, as incorporating richer contextual information increases the dimensional complexity that the model must effectively manage.

In order to address privacy and profiling concerns, we deliberately avoid to provide user-related metadata. Instead, users are represented through "local MPC IDs", meaning that each participant is assigned a new, context-specific ID for every discussion in which they appear. Consequently, cross-conversation identity linking and global user profiling are not possible, ensuring privacy preservation while maintaining the interactional structure.

Given that prior studies have emphasized the importance of training data scale in enabling models to effectively capture contextual dependencies, our experiments primarily focus on a stance detection task using a large dataset collected from the Kialo platform (Scialom et al., 2020). The dataset is described in detail in Section 3.2, and an illustrative example of its discussion structure is shown in Figure 3.1.

To address **RQ(1.2)**, we further investigate the impact of dataset size by conducting an analysis of the learning curve (Section 3.5.2). For comparison, we also tested our approach on two smaller datasets targeting stance detection and abusive language detection, confirming the influence of training size on exploiting the contextual features (Section 3.5). Finally, to address **RQ(1.3)**, we evaluate model performance across multi-party conversations of varying complexity, measured in terms of interaction graph structure, number of user-turns, and number of users (Section 3.5.4).

The content of this chapter has been published in the paper *"Putting Context in*

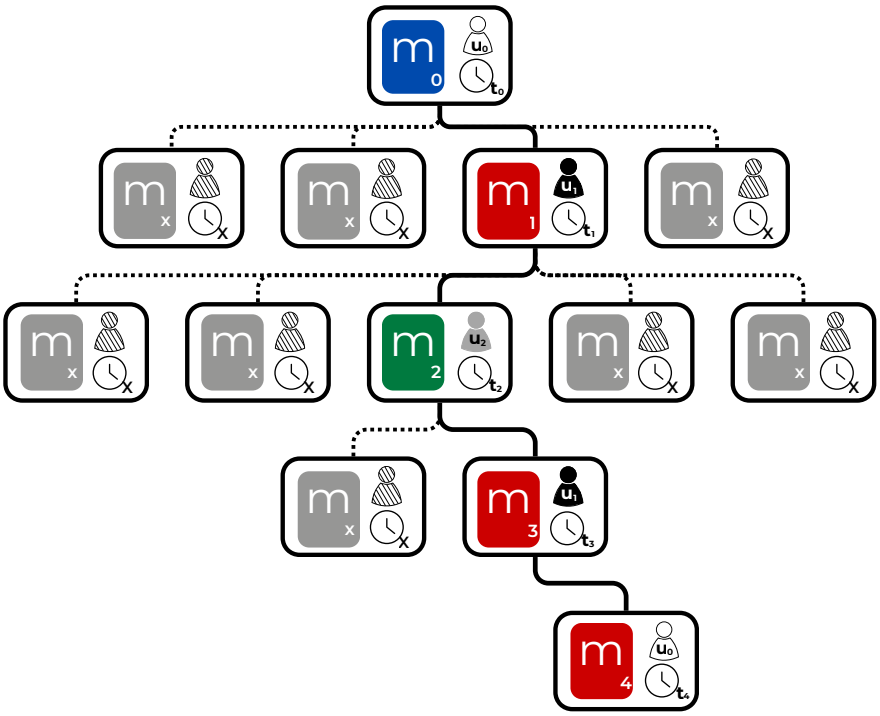

Figure 3.1: discussion tree from Kialo dataset. Each turn has a textual content (or claim) m_i, a user id u_j and a timestamp t_i. A *support* (green) or *contrast* (red) label with respect to the previous statement is assigned to each claim. The initial claim c_0 has no stance (blue). This representation can be easily generalized to experiments on other datasets.

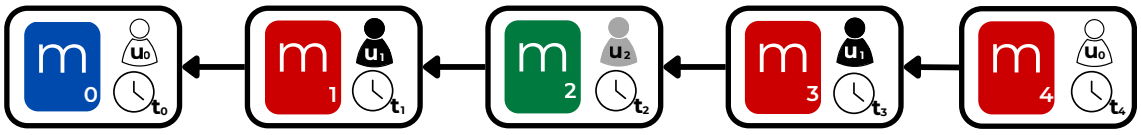

Figure 3.2: multi-party conversation retrieved from Kialo dataset. The conversation is extracted from the discussion tree represented in Figure 3.1.

Context: the Impact of Discussion Structure on Text Classification." in the Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Penzo et al., 2024a).

The software to perform the experiments is available on the dedicate Github repository at the link https://github.com/dhfbk/PuCC.

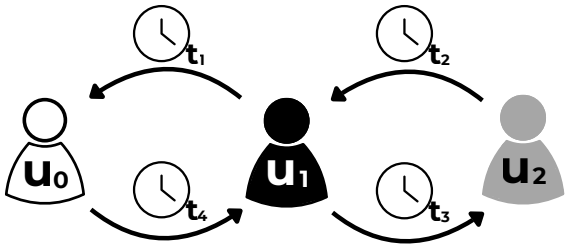

Figure 3.3: Interaction graph from Kialo dataset from the MPC reported in Figure 3.2.

3.1 Related Work on Conversational Context Modeling

In the literature, only a limited number of studies have attempted to integrate the linguistic, interactional, and temporal dimensions of MPCs within a unified modeling framework for downstream tasks. Some notable efforts have been made for dealing with downstream tasks like fake news detection (Nguyen et al., 2020; Song et al., 2021; Min et al., 2022), hate speech detection (Chakraborty et al., 2022), stance detection (Yang et al., 2019) and rumour verification (Zhou et al., 2019). Related work has also explored sentiment analysis and emotion recognition at the level of individual comments (Poria et al., 2019a; Firdaus et al., 2020). Moreover, user-related features have been successfully incorporated in abusive comment moderation , where modeling speaker-level information has been shown to enhance both the robustness and fairness of classification (Pavlopoulos et al., 2017). At the same time, the benefits coming from sociodemographic information for text classification on social media data are not clear, showing the difficulty to inject such information in pretrained language models (Hung et al., 2023).

All of these tasks are closely related to the dynamics of human behavior, yet the joint integration of linguistic, interactional, and temporal information has remained challenging due to several factors:

I. the fusion of heterogeneous knowledge, which requires combining computationally intensive models such as pretrained language models like `BERT` (Devlin et al., 2019) with Graph Neural Networks (GNNs; Zhou et al., 2020), as in Lin et al. (2021);

II. limited access to large-scale private data, which cannot be freely released for research;

III. the need for trained human annotators, which increases the cost and complexity of dataset creation;

IV. the ephemeral nature of social media posts, which leads to missing or deleted content

over time, creating gaps in discussions, particularly in datasets focused on hate speech and fake news (Klubicka and Fernández, 2018).

For a few shared tasks, datasets including contextual metadata, such as user IDs and timestamps, have been released (Gorrell et al., 2019; Cignarella et al., 2020). Nonetheless, the majority of research has remained focused primarily on the textual content, often overlooking the rich interactional and temporal dimensions inherent in multi-party conversations.

One reason why contextual information has been only marginally explored in classification tasks is that its benefits have not been demonstrated consistently. For instance, Menini et al. (2021) show that incorporating textual context does not yield performance gains in abusive language detection, even when the dataset is re-annotated with full conversational context. These findings are corroborated by Anuchitanukul et al. (2022), who further demonstrate that the effectiveness of contextual models is highly dependent on the intrinsic characteristics and size of the training data. In contrast, Yu et al. (2022) report that incorporating a short textual context, consisting of only the parent and target comments, can improve hate speech classification. However, their approach considers only the textual dimension and does not account for structural context within the conversation. Similar to our approach, Beck et al. (2023) also model contextual information in natural language. Nevertheless, their notion of "context" primarily relies on external knowledge sources, such as structured knowledge bases, causal relationships, or information retrieved from pretrained language models, rather than the intrinsic conversational structure.

Regarding stance detection, Agarwal et al. (2022) propose a graph-based inference model to predict the stance of a comment relative to its parent, leveraging graph walks to incorporate contextual information. Their experiments are conducted on a Kialo-derived dataset, as in the current chapter (see Section 3.2 for details). A related task is rumour verification, where the objective is to assess the truthfulness of a claim based on the reactions it elicits. In this case, the context is defined by the subsequent claims (i.e., the "right context") rather than the preceding claims (i.e., the "left context"), reflecting the focus on the effects produced by the original claim. To address this task, Tian et al. (2022) propose a model that combines `BERT` with a Graph Attention Network (GAT; Veličković et al., 2017), capturing both linguistic and extra-linguistic context. Notably, their approach considers the entire discussion tree but performs classification only on the initial claim.

In summary, previous studies that attempted to incorporate contextual information into classification tasks either failed to outperform text-only approaches or achieved improvements only through computationally intensive models, such as Graph Neural Networks (GNNs). Moreover, these approaches often relied on providing the model with

Figure 3.4: Example of supportive (green) and contrastive (red) claim having the same parent claim in Kialo.

all available information, including sensitive user-related data. In contrast, our approach demonstrates that *context benefits classification*, while modeling the diverse types of input in *natural language* and being *privacy-preserving*.

3.2 Kialo Dataset for Stance Detection

Kialo[1] is an online platform where people can debate around a main topic, with moderators being in charge of checking the grammaticality of the claims[2], evaluating the level of support or of contrast between a target claim and its parent claim, and even moving claims to make conversations more consistent. For these reasons, Kialo typically contains less noisy data and a clearer conversational structure with respect to other social media like Twitter/X or Reddit, being an ideal testbed for experiments and analyses.

In Kialo, the author of each claim is required to assign a *stance label* to it with respect to the parent claim. This label (*support* or *contrast*) is then checked by the moderator, who can change it if needed (an example of supportive and contrastive stance from the dataset is displayed in Figure 3.4). Discussions are originally arranged in a reply-tree structure, from which it is possible to retrieve several linear MPCs in thread shape. Datasets extracted from Kialo have already been used in the past to study the linguistic characteristics of impactful claims (Durmus et al., 2019a,b) or perform polarity prediction (Agarwal et al., 2022).

We obtained access to the dataset based on Kialo presented in Scialom et al. (2020), which was used for binary stance detection. Kialo discussions are originally structured as discussion trees. For each discussion tree, we extract all the linear threads going from the initial claim to the leaves. Consequently, it is possible for portions of these threads to overlap, while the target claims, with their respective labels, remain unique. This approach allows the model to process instances in which different MPC progressions result in different outcomes. We kept from the data only multi-party conversations with

[1]https://www.kialo.com

[2]In this chapter we will use the term claim interchangeably with message because of the terminology used by Kialo itself.

	SDK Dataset		
Set	**Contrast**	**Support**	**Total**
Training	49.2%	50.8%	122,681
Validation	50.2%	49.8%	7,447
Test	54.5%	45.5%	8,211

Table 3.1: Distribution of the labels in the Stance Detection Kialo (SDK) dataset.

more than 1 turn (i.e., having at least the initial claim and one reply). In this way, we obtain 122 681 training instances, 7 447 validation instances and 8 211 test instances. Each instance includes: I. the *target* claim; II. the *multi-party conversation*, from the *initial claim* to the *target claim*; III. the *stance* of each claim versus its parent claim; IV. the *user ID* of each claim; V. the *timestamp* of each claim. Given a multi-party conversation $d = \{\bar{c}_0, \bar{c}_1, ..., \bar{c}_n\}$ of length $n+1$ turns, the goal is to classify correctly the stance of the turn $\bar{c}_n$ with respect to $\bar{c}_{n-1}$, choosing between *support* (S) or *contrast* (C).

Formalization. Let $S = \{C_0, C_1, C_2, ..., C_S\}$ be a set of multi-party conversations, where each multi-party conversation is made of an ordered sequence of turns $C_i = \{\bar{c}_0, \bar{c}_1, \bar{c}_2, ..., \bar{c}_n\}$ where each turn $\bar{c}_i$ is a tuple $\{m_i, u_j, t_i\}$, where m_i is the textual message (or claim), u_j the local user ID of the speaker ($u_j = S(m_i)$) and t_i the timestamp ($t_i = t(m_i)$). The claim m_0 is called *initial claim*, and $\bar{c}_0$ is called *initial turn* for consistency. Each claim m_i is a response to the claim $m_{i-1} \forall i \geq 1$. Each multi-party conversation C_i has a label $y_i \in Y$, with $Y = [0, l-1]$ where l is the number of possible labels. The goal is to learn a function f that maps correctly each MPC to its correct label $f : D \rightarrow Y$.

In Kialo setting, the two labels, i.e., *contrast* (C) and *support* (S), are respectively mapped to $\{0, 1\}$. We report in Table 3.1 an overview of the label distribution in our dataset, which we call the Stance Detection Kialo dataset (SDK). Furthermore, to mitigate potential data contamination effects, the dataset is split according to the initial turn $\bar{c}_0$. As a result, all MPCs originating from the same initial turn are exclusively assigned to either training, validation, or test set.

3.3 Context Definition and Modeling

In past works, context has been integrated in MPC classification tasks using two main approaches: by combining linguistic and network information through the combination of node or network embeddings and textual embeddings (Shu et al., 2019; Dou et al., 2021) or by using textual embeddings as features in a network system, and retrieving a general

representation using GNNs or node/network embedding techniques (Yao et al., 2019; Lin et al., 2021).

We follow a third approach by expressing information on interactional and temporal context using natural language, and then giving it in input to a pretrained language model. We use a `RoBERTa`-based model (Liu et al., 2019) to perform the task. This allows us to keep the same classification framework while only changing the input data to progressively add contextual information, adopting a simple yet effective solution which is computationally lightweight.

Given a multi-party conversation $C = \{\bar{c}_0, \bar{c}_1, ..., \bar{c}_n\}$ of length $n + 1$, where $\bar{c}_i = \{m_i, u_j, t_i\}$, we can identify 3 different types of context: a linguistic (textual) context and two extra-linguistic (temporal and interactional) contexts.

3.3.1 Textual context

In our experiments, the textual context is defined as the sequence of all the claims in the discussion chain from m_0 to m_{n-2}, and it is added to m_{n-1} and m_n (i.e., the claims used for defining the stance). We concatenate all m_i for $0 \leq i \leq n$ and between each pair of claims we put a `[SEP]` tag. If the length of the final input exceeds the maximum input length for the model, we iteratively delete m_i, for i from 1 to $n - 2$ (keeping always c_0 at the beginning). We will call this concatenation TXT_CHAIN.

3.3.2 Temporal context

To model the temporal context, we add at the beginning of each claim m_i (from the textual context) the time t_i passed between the publication of the initial turn $\bar{c}_0$ and the publication of $\bar{c}_i$. However, we know that pretrained language models struggle in mathematical reasoning (Patel et al., 2021). To overcome this limitation, instead of reporting t_i as a value in milliseconds (as provided in the dataset) the temporal information is given in the format "`after` d `days,` h `hours,` m `minutes`", with d, h, and m correctly computed. In this way, a model can potentially understand if two comments appear in the same minute, hour or day, giving a glance of time windowing. We will call this prefix TIME, and we make it delimited by two special tags: `<t>` and `</t>`.

3.3.3 Interactional context

To model the interactional context, we add at the beginning of each claim m_i the local user ID of u_j. This piece of information makes it possible to reconstruct the structure of the interaction graph among the users in the discussion d, i.e. if A replies to B, there is a direct edge from A to B. We can therefore see the interaction graph as a multi-edge

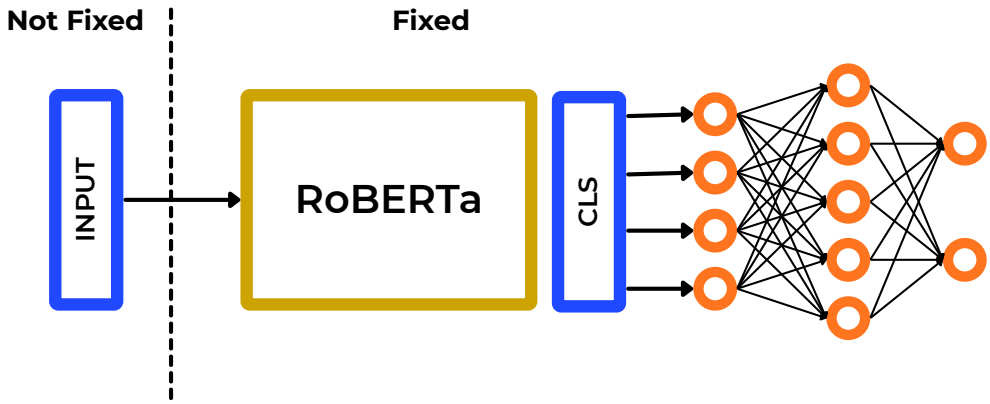

Figure 3.5: schematic view of the model we tested. We distinguish between the component we change in each experiment (the input) and the fixed structure (`RoBERTa` + MLP).

directed graph of the interactions, with the textual content and the order of interactions as labels (Figure 3.3).

The local user ID is *locally unique*: for each multi-party conversation, a value from 0 to $l-1$ is incrementally assigned to each of the l users contributing in the conversation according to their first appearance within the conversation itself. Using local IDs means that when a user is active across different MPCs, they are assigned a different ID in each conversation. This prevents our model from implicitly profiling users' behavior and attitude at global level, thus adopting a privacy-preserving approach.

The interactional information is given in input to the model adding before each comment the prefix "j`th user`", with $0 \leq j \leq l-1$ to declare that the author with local ID j wrote the claim. We will call this prefix USER, and also for this prefix we adopt two special tags to signal the start and the end of the prefix: `<o>` and `</o>`.

3.4 Models and Experimental Settings

3.4.1 Model Architecture

The model architecture is reported schematically in Figure 3.5. It is made of two main components: a `RoBERTa` model with on top a Multi Layer Perceptron (MLP). To perform the prediction, we feed the `RoBERTa` model with the input, and then we extract the final `[CLS]` contextual embedding. So we pass the `[CLS]` contextual embedding to the MLP, which consists in a classic Feedforward Neural Network (FNN), and perform the prediction.

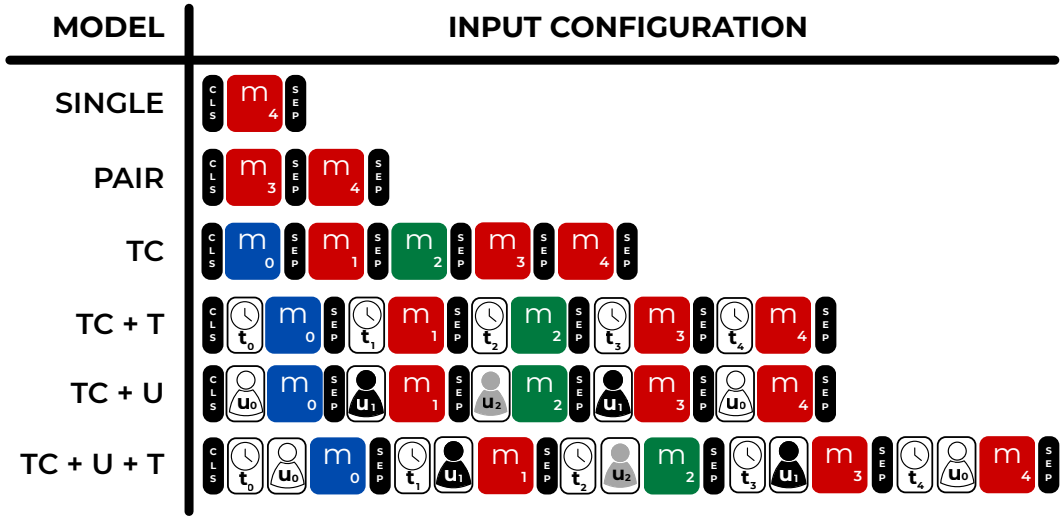

Figure 3.6: schematic view of the input configuration for each model tested, taking as an example the multi-party conversation reported in Figure 3.2. We display the position of each textual content m_i, the `[CLS]` tokens, the `[SEP]` tokens, the USER prefix and the TIME prefix.

The dimension of the `[CLS]` contextual embedding is $d = 768$. The `RoBERTa` model architecture and initial weights correspond to the pretrained version provided by Hugging Face called `roberta-base`[3], with maximum input length $L = 512$ tokens.

The MLP consists in 3 layers: I. the first goes from dimension 768 to 200 with `ReLU` activation function; II. the second goes from dimension 200 to dimension 300, again with `ReLU` activation function; III. the third goes from dimension 300 to dimension n, where n is the number of classes among which we predict the class, with `tanh` activation function. Finally we apply a `softmax` on the n value in output from the last layer, in order to have a probability distribution among the n possible values (the prediction will correspond to the index of highest probability).

3.4.2 Input Configurations

We implement and compare eight different classification models trained on the SDK dataset, which can be divided into three categories: DUMMY, BASELINES and CONTEXTUAL. DUMMY models predict the label ignoring the input (i.e., majority class or random class). Instead, for BASELINES and CONTEXTUAL we always finetune the model presented architecture presented in Section 3.4.1. We use Optuna (Akiba et al., 2019) for hyperparameter optimization of the learning rate and the dropout applied to the

[3] `https://huggingface.co/roberta-base`

Model	**Input**
SINGLE	**<s>** Receiving a benefit while helping others is not morally wrong. Otherwise all the foundations supported by big brands would be morally reprehensible. **</s>**
PAIR	**<s>** Personal "return on investment" should not be a guide on charity. **</s></s>** Receiving a benefit while helping others is not morally wrong. Otherwise all the foundations supported by big brands would be morally reprehensible. **</s>**
TC	**<s>** People should donate to organisations that support gorillas instead of to those that support starving children. **</s></s>** Saving gorillas has less impact on the donor's own well-being than saving a child. **</s></s>** Personal "return on investment" should not be a guide on charity. **</s></s>** Receiving a benefit while helping others is not morally wrong. Otherwise all the foundations supported by big brands would be morally reprehensible. **</s>**
TC + T	**<s><t>** after 0 days, 0 hours, 0 minutes **<t>** People should donate to organisations that support gorillas instead of to those that support starving children. **</s></s><t>** after 0 days, 6 hours, 36 minutes **<t>** Saving gorillas has less impact on the donor's own well-being than saving a child. **</s></s><t>** after 7 days, 20 hours, 23 minutes **<t>** Personal "return on investment" should not be a guide on charity. **</s></s>** <t>after 28 days, 23 hours, 14 minutes **<t>** Receiving a benefit while helping others is not morally wrong. Otherwise all the foundations supported by big brands would be morally reprehensible. **</s>**
TC + U	**<s><o>** 0th user **<o>** People should donate to organisations that support gorillas instead of to those that support starving children. **</s></s><o>** 1st user **<o>** Saving gorillas has less impact on the donor's own well-being than saving a child. **</s></s><o>** 2nd user **<o>** Personal "return on investment" should not be a guide on charity. **</s></s><o>** 1st user **<o>** Receiving a benefit while helping others is not morally wrong. Otherwise all the foundations supported by big brands would be morally reprehensible. **</s>**
TC + U + T	**<s><t>** after 0 days, 0 hours, 0 minutes **<t><o>** 0th user **<o>** People should donate to organisations that support gorillas instead of to those that support starving children. **</s></s><t>** after 0 days, 6 hours, 36 minutes **<t><o>** 1st user **<o>** Saving gorillas has less impact on the donor's own well-being than saving a child. **</s></s><t>** after 7 days, 20 hours, 23 minutes **<t><o>** 2nd user **<o>** Personal "return on investment" should not be a guide on charity. **</s></s><t>** after 28 days, 23 hours, 14 minutes **<t><o>** 1st user **<o>** Receiving a benefit while helping others is not morally wrong. Otherwise all the foundations supported by big brands would be morally reprehensible. **</s>**

Table 3.2: Different types of input related to the same discussion that are fed to the model.

MLP (details in Appendix A.1). In Figure 3.6 we report a schematic view of the input configuration employed for the BASELINE models and the CONTEXTUAL models.

We describe below in details the different classification models, divided into the three following categories.

Dummy. We employed two *dummy* models:

- MAJORITY CLASS: this model always assigns the majority class label (i.e., *support* in the case of the SDK dataset).
- RANDOM: this model assigns the label, for each item, at random, each with the probability $p = 0.5$.

Text-Only Baselines. These two models, based only on the text of the claims, take in input a fixed number of claims:

- SINGLE: we give in input to the model only the textual content of the last claim m_n. The goal is to predict the stance of m_n without considering what was written before. This approach should be able to perform classification just by looking at linguistic or stylistic cues in m_n.
- PAIR: we give in input to the model only the textual content of the last two comments, m_n and m_{n-1}, separated by the `[SEP]` token. The goal here is to predict the correct label looking at the semantics and at the style of the two claims, as well as at the relations between the two. This is the standard solution for Stance Detection.

Contextual. We model contextual information in four different ways:

- TC: we give in input to the model only the concatenated claims in the TXT_CHAIN format.
- TC+T: we give in input to the model the concatenated claims in the TXT_CHAIN format, each claim with the TIME prefix.
- TC+U: we give in input to the model the concatenated claims in the TXT_CHAIN format, each claim with the USER prefix.
- TC+U+T: we give in input to the model the concatenated claims in the TXT_CHAIN format, each claim with the TIME prefix and the USER prefix.

We report in Table 3.2 an example of how the same discussion is given in input to the model in the different configurations. In the pretrained `RoBERTa` model available

Category	Model	C-F1	S-F1	W-F1	M-F1
Dummy	Majority	70.5 (±0.0)	0.0 (±0.0)	38.4 (±0.0)	35.3 (±0.0)
	Random	52.1 (±0.6)	48.0 (±0.4)	50.2 (±0.5)	50.1 (±0.5)
Baseline	Single	75.5 (±0.5)	70.2 (±0.6)	73.0 (±0.1)	72.8 (±0.2)
	Pair	83.1 (±0.4)	79.3 (±0.4)	81.4 (±0.2)	81.2 (±0.2)
Contextual	TC	82.2 (±0.6)	78.8 (±0.4)	80.7 (±0.3)	80.5 (±0.3)
	TC+T	83.3 (±0.4)	80.0 (±0.4)	81.8 (±0.3)	81.7 (±0.3)*
	TC+U	85.2 (±0.5)	82.1 (±0.7)	83.8 (±0.5)	83.7 (±0.5)*◇
	TC+U+T	85.6 (±0.4)	82.3 (±0.3)	84.0 (±0.3)	83.9 (±0.3)*◇

Table 3.3: F1 scores obtained on the test set of SDK dataset, for each class, in weighted average and in macro average (average of the best 5 runs in validation over 10). (∗) and (◇) show a statistically significant improvement with respect to the Pair baseline, for ASO test and Student's t-test respectively. We report the average and the standard deviation for each metric.

on Hugging Face[4], the `[CLS]` token is replaced by a `<s>` tag and the `[SEP]` token is represented by a sequence of special tags (i.e., `</s></s>`). We have taken inspiration from these representations for our new special tokens: `<t>`, `</t>`, `<o>`, `</o>`. The input text is pre-processed by replacing user mentions and urls with placeholders following a standard approach for social media data.[5]

3.5 Experiments on SDK dataset

3.5.1 Stance Detection results

The goal of the first set of experiments is to evaluate on Kialo the performance of the eight models described above by using the whole training set, both for hyperparameter optimization and for the final evaluation. The results are the average and standard deviation over 5 experimental runs. We report in Table 3.3 the F1 score for each class, its weighted average (W-F1), and the macro average (M-F1). The final metric we use for ranking the models is M-F1.

Results. All the results are reported in Table 3.3. We compute statistical significance using Almost Stochastic Order test (Del Barrio et al., 2018; Dror et al., 2019) and Student's t-test for independent sample with Bonferroni correction (Bonferroni, 1936). For ASO, we use the implementation provided in the `deep-significance` library, presented by Ulmer et al. (2022), with the suggested threshold value of $\tau = 0.2$. For the t-test we use the implementation provided in the `scipy` library with threshold value of $\alpha = 0.05$.

[4] `https://huggingface.co/docs/transformers/model_doc/roberta`
[5] `https://huggingface.co/cardiffnlp/twitter-roberta-base-sentiment`

Both BASELINE models lead to better performances than the DUMMY models. Interestingly, the SINGLE model performs well (72.8 M-F1 on average), showing that the style of the target comment already conveys relevant information to detect the correct stance. However, as expected, taking the last two comments in input (PAIR model) increases the M-F1 score by +8.4 over the SINGLE one.

Among the CONTEXTUAL models, the TC model achieves the worst results, slightly lower than the PAIR model. This shows that adding context is not always beneficial. In this case, since the number of claims in a conversation changes, the model is probably not able to focus on the right portion of the chain. Adding the temporal information only, as in the TC + T model, yields a better performance than the simple textual chain in the TC model (+1.2 M-F1) and outperforms significantly the PAIR baseline (+0.5) for the ASO test.

Looking at the different types of context, we observe that adding only the USER prefix as in TC + U, leads to a significant increase of +3.2 M-F1 over the TC model and of +2.5 over the PAIR baseline, for both statistical significance tests. Furthermore, the TC + U + T model with both USER prefix and TIME prefix increases significantly the performance with respect to TC model (+3.4), PAIR model (+2.7) and TC + T model (+2.2), again for both statistical significance tests. However, there is no significant difference between TC + U model and TC + U + T model (only +0.2). This indicates that TIME prefix is no more relevant once we pass to the model the USER prefix.

3.5.2 Learning Curve Analysis

While our experiments show that the discussion context on the SDK dataset is beneficial to stance detection, we aim to assess the impact of the training set size. Our intuition is that, when contextual information is embedded in the model, more training instances are needed than for non-contextual models. Indeed, the model must be given enough training instances to understand what is the role of the special tags and what type of information is included between two specific separators.

We therefore extract from the original training data 5 different training sets, comprising around 5% (6,354 examples), 10% (12,402 examples), 20% (24,748 examples), 40% (49,249 examples) and 80% (98,389 examples) of the original training instances.

We report in Table 3.4 the detailed Learning Curve results on the SDK dataset. We first run hyperparameter optimization on each training set. Then, after fixing the hyperparameters as in Section 3.4, we perform 3 experimental runs on each training set, changing the random seed each time, and compute the average M-F1 among the 3 runs. The same evaluation is performed using the complete training set.

Figure 3.7 shows the results obtained when increasing the training set size as the aver-

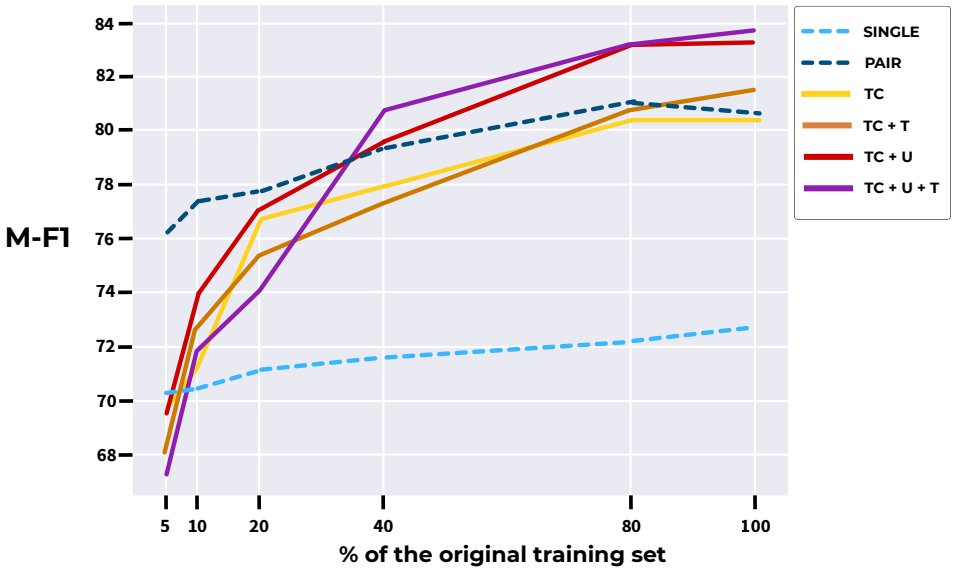

Figure 3.7: Learning curve for each BASELINE and CONTEXTUAL model, in terms of M-F1 score.

age over 3 runs (Table 3.4). We exclude the DUMMY models, since they never outperform BASELINE and CONTEXTUAL models.

With 5% of the training data, all the CONTEXTUAL models are beaten by the worst BASELINE model (i.e., SINGLE), with a drop in M-F1 ranging from −10.8 (TC) to −16.3 (TC+U+T) compared to using the whole training set. At the same time, the PAIR model achieves the best result in this setting, with a performance drop of only −4.6. However, as soon as we add more data, the scenario changes. With 10% training set and 20% training set, CONTEXTUAL models overcome the SINGLE model and progressively approach the PAIR model. With 40% training set, TC + U and TC + U + T outperform the PAIR model and with more data they substantially increase their gap with the latter.

To sum up, these results show that CONTEXTUAL models need between 20% and 40% of the training data (i.e., from 24 thousand to 49 thousand training examples) to achieve comparable results with the PAIR model, while they need more data to outperform it.

3.5.3 Analysis of truncation effects

In Section 3.3, we discuss the processing of strings that exceed the maximum input length by employing a deterministic truncation process on the discussion chains until the length satisfies the model constraint. We conduct an additional evaluation to investigate whether such truncation correlates with the final results, implying potential effects on performance (moreover, we report in Figure 3.8 the plot of the original lengths of the discussion chains in terms of number of claims).

Category	Model	5%	10%	20%	40%	80%	100%
Dummy	Majority	35.3	35.3	35.3	35.3	35.3	35.3
	Random	50.1	50.1	50.1	50.1	50.1	50.1
Baselines	Single	70.2	70.5	71.1	71.6	72.2	72.7
	Pair	76.1	77.3	77.7	79.3	80.9	80.7
Contextual	TC	69.6	71.1	76.7	77.9	80.4	80.4
	TC+T	68.2	72.7	75.4	77.4	80.7	81.6
	TC+U	69.6	73.8	77.1	79.4	83.2	83.3
	TC+U+T	67.4	71.8	74.1	80.7	83.2	83.7
Training Set Size		6354	12402	24748	49249	98389	122681

Table 3.4: Macro-F1 scores obtained on the test set of SDK dataset during the learning curve analysis, for every training set in growing size.

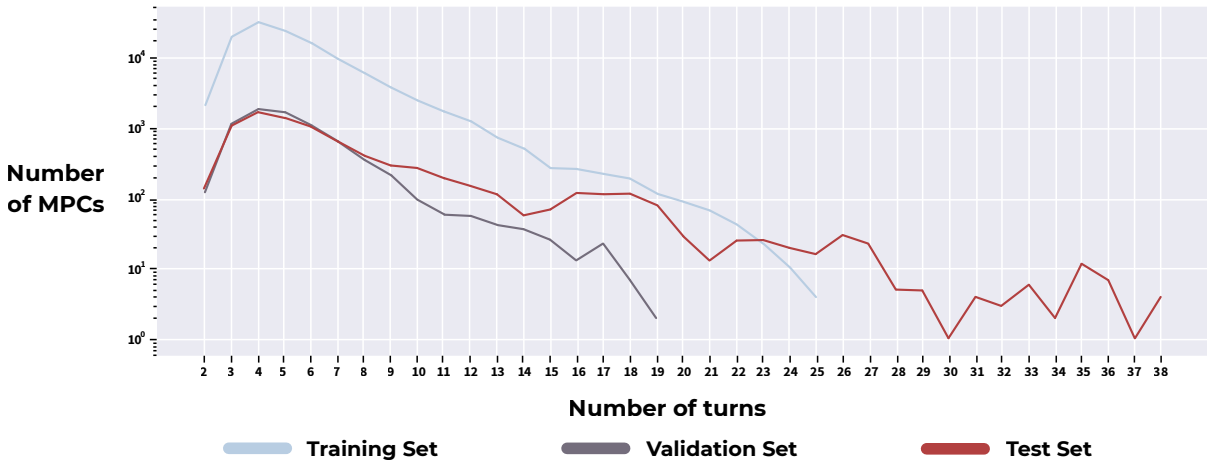

Figure 3.8: Length distribution of the multi-party conversations (i.e. number of turns in the conversation) in SDK dataset.

For each contextual input configuration and dataset split, we compute the following metrics: (i.) Truncation rate (ratio of truncated sequences); (ii.) average truncation (number of truncated claims); (iii.) average original length of the truncated sequences.

The statistics in Table 3.5 reveal that the TC + T and TC + U + T input configurations result in more truncated chains, while TC + U exhibits less truncation than TC + T. Nevertheless, TC + U and TC + U + T configurations perform similarly and both outperform the TC + T model. This analysis suggests that the impact of the truncation process does not significantly influence our findings.

TC	Train	Valid.	Test
Truncation Rate	1.01%	0.70%	8.30%
Avg Truncation	4.20	4.29	6.27
Avg Original	16.68	13.63	19.98

TC+T	Train	Valid.	Test
Truncation Rate	3.60%	3.60%	13.68%
Avg Truncation	4.05	3.30	7.38
Avg Original	13.48	12.68	17.06

TC+U	Train	Valid.	Test
Truncation Rate	2.33%	1.92%	11.40%
Avg Truncation	4.14	3.68	7.20
Avg Original	14.65	13.89	18.19

TC+U+T	Train	Valid.	Test
Truncation Rate	6.70%	6.03%	18.40%
Avg Truncation	3.74	3.44	7.00
Avg Original	11.95	11.54	15.31

Table 3.5: Statistics of the truncation process in the SDK dataset, with a separate table dedicated to each model and a column corresponding to each dataset split.

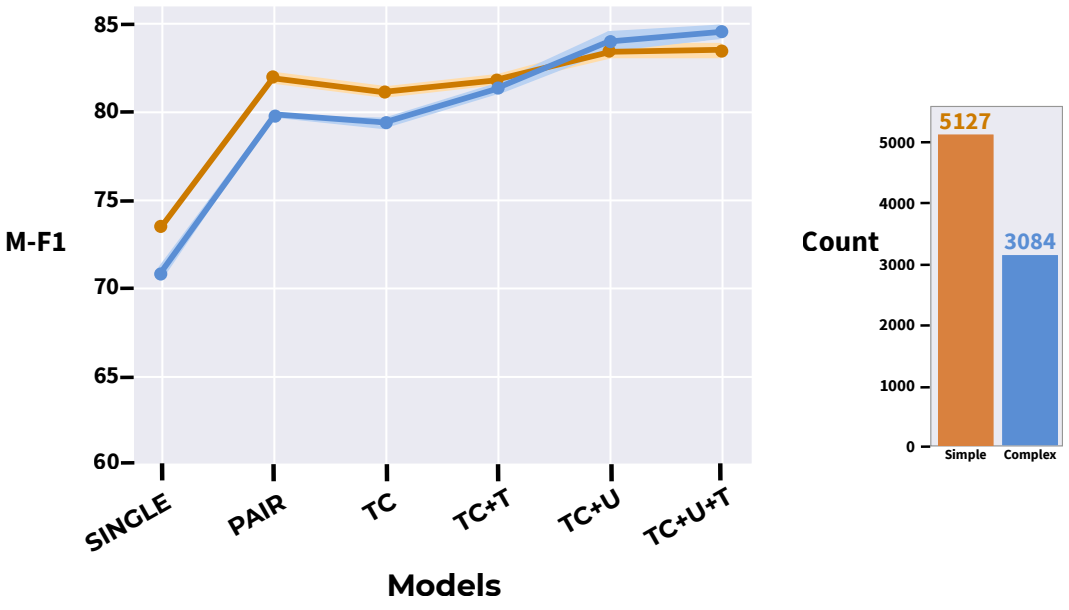

Figure 3.9: Model comparison when testing the classifier on different dimensions: Simple MPCs vs. Complex MPCs.

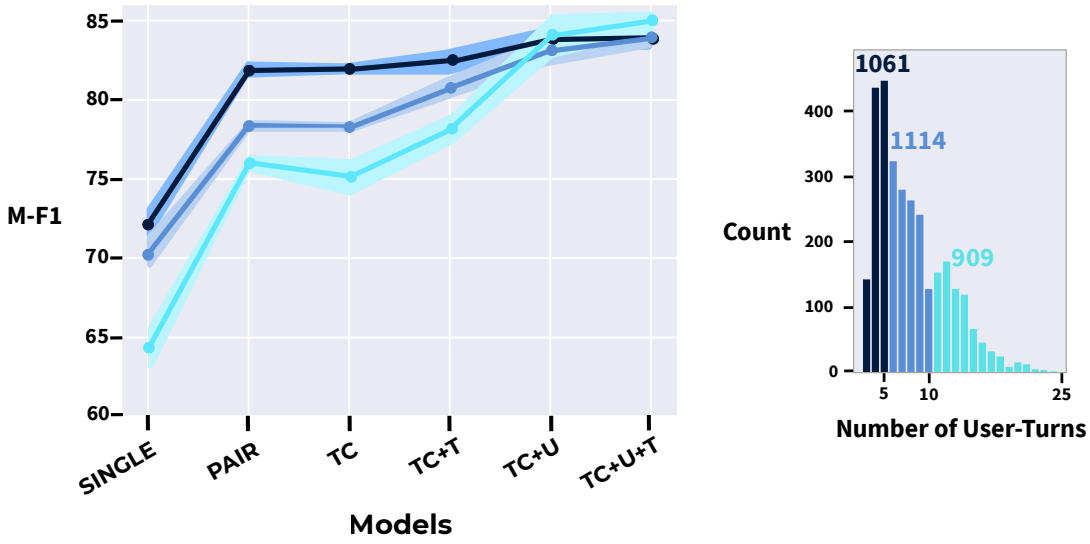

Figure 3.10: Model comparison when testing the classifier on different dimensions: Complex MPCs with different number of user-turns.

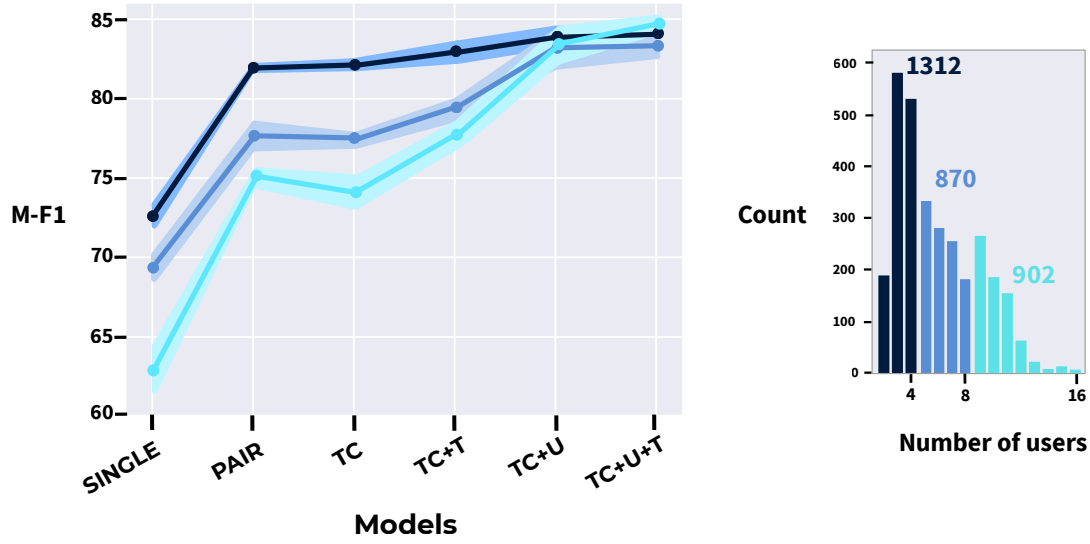

Figure 3.11: Model comparison when testing the classifier on different dimensions: Complex MPCs with different number of users.

3.5.4 Analysis of Conversational Structure

Beside assessing the impact of training set size on classification performance, we are also interested in analyzing the role played by the topology of the underlying interaction graph in terms of complexity and conversation length.

In Kialo, the same author can write several consecutive comments, even in contrast

with each other. However, we are more interested in interactions among different users. For this reason, we introduce the concept of *user-turn*. Given a discussion chain of n turns, we can retrieve a chain of n' user-turns, where two consecutive user-turns have different authors. This is possible by merging all consecutive turns written by the same user into a unique user-turn. For instance if we have a thread d of length 6 claims with user sequence $\{u_0, u_0, u_1, u_1, u_1, u_2\}$, the associated user-turn thread has length 3 merging into one turn the first two claims, then the following three into another turn and the last one is already a turn, with user sequence $\{u_0, u_1, u_2\}$.

We first divide multi-party conversations in the SDK dataset into two groups: simple MPCs, which are characterized by conversations where users contribute with only one user-turn, and complex MPCs, with a user writing several user-turns. A complex MPC might be similar to the following: if the user sequence is $\{u_0, u_1, u_0, u_0, u_2, u_2\}$, in the user-turn thread the user sequence becomes $\{u_0, u_1, u_0, u_2\}$. We run the stance detection experiment with the setting presented in Section 3.5 and compare the results obtained on simple vs. complex chains. We also analyze how the number of user-turns and number of users affects classifier performance on complex MPCs (with and without context). Results are reported in Figure 3.9, which displays the M-F1 score obtained with the different models. The thickness of the line represents the standard deviation over 5 runs.

The analysis shows that extra-linguistic context gives an important contribution to the classification of complex MPCs, in particular the TC+U+T model. This contribution is more limited on simple MPCs, with the Pair model and the Contextual models achieving comparable results.

As regards the impact that the number of user-turns has on the classification of complex MPCs (Figure 3.10), we first group the user-turns into three bins based on their length: from 2 to 5 (dark blue), from 6 to 10 (blue) and >10 (light blue). The comparison among the three groups clearly demonstrates that the inclusion of temporal and interactional context consistently results in a performance improvement, regardless of the number of user-turns in the conversation. We finally investigate the effect that the number of users involved in the complex MPC has on classification performance (Figure 3.11). Also in this case, the chains are grouped into three bins: having less that 4 users (dark blue), from 5 to 8 users (blue), and more than 8 (light blue). Again, the comparison demonstrates that the inclusion of the extra-linguistic contexts consistently results in improvement, regardless of the number of users involved in the discussion.

3.6 Experiments on other Datasets

As a comparison, we run the same experiments on two smaller datasets, which provide the same type of information included in SDK: the SQDC dataset Gorrell et al. (2019) for stance detection (Section 3.6.1), and the ContextAbuse dataset Menini et al. (2021) for abusive language detection (Section 3.6.2). These datasets present a size of respectively 5% and 7% compared to SDK. On the SQDC dataset, the SINGLE baseline yields the best result (47.2 M-F1), probably because the official test set contains only chains of length 2. After creating a better balanced train and test split, instead, the best result is obtained with the PAIR baseline (46.4 M-F1). On the ContextAbuse dataset, adding textual context (i.e., TC model) yields the best performance (81.4 M-F1), which however is not statistically significant compared to the SINGLE baseline (80.7 M-F1).

These experiments suggest that, independently from the specific task, contextual information may not yield substantial enhancements in performance if the amount of training data is too limited. We go more in depth with the results of each dataset in following subsections.

Category	Model	S-F1	Q-F1	D-F1	C-F1	W-F1	M-F1
DUMMY	MAJ.	0.0 (±0.0)	0.0 (±0.0)	0.0 (±0.0)	86.6 (±0.0)	66.1 (±0.0)	21.6 (±0.0)
	RAND.	12.6 (±2.1)	9.5 (±1.1)	13.9 (±2.6)	37.5 (±1.9)	31.6 (±1.3)	18.3 (±0.6)
BASELINE	SINGLE	14.1 (±7.7)	54.4 (±2.9)	47.5 (±3.5)	72.6 (±5.7)	64.1 (±4.2)	47.2 (±2.3)
	PAIR	13.5 (±1.6)	58.4 (±3)	44.9 (±0.1.5)	71.1 (±3.2)	62.8 (±2.3)	47.0 (±0.5)
CONTEXTUAL	TC	12.9 (±4.1)	58.6 (±2.4)	42.7 (±7.2)	71.5 (±4.3)	62.9 (±4.2)	46.4 (±4.0)
	TC+T	15.4 (±0.8)	59.0 (±2.6)	44.1 (±4.5)	63.4 (±3.7)	57.0 (±2.8)	45.5 (±1.6)
	TC+U	13.2 (±5.1)	56.3 (±4.6)	41.6 (±3.8)	65.1 (±11.4)	57.8 (±8.6)	44.0 (±3.3)
	TC+U+T	19.2 (±4.7)	52.3 (±3.5)	43.1 (±1.8)	68.6 (±4.7)	61.1 (±3.9)	45.8 (±2.3)

Table 3.6: *SQDC - Challenge.* F1 scores obtained on the test set of SQDC dataset, on the original split given for the challenge. The F1 score is reported for each class, in weighted average and in macro average. The results are the average over the best 5 runs in validation over 10. We report the average and the standard deviation for each metric.

Category	Model	S-F1	Q-F1	D-F1	C-F1	W-F1	M-F1
DUMMY	MAJ.	0.0 (±0.0)	0.0 (±0.0)	0.0 (±0.0)	82.9 (±0.0)	58.6 (±0.0)	20.7 (±0.0)
	RAND.	15.3 (±2.0)	14.4 (±3.4)	11.7 (±2.0)	39.1 (±1.5)	31.8 (±0.7)	20.1 (±0.8)
BASELINE	SINGLE	31.3 (±3.7)	52.5 (±2.6)	27.7 (±5.7)	56.2 (±6.6)	51.0 (±5.3)	42.0 (±3.4)
	PAIR	30.2 (±1.5)	54.3 (±1.6)	33.7 (±1.4)	67.2 (±3.7)	59.3 (±2.9)	46.4 (±1.8)
CONTEXTUAL	TC	28.3 (±3.0)	53.1 (±4.7)	31.1 (±4.2)	68.4 (±5.3)	59.6 (±3.9)	45.3 (±2.3)
	TC+T	27.9 (±1.8)	49.8 (±1.9)	33.6 (±2.9)	63.3 (±4.5)	55.8 (±3.3)	43.6 (±2.0)
	TC+U	27.9 (±1.1)	52.7 (±2.2)	32.2 (±3.0)	64.8 (±4.0)	57.0 (±3.0)	44.4 (±1.5)
	TC+U+T	27.2 (±2.1)	51.4 (±3.0)	32.8 (±1.4)	62.2 (±3.2)	55.0 (±2.8)	43.4 (±2.0)

Table 3.7: *SQDC - New split.* F1 scores obtained on the test set of SQDC dataset, with our new split to obtain complex structures even in training. See caption in Table 3.6 for further details.

Category	Model	NS-F1	S-F1	W-F1	M-F1
DUMMY	MAJ.	82.9 (±0.0)	0.0 (±0.0)	58.6 (±0.0)	41.4 (±0.0)
	RAND.	59.6 (±1.0)	38.4 (±1.5)	53.4 (±0.8)	49.0 (±0.9)
BASELINE	SINGLE	74.4 (±2.9)	52.9 (±0.8)	68.1 (±2.3)	63.6 (±1.8)
	PAIR	73.4 (±3.4)	53.8 (±1.5)	67.7 (±2.6)	63.6 (±2.0)
CONTEXTUAL	TC	73.3 (±3.2)	49.3 (±1.3)	66.3 (±2.5)	61.3 (±2.1)
	TC + T	75.3 (±3.0)	51.1 (±1.4)	68.3 (±2.4)	63.2 (±2.0)
	TC + U	74.7 (±3.0)	49.9 (±1.0)	67.5 (±2.1)	62.3 (±1.6)
	TC + U + T	74.7 (±1.5)	48.4 (±1.9)	67.0 (±1.3)	61.6 (±1.3)

Table 3.8: *SQDC - Binary.* F1 scores obtained on the test set of SQDC dataset, with our new split to obtain complex structures even in training, for the binary task to detect Stance class vs No Stance Class. See caption in Table 3.6 for further details.

3.6.1 Results on SQDC dataset

We perform the same set of experiments as in Section 3.5.1 on a second dataset, which was developed for the task "SQDC support classification" at the RumourEval 2019 challenge (Gorrell et al., 2019). For each item we have the same information as in the SDK dataset, and given a discussion tree, all the discussion chains from the initial claim to any node (even internal) are extracted, and each item labeled according to the last comment. However, the label of each claim does not represent the stance versus the previous claim, but rather the stance with respect to the rumour discussed in the chain. This chain is treated as the common ground topic on which the discussion is taking place, even if it is not necessarily stated explicitly in the initial claim. Again, the dataset split is based on the initial claim, avoiding any data contamination.

There are four possible labels: I. *support*, II. *query*, III. *deny*, and IV. *comment*. Those labels are respectively shortened as S, Q, D and C, from which the name of the task (SQDC support classification). The original dataset is highly unbalanced among the classes and comprises threads from Reddit[6] and Twitter[7]. We focus this second set of experiments on the Twitter part of the dataset.

At first, we run our experiments on the original train-validation-test split, reaching different results w.r.t. those obtained on Kialo, since the SINGLE model yields the best performance (see full results in Table 3.6).

We further inspect the dataset and we find that the test set was formed only by chains of length 2, where the usefulness of the context is limited. So, we exclude the original test set and generate a new train-validation-test split, analyzing the distribution of labels and chain lengths. The results are different w.r.t. the original SQDC dataset: the CONTEXTUAL model achieves a performance between SINGLE model (lower bound) and PAIR model (upper bound). For details, see Table 3.7. Overall, the results on the new split of the SQDC dataset confirm the overall findings obtained by analyzing the learning

[6]https://www.reddit.com
[7]https://twitter.com

	SQDC Dataset - Challenge				
Set	**S**	**Q**	**D**	**C**	**Total**
Train	20.2%	7.9%	7.6%	64.3%	4519
Valid.	9.0%	10.1%	6.8%	74.1%	1049
Test	13.2%	5.8%	8.6%	72.4%	1066
	SQDC Dataset - New split				
Set	**S**	**Q**	**D**	**C**	**Total**
Train	13.9%	8.6%	7.6%	69.9%	3957
Valid.	12.0%	8.9%	8.7%	70.4%	689
Test	11.3%	10.9%	7.1%	70.7%	595
	SQDC Dataset - Binary				
Set	**No Stance**		**Stance**		**Total**
Train	69.9%		30.1%		3957
Valid.	70.4%		29.6%		689
Test	70.7%		29.3%		595

Table 3.9: Distribution of the labels in SQDC dataset, distinguishing training set, validation set, and test set We report the three versions experiments: chellenge version, new split version and binary version.

curve for different training sizes in Kialo (discussed in Section 3.5.2): the SQDC dataset is not large enough to allow the model to learn how to exploit the context in an effective way. We also try to test our models on a binary task, more similar to stance detection in Kialo, by merging the *query* class, the *deny* class and the *support* class into a unique stance class, and the comment class as a no-stance class. Results are reported in Table 3.8. Again, the SINGLE model is the best performing one probably due to the data size and the context does not yield any improvement.

For these datasets, we report the descriptive statistics in Table 3.9 and plot the length distribution of the discussion chains in Figure 3.12 and Figure 3.13.

To balance the classes during training, for each epoch we undersample each class in the training set in order to have s samples for each class, where s is the cardinality of the less represented class. We use as loss function the unweighted Cross Entropy. Then, for validation, we use a weighted Cross Entropy Loss according to the cardinality of each class, with weight $w_c = 100/s_c$ for each class, where s_c is the cardinality of the class c. We use the same pipeline for hyperparameter optimization and test on fixed hyperparameters as in SDK dataset (i.e. 5 best runs in validation over 10), performing even the same statistical test. Further details are reported in Appendix A.1.

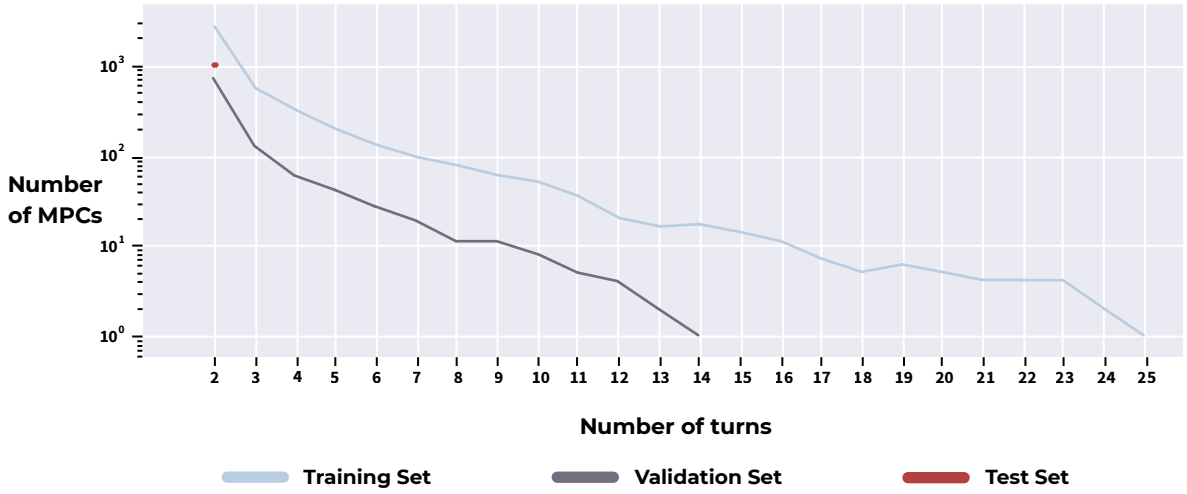

Figure 3.12: Length distribution of the multi-party conversations (i.e. number of turns in the conversation) in SQDC dataset - challenge version.

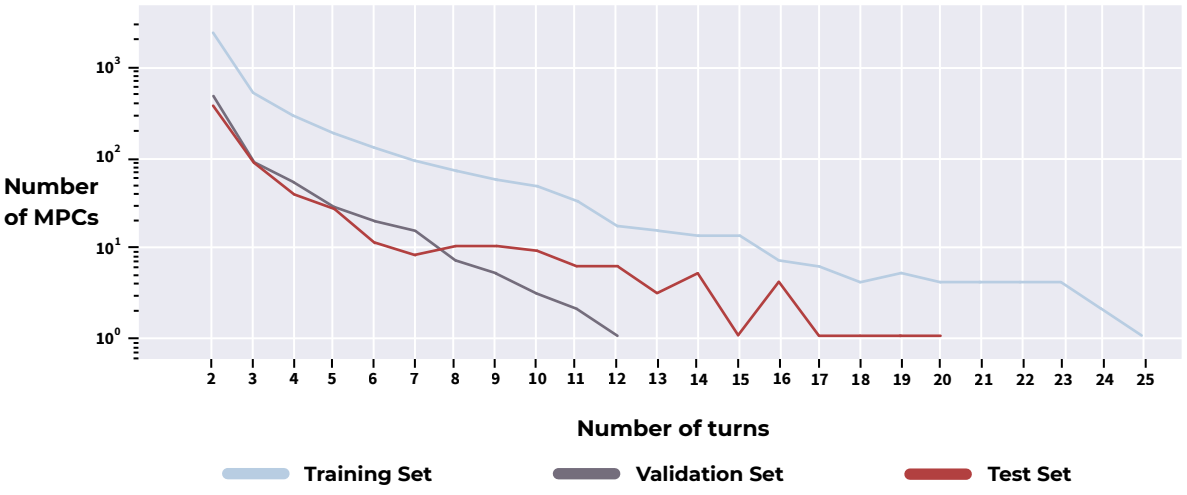

Figure 3.13: Length distribution of the multi-party conversations (i.e. number of turns in the conversation) in SQDC dataset - new split version.

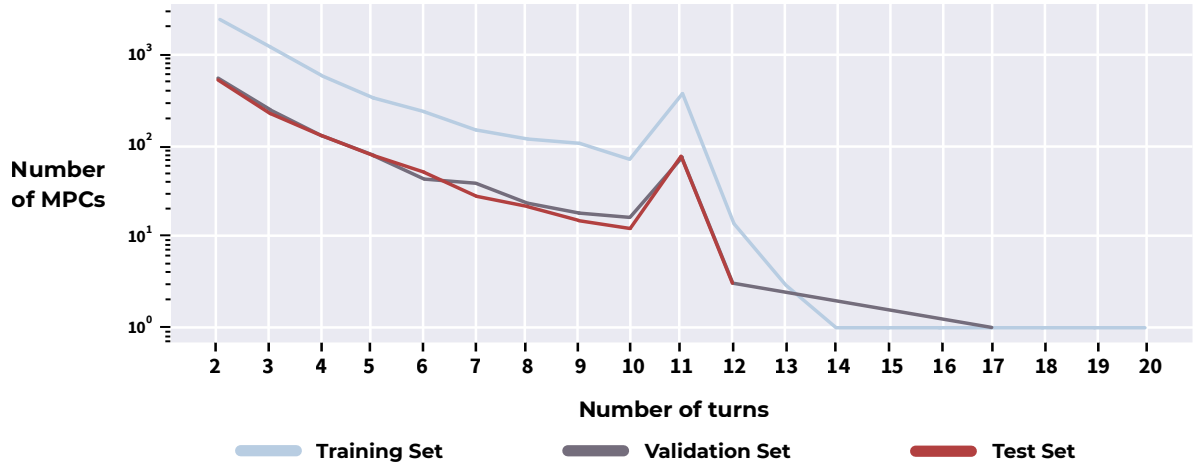

Figure 3.14: Length distribution of the multi-party conversations (i.e. number of turns in the conversation) in ContextAbuse dataset.

Category	Model	A-F1	NA-F1	W-F1	M-F1
Dummy	Majority	89.9(±0.0)	0.0(±0.0)	73.4(±0.0)	45.0(±0.0)
	Random	82.2(±0.4)	21.1(±2.7)	71.0(±0.7)	51.7(±1.4)
Baselines	Single	91.0(±0.4)	70.5(±0.8)	87.2(±0.5)	80.7(±0.6)
Contextual	TC	91.4(±1.2)	71.4(±2.2)	87.7(±1.3)	81.4(±1.7)
	TC + T	90.6(±1.3)	69.6(±2.1)	86.7(±1.5)	80.1(±1.7)
	TC + U	90.1(±1.8)	68.7(±2.8)	86.2(±2.0)	79.4(±2.3)
	TC + U + T	91.6(±0.8)	70.8(±1.0)	87.8(±0.8)	81.2(±0.9)

Table 3.10: *ContextAbuse.* F1 scores obtained on the test set of ContextAbuse dataset. The F1 score is reported for each class, in weighted average and in macro average. The results are the average over the best 5 runs in validation over 10. We report the average and the standard deviation for each metric.

	ContextAbuse Dataset		
Set	**No Abuse**	**Abuse**	**Total**
Training	82.6%	17.4%	5651
Validation	82.4%	17.6%	1216
Test	81.7%	18.3%	1151

Table 3.11: Label distribution in the ContextAbuse dataset

3.6.2 Results on ContextAbuse dataset

ContextAbuse (Menini et al., 2021) is a subset of the well-known hate speech dataset by Founta et al. (2018), where the items have been relabeled as "Abusive" or "Not Abusive" taking into account not only the tweet to classify, but also the previous tweets (textual

context). This re-annotation led to a remarkable reduction of items annotated as "Abusive", suggesting that context is vital to disambiguate real abusive tweets from other cases (e.g. irony, satire, etc.). Given the set of tweets from Founta et al. (2018), the authors did not retrieve the full discussion tree, but just the discussion chain from the initial claim to the target comment. In this way, there is no overlap among different items, but each tweet in each sequence is seen only once. This could result in major difficulties for contextual models to extract useful information to perform the classification.

The dataset is provided on Github[8] without official splits. So we create a training/validation/test set according to a 70/15/15 strategy. We report the descriptive statistics in Table 3.11 and the length of the discussion chain in Figure 3.14. In this case we have only the SINGLE model as a baseline because the goal is to classify a single claim.

The results obtained on the ContextAbuse dataset exhibit similarities to the ones obtained from SQDC dataset (new split version). These findings align with the outcomes of the learning curve experiment from the SDK dataset. In this scenario, the contextual models fail to significantly outperform the baseline (which is the SINGLE model in this case). Nevertheless, it is worth noting that the TC model and TC+T+U model exhibit some improvement, albeit not statistically significant, with the latter showing lower variance. However, it remains uncertain whether, in presence of a larger training set, the contextual model would be capable of increasing the performance gap with the baseline. All the results are reported in Table 3.10.

Differently from the SQDC dataset, for each epoch we use the entire training set without undersampling, and make use of weighted cross-entropy loss both for training loss and validation loss, according to the cardinality of each class (as in Appendix 3.6.1). We use the same pipeline for hyperparameter optimization and test on fixed hyperparameters as in SDK dataset (i.e. 5 best runs in validation over 10), performing the same statistical test. Again, further details are reported in Appendix A.1.

3.7 Discussion

The results reported in Section 3.5 show that adding extra-linguistic context is beneficial to improve performance on stance detection. However, this benefit arises only if the CONTEXTUAL models have access to enough data, which in our experiments on the SDK dataset means between $24,000$ and $49,000$ items. This result explains also the different performance obtained on smaller datasets (Section 3.6).

As regards the interactional analysis of multi-party conversation, the more complex is the conversations, the more evident are the benefits from the interactional context.

[8] `https://github.com/dhfbk/twitter-abusive-context-dataset/tree/main`

This suggests that our transformer-based model is able to capture the structure given by the interactions among the users, even if implicit, when enough data are available. Our analyses show also that capturing contextual information is particularly beneficial with longer chains of turns, and discussion chains with more users. When all contextual information (both linguistic and extra-linguistic) is included in the model, the classifier performs equally well on long and on short chains, making the results more consistent and the model more robust to chain length and user activity.

As regards the temporal context, we show that it is still useful to achieve a better performance, but we argue that in Kialo it may not be particularly relevant because this is a platform where users are more likely to ponder their responses and take some time to reflect before posting, also thanks to a strict moderation policy (Vosoughi et al., 2018).

Still, the work presented in this chapter shows some important limitations. The findings presented in this work were mainly focused on the Kialo dataset on the specific task of stance detection. Kialo is an ideal testbed for our hypotheses because it is a moderated platform with well-structured discussions written in plain English. It is not possible to infer that the same findings would be confirmed on any social network, where discussions may be more fragmented and lacking moderation. Indeed, to have a clear picture of our findings, other large datasets with similar characteristics would be needed. Nevertheless, as a preliminary exploration, our experiments on the two smaller datasets from Twitter/X confirmed our expectation about the importance of the amount of training data. Moreover, our work presents a limited number of classification models. We tested a few other combinations without reaching interesting results, therefore we decided to focus only on few configurations and to analyze their behavior more thoroughly. Overall, our contribution is not focused on generally achieving the best results, but rather on assessing how and why contextual information influences the behavior of a model.

Finally, it is worth noticing that integrating user information into a text classification task may pose ethical risks, since profiling may affect classification fairness, hurting some individuals with a specific profile, and is explicitly prohibited in a number of countries. For this reason, we adopt a solution that minimises such risks in that it does not use global user information but only local one, making it impossible to infer user information at platform level. Furthermore, no additional information about users' preferences and attitude is explicitly coded: the model is given in input only *what* and *when* users post in each discussion, and in response *to whom*.

In terms of reproducibility, our models are extremely lightweight and allow the reproduction of the experiments on common GPUs, using implementations available online.

3.8 Main Findings on Modeling the Conversational Context

In this chapter we have tested the effectiveness of using linguistic and extra-linguistic contexts for text classification. Our results show that full linguistic context alone worsens or does not significantly improve the results with respect to the non-contextual baseline. Instead, with extra-linguistic context, the performance improves, especially with the contribution of interactional context. Further analysis shows that such results strongly depend on the amount of data on which the models are trained. Moreover, we found that extra-linguistic context makes results more robust across conversational interactions of different lengths and more or less active users. Our experiments show also that transformer-based models are able to embed interactional features, which can be effectively given in input to the model in the form of simple natural language statements.

Throughout these experiments, we also observed a substantial gap in the literature concerning the evaluation of MPC models, specifically, the tendency to deal with all items equally, regardless of their structural complexity. Motivated by this limitation, in the next chapter we address this issue by introducing a diagnostic evaluation pipeline designed to more fairly assess model performance on multi-party conversation tasks, this time with a focus on component-level modeling.

Chapter 4

Modeling the conversational components

In the previous chapter, we demonstrated that MPC classification systems exploiting the conversational context require more training data compared to non-contextual models, which tend to reach performance saturation earlier. Moreover, we showed that macro-level evaluations can obscure the specific scenarios in which contextual information is most beneficial. For instance, context is particularly useful in conversations characterized by more complex interaction structures, and they help to maintain stable performance across conversations that vary in length and number of participants. Building on these findings, the next step of this thesis focuses on the evaluation perspective, conducting a more fine-grained analysis of MPC system performance across different interactional situations. Through this analysis, we aim to discuss the limitations in current evaluation methodologies, emphasizing the need for more structurally aware evaluation frameworks in MPC processing tasks.

Specifically, in this chapter, we investigate the ability of an LLM to model the conversational components of a multi-party conversation by performing two classification tasks in a zero-shot setting. We address the tasks of *Response Selection* and *Addressee Recognition*, employing `Llama2-13b-chat` (Touvron et al., 2023) not only to identify the last turn of a multi-party conversation but also to summarize the preceding conversation and generate user descriptions, which are subsequently incorporated into the prompts for zero-shot classification.

Our choice of these two tasks is motivated by the following key considerations:

1. the tasks deal with two specific aspects that a model working on MPCs needs to address, i.e. response selection for linguistic aspects and addressee recognition for interactional and non-linguistic aspects;

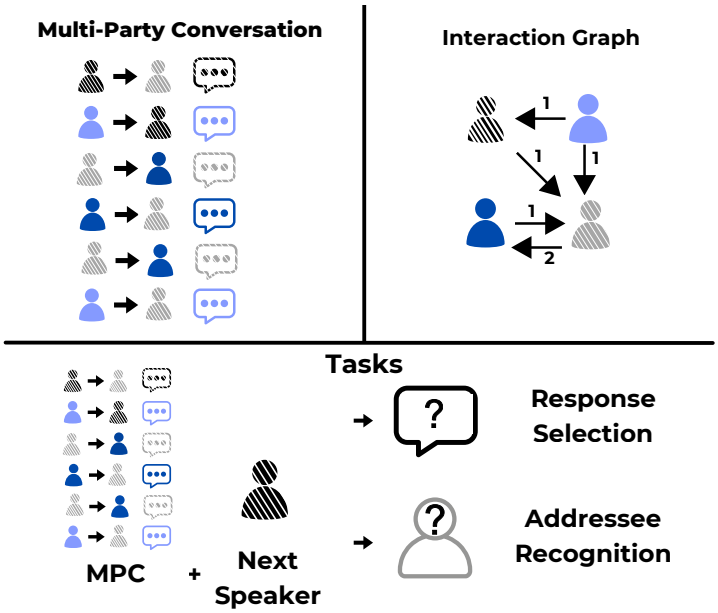

Figure 4.1: a graphical representation of the experiments. Each turn in a conversation includes a speaker, an addressee and a textual message. From the conversation, we extract the interaction graph to diagnose model capabilities by performing two tasks: addressee recognition and response selection.

II. both tasks can be potentially performed on any conversational corpus and across different domains, without requiring additional manual annotation.

These properties make our framework widely applicable. Unlike the experiments presented in the previous chapter, in the current one, a message may be addressed to any speaker within the conversation, not necessarily to the immediately preceding one. This represents an important advancement, as it allows for the presence of "subconversations" occurring in parallel within a single multi-party conversation. Nonetheless, in this chapter we restrict our analysis to one-to-one addressee relations. In the following chapter, we extend the discussion to one-to-many addressee configurations. A graphical overview of the experimental framework of this chapter, illustrating the MPC structure, the interaction graph, and the classification tasks addressed, is provided in Figure 4.1.

Understanding the effects of conversation summarization and user descriptions is important because they could make processing more efficient, replacing multiple turns with a more concise text representation. Furthermore, using summarizations and user descriptions instead of the original conversations would make data sharing easier and more privacy-preserving, addressing growing concerns about this issue (Kim et al., 2023). For instance, it would comply with data minimisation principles, as required by the European

General Data Protection Regulation. Replacing original conversations with summaries and user descriptions would also make it nearly impossible to train generative models that imitate specific users (Huang et al., 2022; Lu et al., 2023).

In this chapter we break down the *Research Question 2* presented in Section 1.1, which can be further defined in three sub-questions:

RQ(2.1): How do LLMs perform in classification tasks involving multi-party conversations in a zero-shot setting, using different input combinations to capture textual and interactional information?

RQ(2.2): What is the model sensitivity to different prompt formulations when classifying MPCs?

RQ(2.3): How does structural complexity of the conversational interactions affect classification performance?

To address **RQ(2.1)**, we evaluate `Llama2-13b-chat` on response selection and addressee recognition in a zero-shot scenario (Section 4.2). These tasks capture different types of information: response selection relies on textual information to choose the next message in a conversation, while addressee recognition requires more structural awareness to infer speaker characteristics, conversation flow and interactional patterns. For each conversation, we design input combinations of conversation transcripts, interaction transcripts, generated summaries, and generated user descriptions, with the latter two being generated by `Llama2-13b-chat` (Section 4.3). We address also **RQ(2.2)** by designing prompts of different levels of verbosity for each input combination and task. Finally **RQ(2.3)** is addressed by designing a diagnostic approach, where the two tasks are evaluated on MPCs with a different number of speakers and structural characteristics (Section 4.4). This allows us to analyze the connection between task scores and structural characteristics of MPCs. The content of this chapter has been published in the paper *"Do LLMs suffer from Multi-Party Hangover? A Diagnostic Approach to Addressee Recognition and Response Selection in Conversations."* in the Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (Penzo et al., 2024b).

The software to perform the experiments and the processed data are available on the dedicate Github repository at the link https://github.com/dhfbk/MPH.

4.1 Related Work on Conversational Component Modeling

Classifying conversational components of multi-party conversation can span from identifying roles among the speakers (Sapru and Bourlard, 2014), to detecting reply-to-relations,

like in the speaker identification task, turn taking task and addressee recognition task (Le et al., 2019). A similar task that focuses more on the reply-to-structure among the utterances is the dialogue disentanglement task (Qiu et al., 2020), useful for identifying parallel discussions inside the same conversation.

Gu et al. (2021) fine-tuned a `BERT`-base model for MPC processing, pretrained on five pretraining objectives (three for learning the interlocutor structure modeling and one for utterance semantics modeling) and tested on three downstream tasks: addressee recognition, speaker identification and response selection. With the advent of LLMs, researchers have tried to perform MPC tasks by using LLMs in a zero-shot setting (Tan et al., 2023), but lacking from the evaluation point of view and without reporting any insight about the capability on more complex items.

Recent MPC understanding studies focus on response selection and addressee recognition tasks (Ouchi and Tsuboi, 2016; Zhang et al., 2018b) to compare different classification approaches. Indeed, response selection is strictly related to textual (linguistic) information, while addressee recognition focuses on interaction information, thus permitting to analyze the performance of classification models from two different angles. However, both tasks can ideally benefit from cross-information between linguistic and interactional cues.

For both response selection and addressee recognition, researchers have fine-tuned transformer-based models incorporating speaker information (Wang et al., 2020; Gu et al., 2021; Zhu et al., 2023; Gu et al., 2023), used Graph Neural Networks (GNNs) for interaction modeling (Hu et al., 2019; Gu et al., 2022a), or leveraged dialogue dependency parsing (Jia et al., 2020). Recently, Tan et al. (2023), explored zero-shot capabilities of ChatGPT (OpenAI, 2022) and `GPT4` (OpenAI et al., 2024) in MPCs, focusing only on the overall classification performance. At the same time, there is a gap in the NLP literature concerning the evaluation of MPC systems based on structural aspects. Past research has focused on textual information, for instance by using candidate rankings (Mahajan et al., 2022) or just looking at conversation length and number of users (Gu et al., 2023). The work we present in Chapter 3 provided a first exploration of the role of conversation structure in stance detection, showing that it benefits classification only when large training data are available.

For what concerns summarizing multi-party conversations, models need to contextualize both message content and interactions. Approaches have included fine-tuning transformer-based models (Feng et al., 2021), recombining sentence embeddings (Gao et al., 2023), leveraging GNNs (Chen and Yang, 2021; Xiachong et al., 2021), and using instruction-based models (Hua et al., 2024). Summarizing the dynamics and trajectories of MPCs, where a model's understanding of the conversation structure and interactions is critical, has been recently addressed by Hua et al. (2024). The authors also evaluate the

summaries of conversation dynamics with a classification task (i.e., forecasting the future derailment of the conversation as in Zhang et al., 2018a). Hua et al. (2024) point out that conventional summaries heavily focus on the textual content and what individual speakers say, while ignoring the interactions between speakers and the conversation flow. Past works (Yu et al., 2019) have also shown that user personality significantly impacts multi-party conversations, as shown by tasks of automatic personality detection from conversational contexts (Stajner and Yenikent, 2020) and user relationship dynamics (Choi et al., 2020).

The work that is most similar to our contribution is Tan et al. (2023), since they also use a generative model in a zero-shot setting to address response selection and addressee recognition. The main difference is that, instead of focusing only on generic accuracy scores, we propose a diagnostic approach for evaluating models for MPC processing. We use response selection and addressee recognition as proxy tasks and focus particularly on the contribution of interactional information, by (I.) creating diagnostic datasets, each with a fixed number of users, and (II.) putting in relation classification performance to specific network metrics (i.e., degree centrality and average outgoing weight of the speaker node).

4.2 Task Description

Our experiments revolve around two tasks that do not need a manual annotation as long as the used MPC data include speaker, addressee and related utterances.

Response Selection. Response selection (RS) is the task of choosing the text of the next message given a conversation C, the id of the speaker of the next message and a set of candidate responses. In our experiments, we cast response selection as a binary classification task, since the system has to choose between two possible candidates (similar to the R2@1 task in Gu et al., 2021).

Addressee Recognition. Addressee recognition (AR) is the task of predicting the addressee of the next message given a conversation C, the id of the speaker of the next message and a set of candidate addressees. The set of candidate addressees include all speakers involved so far in the conversation C plus a "dummy" option, i.e., a user unseen in the conversation to check whether the classifier choice is fully random.

In both cases, the next speaker is given, and the model has to choose what will be the content of the message (response selection) or who will be the addressee (addressee

recognition).

For instance, consider a simple multi-party conversation with three users U_1, U_2, U_3 and five messages:

Speaker	Addressee	Message
U_1	U_2	*"Did you see the new project update?"*
U_2	U_1	*"Yes, I checked it this morning."*
U_3	U_1	*"I'm still reviewing it, any major changes?"*
U_1	U_3	*"Mostly small tweaks, nothing too critical."*
U_2	U_3	*"I think the timeline is reasonable."*

In this scenario, for response selection, the model would need to choose the content of the last message given the preceding conversation and the speaker U_2. A potential negative candidate for the last message could be: *"I haven't looked at it yet, so I can't comment."*. This response is plausible in general but does not fit the actual interaction context, as U_2 has already seen the project update. For addressee recognition, the model would have to predict that the last message from U_2 is addressed to U_3 given the previous conversation. This shows how structural and contextual cues are crucial for understanding and modeling multi-party conversations.

4.3 MPC Classification Workflow

In this section we describe the classification workflow implemented to perform response selection and addressee recognition. The workflow is shared between the two tasks.

4.3.1 Conversation Representation

The first step is modeling the input data to be included in the prompt used for classification. To analyze the contribution of contextual and structural information for response selection and addressee recognition we identify four ways to represent the conversation content. The first includes just the chronologically ordered list of speaker-message pairs. This input format is called **(i.) Conversation Transcript**.

The second way aims at including only interactional information to assess its contribution in the classification tasks when no textual content is given. We call it **(ii.) Interaction Transcript**, and we model it as a chronologically ordered list of speaker-addressee pairs without the actual turn content.

The third and fourth settings aim at assessing how reliable LLMs are in representing a sequence of turns and capturing the most relevant information. We prompt an LLM

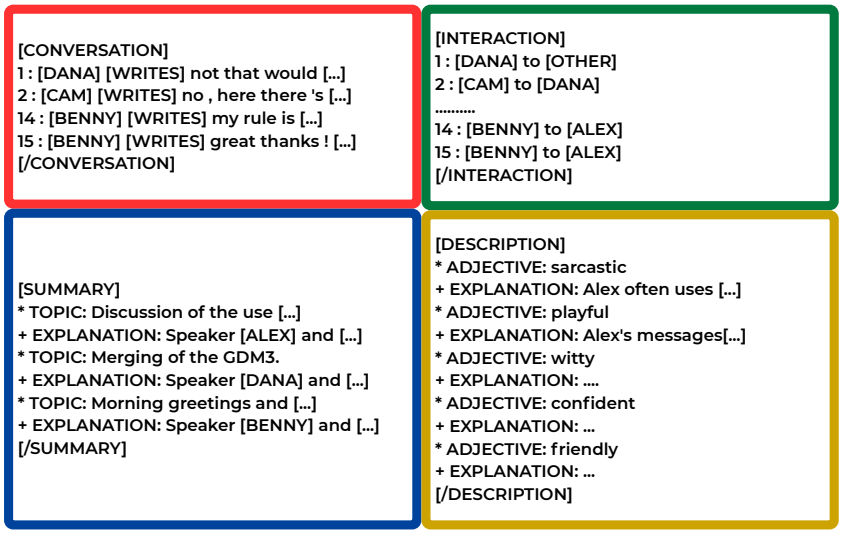

Figure 4.2: Example of the 4 possible conversation representations: I. Conversation Transcript (top left), II. Interaction Transcript (top right), III. Summary (bottom left) and IV. User Description (bottom right).

to provide two types of output, given the Conversation Transcript and the Interaction Transcript: **(iii.) Summary** of the conversation, expressed by the three main topics discussed, each followed by a brief explanation, and **(iv.) User Description**, i.e., a description of the behavior of the next speaker inside the given conversation, using five adjectives with a brief explanation for each. An example of each type of conversation representation is reported in Figure 4.2.

These last two representations are designed to replace the actual discussion content, retaining only the most relevant information. This approach can also be viewed as a first alternative solution to the problem in which the overall task is decomposed into a series of intermediate steps. Moreover, such approach can be useful in settings where storing and/or classifying whole conversations may be too expensive or when the actual conversation may become unavailable or impossible to reshare. Distributing raw conversational data with user IDs and full messages could in fact lead to potential malicious use, such as user profiling (Wen et al., 2023) or training LLMs to create fake personas (Huang et al., 2022). To ensure anonymization and avoid gender bias in classifier decisions, the original conversations are pre-processed by replacing real usernames with fake gender-neutral names, forcing the model to perform the MPC tasks using only "local" users, similarly to what we have done in Chapter 3, so that it is not possible to identify which users are the same across different conversations (details in Section 4.3.3).

4.3.2 Pipeline and Prompt Design

We use `Llama2-13b-chat` (Touvron et al., 2023) to perform text generation. Specifically, it is employed in four steps of our workflow:

I. to generate a summary of each conversation;

II. to generate user descriptions for each conversation;

and then for zero-shot classification, namely

III. response selection;

IV. addressee recognition.

For creating prompts, we follow the guidelines provided by Meta in their dedicated webpage[1].

In `Llama2-13b-chat`, each prompt is composed by a system prompt s that describes the task concatenated to an input prompt i that provides input information and the instruction command (i.e., the command to start the task to perform). For performing the generation of summaries and user descriptions, we use a greedy decoding mechanism and we design a generation prompt p_g with the following structure:

`[INST] <<SYS>> s <</SYS>> i [/INST]`

Instead, for the two classification tasks, the candidate responses are given. So, instead of having the LLM generate the output response, we evaluate the Conditional Perplexity, CPPL (Su et al., 2021; Occhipinti et al., 2024b) of all candidate responses given the classification prompt p_c, selecting as best output the candidate with the lowest CPPL. Other works dealing with classification tasks compute the probability of each candidate instead of CPPL (Liusie et al., 2024). However, probability can be applied to settings where each candidate includes only one token/word, whereas in our response selection task the candidates are sentences of variable length. We provide further details about the conditional perplexity in Section 4.3.5.

Each classification prompt p_c includes a system prompt s and an input prompt i. Moreover, we add a "beginning of output" prompt b, in order to evaluate CPPL only on the candidate responses. The prompt p_c presents the following structure :

`[INST]<<SYS>>` s `<</SYS>>` i `[/INST]` b

which leads to the full prompt p_{c_i} with the candidate responses r_i being:

`[INST]<<SYS>>` s `<</SYS>>` i `[/INST]` b r_i

[1] `https://llama.meta.com/get-started/#prompting`

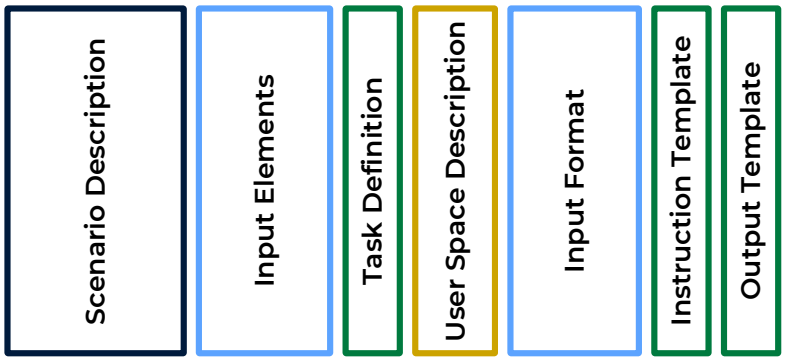

Figure 4.3: Graphical representation of the system prompt organization.

4.3.3 Prompt schemes and combinations

In our experimental setup, we establish a fixed template for all system prompts, as shown in Figure 4.3, consisting in 7 sections:

- **Scenario Description**: describes the scenario, defining messages and interactions between speakers and addressees;
- **Input Elements**: lists the input elements provided to the model according to the input combination, for example CONV, CONV+INTER, INTER+SUMM, etc.;
- **Task Definition**: defines the task to be performed (response selection, addressee recognition, generating summaries, or generating user descriptions);
- **User Space Description**: defines which users are involved as speakers or addressees;
- **Input Format**: specifies how the input elements are presented in the prompt;
- **Instruction Template**: details how the task instruction command is written in the prompt;
- **Output Template**: defines how the generated output should be organized.

The Scenario Description and User Space Description remain consistent across all tasks and combinations; three sections, i.e. Task Definition, Instruction Template and Output Template, vary depending on the specific task (e.g., response selection, addressee recognition, summarization, description); two sections, i.e. Input Elements and Input Format, are constructed modularly based on the chosen input combination (e.g., CONV, INTER, SUMM, DESC). In Figure 4.4 we report how the different pieces of input information are related to each other.

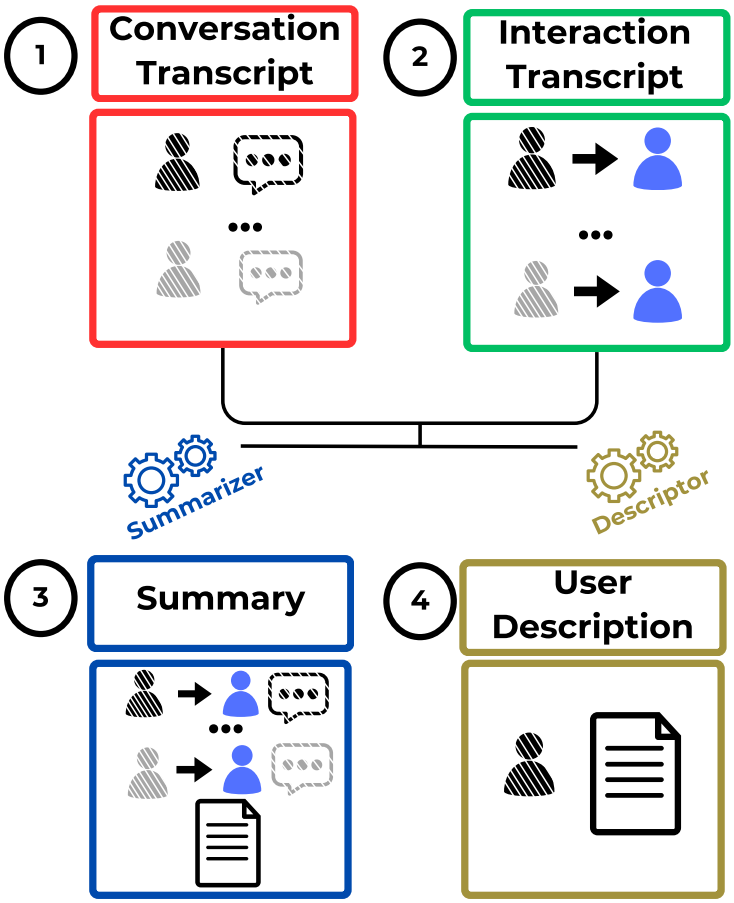

Figure 4.4: *Experimental setup.* First we create the conversation transcript (1) and the interaction transcript (2). From these, we extract the summary and the user description by using a specifically prompted LLM (3,4).

There is an ongoing discussion about evaluating instruction-based models particularly considering the high sensitivity of their performance to different levels of prompt verbosity (Sclar et al., 2024). For this reason, we identify a first dimension across all task, calling it "prompt scheme". Each prompt scheme consists in totally writing all the sections from scratch and recreating the prompts across tasks and combinations.

For user anonymization, we replace each original username with one of the following ungendered user tags: [ALEX], [BENNY], [CAM], [DANA], [ELI], and [FREDDIE]. The tag [ALEX] is always assigned to the next speaker. A tag [OTHER] is used in the Interaction Transcript when the related message explicitly refers to a user who does not contribute to the conversation. This is consistent with the work in Chapter 3, since the user tag is only locally consistent, making the model able to perform the task looking only at the local conversation dynamics.

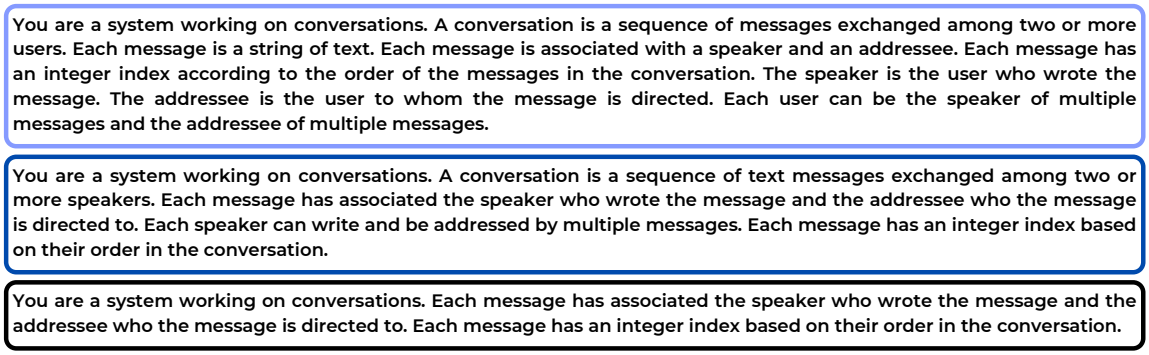

Figure 4.5: Example of the beginning of the system prompt in the three prompt schemes, from the most verbose (top) to the most concise (bottom).

4.3.4 Prompt Details

We compare three distinct prompt schemes with varying levels of verbosity to test LLM classification robustness and prompt sensitivity (Sun et al., 2024). Each prompt varies in terms of being more or less explicit in providing information. This leads us to have, for each prompt for a specific input combination and task, one *verbose* version, one *concise* version and one *medium* version. Our hypothesis is that the verbose prompts, giving more detailed instructions can potentially improve classification performance of `Llama2-13b-chat`. Figure 4.5 shows how the beginning of the system prompt changes across the three versions. More details and examples are provided in Appendix B.

4.3.5 Classification using CPPL

In most current LLM-based classification approaches, the model directly generates the output label through its own decoding process. However, this method offers limited control over the generation and may introduce biases due to the influence of additional or unintended tokens. As an alternative, since the set of possible labels is finite and predefined, we choose not to rely on the model's decoding strategy. Instead, we compute the conditional perplexity (CPPL) associated with each possible label and select the one with the lowest perplexity score as the final prediction.

Given the task $T \in \{RS, AR\}$, a classification prompt p_T, and the set of candidate responses $R_T = \{r_1, ..., r_m\}$, we extract as output the candidate with minimum conditional perplexity $\min CPPL(r_i|p), i \in [1, m]$, where

$$CPPL(r_i|p) = \frac{1}{P(r_i|p)^{1/|r_i|}} \tag{4.1}$$

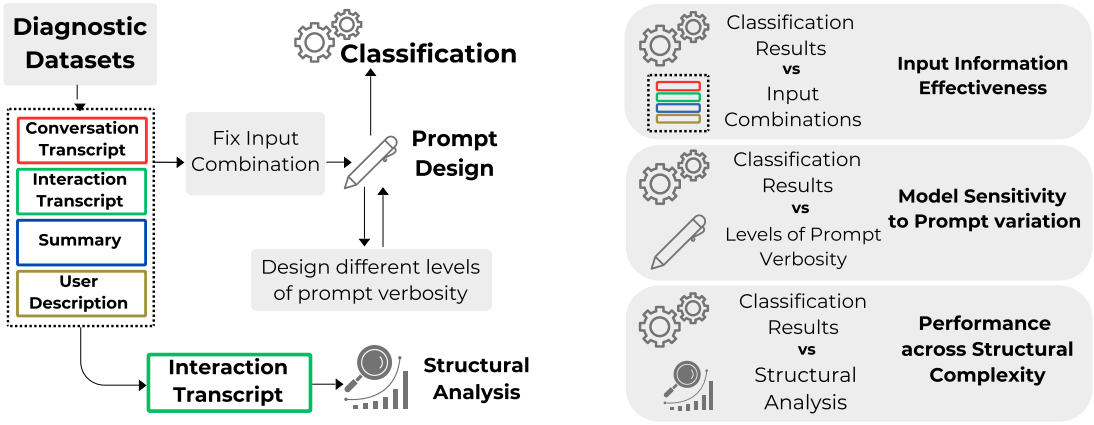

Figure 4.6: Schematic representation of our evaluation pipeline: on the left, the pipeline and the relation among the elements; on the right, the type of diagnostic evaluation we can perform.

according to the probability distribution of the model.

From the output CPPL, we can obtain a probability distribution over the set of candidates, so

$$P(r_k) = \frac{1/CPPL(r_k)}{\sum_{r_i \in R_T} 1/CPPL(r_i)} \tag{4.2}$$

4.4 Diagnostic Approach

To address the three research questions introduced at the beginning of this chapter, we aim to develop a diagnostic approach that isolates specific phenomena and minimizes confounding factors. A key aspect under analysis is the interplay between interaction structure in the conversation and classification performance. We identify two metrics to capture conversation complexity in terms of interaction graph and we also create subcorpora from a large conversation corpus, called *diagnostic datasets*, each with specific characteristics to test in relation to classification performance. In Figure 4.6 we report a schematic representation of our evaluation pipeline and the components involved.

4.4.1 Diagnostic Datasets

To analyze the impact that different interaction structures have on response selection and addressee recognition, we create four datasets derived from the Ubuntu Internet Relay Chat corpus (Ouchi and Tsuboi, 2016), which includes more than 800,000 conversations in English about how to solve technical issues. We use such large corpus because, to the best of our knowledge, it is the only one with an adequate dimension to allow us to extract

a good number of diagnostic MPCs with:

I. a defined number of users;

II. a good length of discussion;

III. a good structural variety, for each "diagnostic" subsets.

Moreover, it involves natural conversations with explicit addressee, which are necessary for the addressee recognition task.

To control the fluctuations in structural complexity, we limit the maximum conversation length to 15 messages (in line with the Len-15 version in Gu et al., 2021). We then create 4 MPC diagnostic subsets with conversations involving 3, 4, 5 and 6 users, which we call respectively *Ubuntu3 / Ubuntu4 / Ubuntu5 / Ubuntu6*. Then, for all 4 subsets, we proceed as follows:

I. for each conversation, we extract the undirected and unweighted interaction graph as explained above;

II. we keep only the conversations where the corresponding undirected and unweighted interaction graph is connected.

Finally, we anonymize the users, by replacing each username with a fake username, as already mentioned in Section 4.3.3. In this way, Ubuntu3 involves user tags up to [CAM], Ubuntu4 up to [DANA], Ubuntu5 up to [ELI], and Ubuntu6 up to [FREDDIE], in alphabetical order.

The resulting diagnostic datasets have respectively 1200, 635, 520 and 350 conversations. These datasets are used as test sets for evaluating response selection and addressee recognition in a zero-shot setting.

4.5 Experiments

Given the four types of input presented in Section 4.3.1, we design five input combinations to test in our prompts for both tasks:

I. only the conversation transcript (CONV);

II. the conversation transcript and the interaction transcript (CONV + INTER);

III. the interaction transcript and the conversation summary (INTER + SUMM);

IV. the interaction transcript and the user description (INTER + DESC);

v. the interaction transcript, the conversation summary and the user description (INTER + SUMM + DESC);

where CONV stands for the conversational transcript, INTER for the interaction transcript, SUMM for the summary and DESC for the user description.

For the addressee recognition task, we test a sixth combination:

vi. INTER, which corresponds only to the interaction transcript.

INTER is not relevant for response selection since it does not include any linguistic information. All combinations and prompt schemes are tested across the 4 diagnostic datasets.

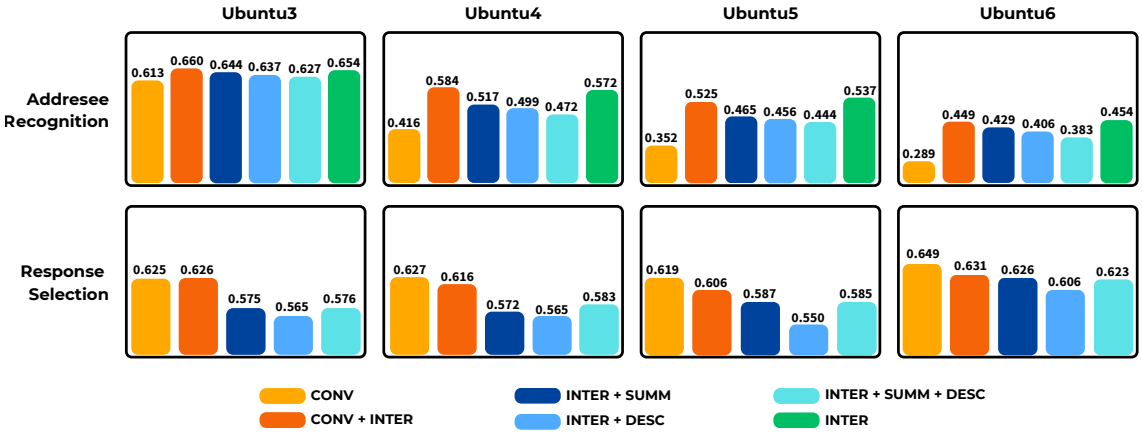

Figure 4.7: Addressee recognition and response selection macro-accuracy results (y axis), for each combination and for each dataset. The height of the columns represents the best macro result across the three prompt schemes. Note that for addressee recognition the number of classes in each Ubuntu subset changes, ranging from four (Ubuntu3) to seven (Ubuntu6), since the set of possible addressees includes the speakers involved in each conversation, plus the dummy option (see Section 4.2). For this reason, results across different Ubuntu subsets on addressee recognition should not be compared, and the lowest accuracy is achieved on Ubuntu6.

4.6 Macro Results and Structural Evaluation

4.6.1 Macro-results on the best run

In Figure 4.7, we present the macro accuracy for both tasks across all 4 diagnostic datasets. The columns show the highest accuracy achieved among the 3 prompt schemes with varying level of verbosity. In Table 4.1 we report the relative gaps between accuracy achieved

with the best prompt and the average accuracy obtained for each input combination and diagnostic dataset. In Table 4.2 and Table 4.3 we report the raw results for each prompt version, respectively for addressee recognition and response selection.

Addressee Recognition. In the addressee recognition task, the number of classes (i.e. number of addressees) on each Ubuntu subset changes, ranging from four (Ubuntu3) to seven (Ubuntu6), since the set of possible addressees includes the speakers involved in each conversation, plus the dummy option (see Section 4.2). For this reason, results across different Ubuntu subsets on addressee recognition should not be compared, and the lowest accuracy is achieved on Ubuntu 6, being its classification based on seven possible addressees. In addressee recognition, the CONV+INTER and INTER combinations consistently perform best across all datasets. Instead, the CONV combination, serving as our "text-only" baseline consistently shows the worst performance, with a relative performance gap with the best input combination growing from 7.1%, to 36.5% as soon as the number of users increases. If we consider replacing the original conversation with a summary (SUMM) or user description (DESC), we observe that the former outperforms the other on all datasets, although adding the original conversation to the structure (CONV+INTER) still outperforms both alternatives.

Response Selection. In the response selection task, the CONV and CONV+INTER combinations consistently perform the best across all datasets. Among the combinations with summary and/or user description, INTER+SUMM+DESC performs the best (in Ubuntu3 and Ubuntu4) or extremely close to INTER+SUMM, which is the best in Ubuntu5 and Ubuntu6. The INTER+DESC input combination yields the lowest classification performance on all datasets.

This analysis shows that the interaction transcript (i.e., the interactional information) is fundamental for achieving the best result in addressee recognition. On the other hand, the conversation transcript is fundamental for achieving the best results in response selection, which in fact is a more text-oriented task, based on information mainly available in the conversation itself. However, using summaries of conversations may be a viable alternative, achieving results closer to the best input combinations in the setting with more users (i.e. Ubuntu6) for both tasks.

4.6.2 Prompt Sensitivity

In this section we compare the highest accuracy and average accuracy among the 3 prompt schemes, for each input combination. Given b the accuracy of the best prompt scheme

and a the average among the 3 prompt schemes, we define as *relative gap* the relative worsening from the best to the average: $gap_{rel} = 1 - a/b$. A larger relative gap suggests greater sensitivity of the model to the prompts used, which leads to fluctuations in the classification results.

COMBINATION	**Task**	**Ubuntu3**	**Ubuntu4**	**Ubuntu5**	**Ubuntu6**
CONV	AddRec	2.7$^\diamond$	0.6	5.8$^\diamond$	2.0
	ResSel	0.8$^\diamond$	0.9$^\diamond$	0.8$^\diamond$	0.6
CONV+INTER	AddRec	7.1$^\diamond$	10.9$^\diamond$	4.5$^\diamond$	4.9$^\diamond$
	ResSel	0.4$^\diamond$	0.6	1.1*	0.8*
INTER+SUMM	AddRec	2.5*	6.5$^\diamond$	3.6*	6.7*
	ResSel	0.8*	1.0	1.7	0.8$^\diamond$
INTER+DESC	AddRec	2.7$^\diamond$	5.6$^\diamond$	4.8$^\diamond$	8.9$^\diamond$
	ResSel	2.1$^\diamond$	0.9*	1.3*	1.6
INTER+SUMM+DESC	AddRec	1.5$^\diamond$	4.0$^\diamond$	3.2*	6.0$^\diamond$
	ResSel	0.2$^\diamond$	1.4	1.2	0.6$^\diamond$
INTER	AddRec	5.6$^\diamond$	8.3$^\diamond$	8.5$^\diamond$	6.3$^\diamond$

Table 4.1: Relative gap (%) between the best prompt result and the average, for each input combination and diagnostic dataset, and for each task (i.e., addressee recognition and response selection). We put a $\diamond$ when the best prompt is the verbose version, a $*$ when the medium-length version is the best and nothing when the best is the concise version.

In Table 4.1 we report the relative gaps between accuracy achieved with the best prompt and the average accuracy obtained for each input combination and diagnostic dataset. Results on the addressee recognition task tend to be more sensitive to prompt formulation compared to the response selection task, especially in the CONV+INTER combination. Indeed, the relative gap between the best run and the average results is remarkably larger for addressee recognition than for response selection across all diagnostic datasets. Moreover, in the addressee recognition task, the INTER combination has similar prompt sensitivity to CONV+INTER.

Overall, we observe that for addressee recognition, where interactional information is more relevant, classification performance tends to vary more with different prompt verbosity compared to response selection, where linguistic information has a higher weight.

If we analyze what is the effect of prompt verbosity on classification performance, we observe that in the majority of settings and configurations, the verbose version of the prompt is the best performing one for addressee recognition. For response selection, instead, there is no evidence of benefit from using more or less verbose options.

COMBINATION	Prompt Scheme	Ubuntu3	Ubuntu4	Ubuntu5	Ubuntu6
CONV	verbose	0.613	0.414	0.352	0.277
	medium	0.582	0.409	0.344	0.283
	concise	0.595	0.416	0.298	0.289
CONV+INTER	verbose	0.660	0.584	0.525	0.449
	medium	0.609	0.501	0.513	0.431
	concise	0.571	0.477	0.465	0.400
INTER+SUMM	verbose	0.623	0.517	0.448	0.397
	medium	0.644	0.491	0.465	0.429
	concise	0.617	0.441	0.433	0.374
INTER+DESC	verbose	0.637	0.499	0.456	0.406
	medium	0.604	0.457	0.442	0.380
	concise	0.618	0.458	0.404	0.323
INTER+SUMM+DESC	verbose	0.628	0.472	0.429	0.383
	medium	0.620	0.455	0.444	0.374
	concise	0.607	0.433	0.417	0.323
INTER	verbose	0.654	0.572	0.537	0.454
	medium	0.626	0.515	0.498	0.434
	concise	0.573	0.487	0.438	0.389

Table 4.2: Performance comparison (measured as accuracy) for *addressee recognition* across prompt schemes and input combinations.

COMBINATION	Prompt Scheme	Ubuntu3	Ubuntu4	Ubuntu5	Ubuntu6
CONV	verbose	0.625	0.627	0.619	0.640
	medium	0.624	0.619	0.613	0.646
	concise	0.612	0.617	0.610	0.649
CONV+INTER	verbose	0.626	0.611	0.602	0.620
	medium	0.626	0.609	0.606	0.631
	concise	0.618	0.616	0.590	0.629
INTER+SUMM	verbose	0.572	0.570	0.569	0.626
	medium	0.575	0.556	0.573	0.614
	concise	0.564	0.572	0.587	0.623
INTER+DESC	verbose	0.565	0.553	0.540	0.597
	medium	0.553	0.565	0.550	0.586
	concise	0.542	0.562	0.538	0.606
INTER+SUMM+DESC	verbose	0.576	0.570	0.573	0.623
	medium	0.574	0.570	0.575	0.614
	concise	0.573	0.583	0.585	0.620

Table 4.3: Performance comparison (measured as accuracy) for *response selection* across prompt schemes and input combinations.

4.6.3 Effect of Different Output Templates for summarization/user description

As discussed in Section 4.3.1, when LLMs are employed to generate conversation summaries and user descriptions, the output is not restricted to a fixed label set or predefined schema. To enable a fair and robust analysis, we therefore investigate how variations in the output format influence the generated content. In particular, to assess the robustness of the generation process and control for potential prompt-induced biases, we explicitly test the impact of the output template used in the summarization and user description prompts.

We compare two prompt variants that differ in their output structure: in one version, the template includes an explicit *explanation* field for each generated topic or adjective, while in the other version this field is omitted. Importantly, in both cases the prompt instructions still request explanation, allowing us to isolate the effect of enforcing explanations at the output level rather than at the instruction level. This design enables us to evaluate whether constraining the output format affects the stability of the generated summaries and descriptions.

In Figure 4.8 we report the results across diagnostic datasets and tasks for INTER+ SUMM, INTER+DESC, and INTER+SUMM+DESC by averaging the results across the different classification prompt schemes but with the same output template for summarization/description. The average between the two different output templates does not differ much, with a maximum difference of 1.9% in the addressee recognition task-Ubuntu5 for INTER+SUMM+DESC, between the averages with same combination but different output templates.

Among the two output template variants, although the differences are generally small, a clear pattern emerges only for the INTER+SUMM+DESC configuration. In this setting, the template with explicit explanations leads to the the best performance in 7 out of 8 cases. In contrast, for the other configurations the advantage of including explicit explanations is less consistent: in INTER+SUMM, the explanation-based template performs better in 5 out of 8 cases, while in INTER+DESC it outperforms the alternative in only 2 out of 8 cases, with one additional tie.

Nevertheless, averaging the results across the two output templates provides trends that are consistent with those reported in Section 4.6. For this reason, and to reduce complexity in the analysis, we focused on a single output template (specifically, the more explicit one) in the previous experiments.

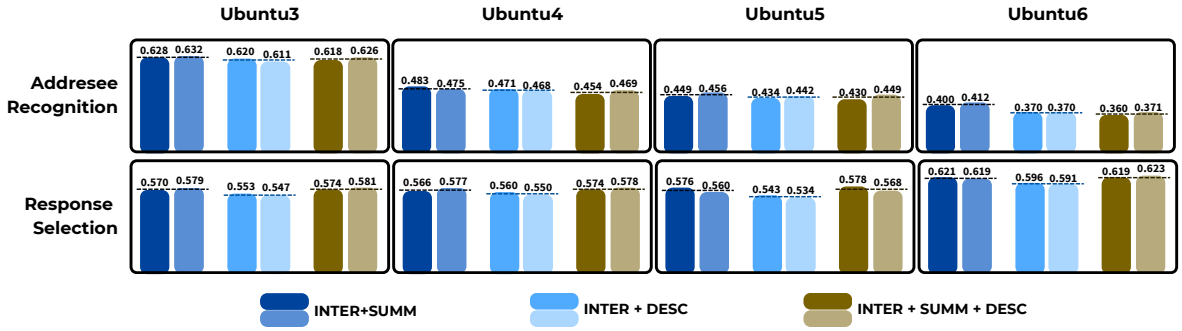

Figure 4.8: Model performance (macro accuracy results) on addressee recognition and response selection, across all diagnostic datasets, for each of the two output template for conversation summarization and user description.

4.6.4 Structural Evaluation

In Figure 4.9 and Figure 4.10, we present how the best run for each input combination varies in relation to the two node-level network metrics introduced in Section 2.4.2, i.e. degree centrality $deg(u)$, reported in Figure 4.9, and average outgoing weight $w^o_{avg}(u)$, reported in Figure 4.10, where u is the next speaker node (across all 4 diagnostic datasets). More informally, $deg(u)$ represents the number of users the next speaker has interacted with in the history of the conversation. Instead, $w^o_{avg}(u)$ indicates the average number of messages sent to the users with whom the next speaker has interacted (in our graphs, we rounded it at the closest integer number).

In the addressee recognition task, combinations containing the interaction transcript (+INTER) exhibit similar patterns, while the CONV combination displays distinct trends compared to the other combinations. Notably, $deg(u)$ shows the strongest correlation with accuracy scores across all datasets: higher $deg(u)$ values consistently correspond to lower accuracy. Furthermore, the gap between the top-performing combinations (CONV+INTER and INTER) and others widens significantly at lower $deg(u)$ values. For example, while the INTER combination consistently ranks among the best in terms of macro-results, it is outperformed by (or comparable with) other combinations across all diagnostic datasets as $deg(u)$ increases. This shows that using in the prompt INTER-only information is highly effective when the next speaker has spoken with few users in the transcripts (one or two), but it performs like the other combinations when the next speaker interacted with more than two users. As regards $w^o_{avg}(u)$, the correlation with accuracy is less pronounced, but generally, higher $w^o_{avg}(u)$ values correspond to higher accuracy in models that use interaction transcripts as input (with some minor fluctuations).

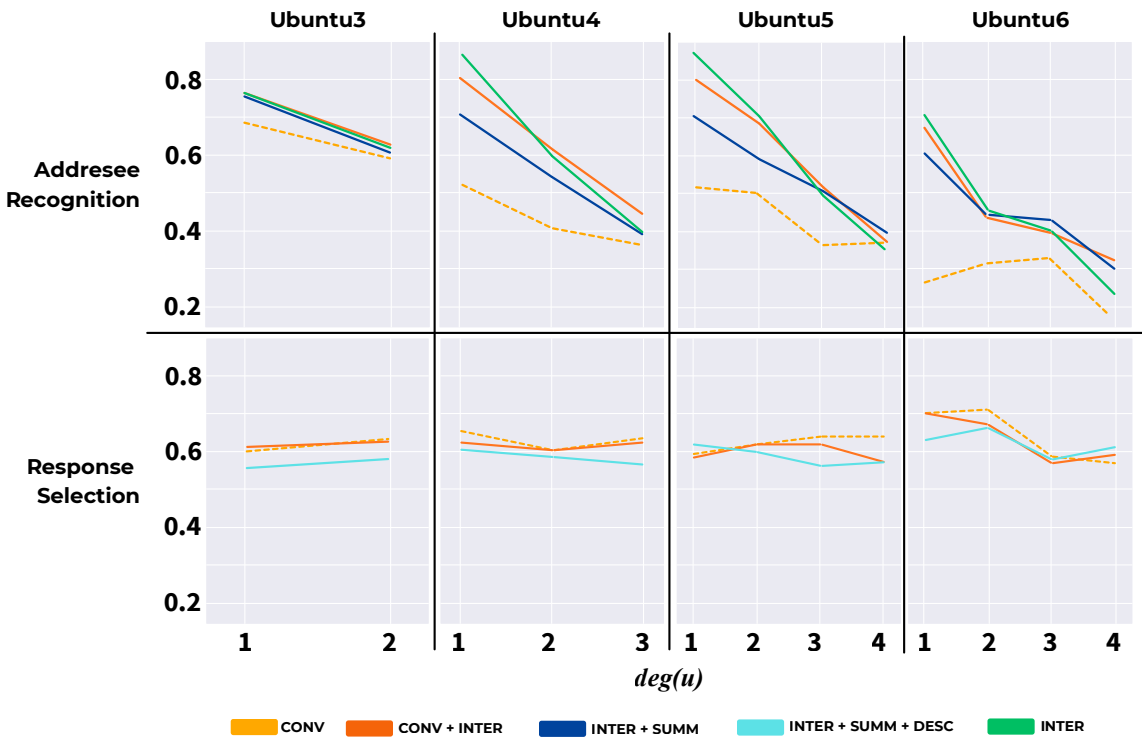

Figure 4.9: addressee recognition and response selection accuracy results (y axis) for the different values of $deg(u)$ of the speaker node u (x axis). We report the performance of the three best input combinations for each task, plus CONV in addressee recognition which serves as text-only baseline.

In the response selection task, we do not notice any clear correlation between $deg(u)$ and any increasing/decreasing behavior in the accuracies. Also for what concerns the gap among the models, there is no consistent trend across the different datasets. The same holds for $w^o_{avg}(u)$. This suggests that the performances on the response selection task are not related to the structural dimension.

4.7 Discussion of the results on component modeling

Our comparative evaluation shows three main findings, listed below.

Input combination performance – RQ(2.1). Regarding the best-performing combinations, in addressee recognition, INTER and CONV+INTER consistently emerge as the top performers, with comparable results. This suggests that having only the interaction transcript is sufficient in our experimental setting for this task. Similarly, in response selection, CONV and CONV+INTER consistently outperform other combinations, with

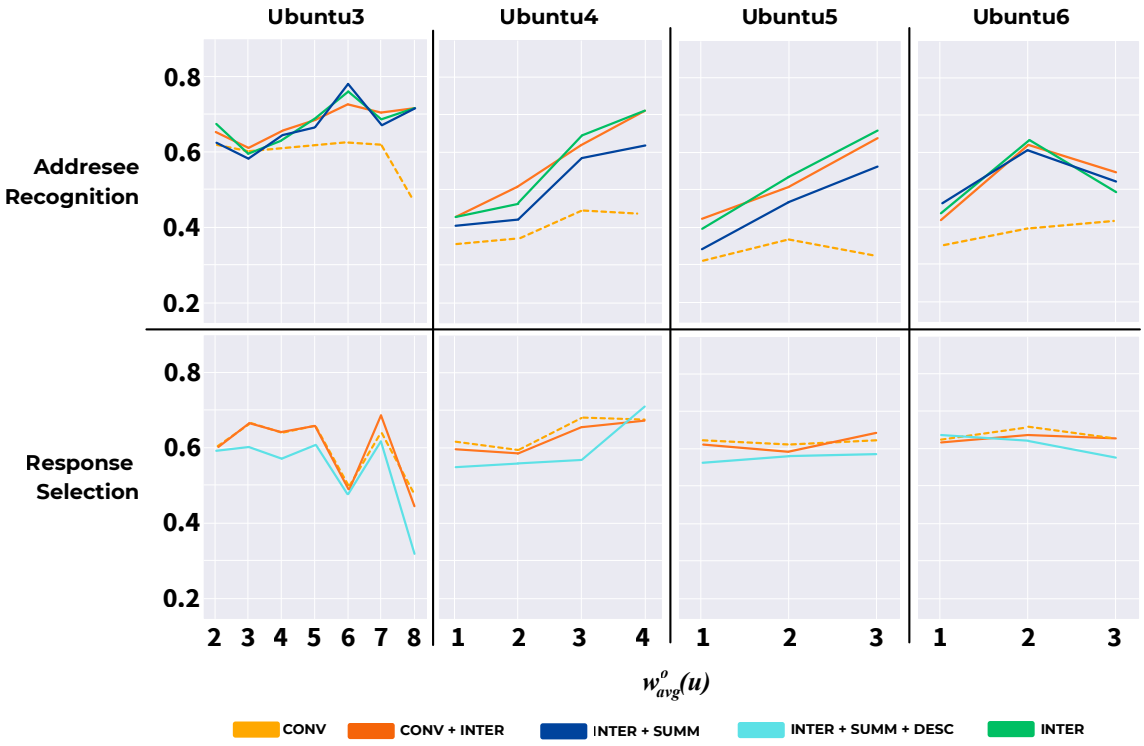

Figure 4.10: addressee recognition and response selection accuracy results (y axis) for the different values of $w^o_{avg}(u)$ of the speaker node u (x axis). We report the performance of the three best input combinations for each task, plus CONV in addressee recognition which serves as text-only baseline. $w^o_{avg}(u)$ is rounded at the closest integer number.

the former widening the gap from the latter as more users are added. This indicates that having only the conversation transcript is adequate for the task in our experimental setup mostly based on textual information. The inclusion of summary and/or the user description leads to a decline in performance. This may depend on the fact that `Llama2-13b-chat` sometimes is prone to generate bad summaries and descriptions, struggling to accurately capture the content of the conversation. On the other hand, it may also depend on the model difficulties to employ this information for classification.

Interestingly, in both tasks, the gap between the best input combination and the best among the ones including summaries decreases as more users are involved (except for Ubuntu3 in the addressee recognition task): this suggests that summaries may be effective when dealing with a large number of users, as possible noise introduced by the summary is equally challenging to dealing with complex conversations. In general, user descriptions appear to be ineffective.

Prompt Verbosity – RQ(2.2). Addressee recognition, which benefits from interactional information, shows greater sensitivity to prompts compared to response selection, which is mostly a text-based task. This difference could be due to the similarity of response selection to tasks used to pretrain LLMs. Indeed, the response selection task is similar to a "response generation" task, where the perplexity of the two candidates is evaluated.

As regards classification performance obtained with the different prompt versions, the verbose version of the prompt tends to be the best option for addressee recognition, probably because it helps the model in better capturing the interactional information, which is crucial for this task. For response selection, instead, there is no consistent improvement in using a more verbose prompt, probably because all the linguistic information necessary to perform the task is already expressed in the conversation.

Structural Complexity – RQ(2.3). Our structural analysis (Figure 4.9 and Figure 4.10) highlights the limitations of relying solely on macro-level accuracy metrics, particularly for the addressee recognition task, and shows the importance of breaking down performance with respect to underlying network properties.

In addressee recognition, if we consider the correlation between classification accuracy and $deg(u)$, especially in Ubuntu4/Ubuntu5/Ubuntu6, we observe that the performance gaps between the best input combination (i.e., INTER for Ubuntu3/Ubuntu4, CONV+INTER for Ubuntu5/Ubuntu6) and INTER+SUMM combination in macro results is mainly driven by instances where the next speaker node interacts with only a few other users, for all diagnostic datasets. As the degree centrality increases, all combinations experience a general drop in performance. This analysis underscores that macro

results offer only a surface-level understanding of the model's capabilities and are heavily influenced by dataset characteristics. A closer examination reveals that the best input combinations perform very well in simpler conversations, with limited generalization to more complex interaction structures.

A model being able to effectively capture both interactional and linguistic information should ideally show less performance degradation at increasing degree centrality, rather than performing well only on samples with lower degree centrality. Regarding $w^o_{avg}(u)$, it suggests that having more messages directed towards the involved users may help determine the last addressee. However, as shown by the performance of INTER, this is not due to message information. Nevertheless, this could still be an effect of the degree centrality, as the conversation length is fixed at 15 messages and higher values of $w^o_{avg}(u)$ likely correspond to lower values of $deg(u)$.

In addressee recognition, if we consider only conversations with a complex structure, we observe that the inputs that address data minimization (e.g. those using conversation summary) reach a performance close to the best performing input combination. Instead, the worse performance with data minimization input is obtained when classifying examples with low structural complexity. Therefore, in the future it may be worth focusing on simple structures and try to address this performance gap, understanding its causes.

The structural analysis of the response selection results indicates that performance for this task is largely unaffected by network metrics. This observation can be interpreted in two ways: firstly, information on the structure of the conversational interactions may not be relevant when selecting a response. Alternatively, the model might effectively infer the conversation flow based solely on message content, maintaining consistent performance regardless of the "node complexity".

For both tasks (addressee recognition and response selection), we analyzed other node metrics (i.e., closeness centrality and clustering coefficient). Such metrics showed high correlation with the degree centrality, and for this reason we do not report them for simplicity of the discussion.

4.8 Main Findings on Modeling the Conversational Components

In this chapter, we evaluate the zero-shot performance of an LLM (i.e., `Llama2-13b-chat`) on two tasks based on multi-party conversations, namely response selection and addressee recognition. Our goal is to provide an in-depth analysis of different experimental settings tested for the two tasks, which include three different prompt types and six configurations to model the conversation text and its underlying interactions. Our analysis is performed on four diagnostic datasets with a fixed number of users. For each of them, we compute

two network metrics, i.e., degree centrality and average outgoing weight, to analyze how structural complexity interplays with classification performance. We devote particular attention to evaluating how strategies to replace the original conversation text could be effectively used in the prompts. This is very relevant to ensure a safe use of MPC corpora: if the same classification performance could be achieved by removing the original conversation, data resharing would not imply the risk of making personal or sensitive data available. Furthermore, malicious use of multi-party conversations, for example using them to train models with fake personas, would not be possible. Although promising, this research direction has not achieved fully satisfactory results.

The goal of our work is not much to yield the best possible classification accuracy on addressee recognition and response selection, but rather to provide an in-depth analysis of the possible dimensions contributing to the classifier performance on the two tasks. We believe that the interplay between textual and interactional information in multi-party conversations should be better analyzed in current evaluations, merging contributions from NLP and the network science community.

Still, this work presented some limitations. Indeed, the findings presented in this chapter are based only on subsets of a single dataset. During the development of this work, we took other datasets into account as possible candidates for our experiments, but when we analyzed them more in depth we found that they presented neither the structural characteristics which are necessary to build diagnostic datasets, nor the necessary amount of data to perform a good diagnostic analysis. Moreover, our experiments were conducted using only one instruction-based LLM in a zero-shot setting, as our primary goal was to present a novel evaluation pipeline. Furthermore, we evaluated classification performance based on the best run for each model and combination. However, it is important to note that claiming general capabilities of the model based on these results would be scientifically inaccurate. Comparing different LLMs would be necessary to better prove the generalization of our approach.

Given all these premises, a natural question arises: how can we obtain large amounts of high-quality and structurally rich conversational data? In the next chapter, we address this issue by exploring the use of LLMs to generate synthetic multi-party conversations with controlled characteristics, with the goal of producing realistic and engaging interaction patterns. Furthermore, we introduce a set of evaluation steps to assess the quality of such synthetic data.

Chapter 5

Generating Conversations

In the previous chapters, we first explored how to leverage the entire MPC context to improve text classification performance for individual messages by fine-tuning a `RoBERTa`-based model (Chapter 3). Subsequently (Chapter 4), we examined the zero-shot performance of `Llama2-13b-chat` on conversational component modeling tasks, namely addressee recognition and response selection. `Llama2-13b-chat`, at the time, represented one of the state-of-the-art models across a broad range of tasks. We also assessed the robustness of its performance with respect to prompt formulation and interaction complexity (Wei et al., 2023).

A common challenge observed across both chapters was the difficulty of obtaining high-quality and sufficiently large datasets that meaningfully represent the diverse possible structures of interactions. Additionally, there exists a general heterogeneity among available datasets in terms of annotations, data types, and other characteristics. Indeed, one of the major issues in working with multi-party conversations lies in data retrieval itself: social media data are often protected by privacy policies, and meaningful annotation processes typically require expensive human effort. For example, defining the addressee of a message is often ambiguous: some users may be considered more directly targeted than others, yet there is rarely clear right or wrong assignments. At the same time, messages with similar content but differing addressee focus can lead to substantially different conversational contexts. This is why human assessment should always remain part of the process; however, it also presents risks of being highly subjective, even when clear annotation guidelines are provided.

One potential remedy for the lack of structural diversity in multi-party conversations derived from social media data is to synthesize them, by explicitly constraining LLMs to generate multi-party conversations with specific characteristics, such as specific number of messages, number of speakers, precise speakers' stance distribution about the topic discussed, output format or interaction rules. To reflect real-world conversational complexity,

generated MPCs should include varied conversations, show different interaction patterns and topics as well as provide rich speaker-addressee relationships, e.g., multi-addressee interactions.

In this chapter, we propose generating synthetic MPCs using LLMs guided by specific constraints related to the above capabilities. We explore two generation strategies:

I. *One-Long* (OL) generation, where the LLM generates an entire multi-party conversation in a single step;

II. *Turn-by-Turn* (TT) generation, which builds the conversation sequentially, generating one turn at a time.

A comparison among the resulting multi-party conversations from both strategies highlights the (potential) discrepancies between how LLMs cast entire multi-party conversations to look human-like (One-Long strategy) and how they behave as participants in a multi-party conversation (Turn-by-Turn strategy). We propose a novel evaluation framework that combines several quantitative and qualitative dimensions of generated multi-party conversations, focusing on the extent of LLMs' compliance to provided content and structural constraints. We break down the *Research Question 3* presented in Section 1.1 in the following three key sub-questions:

RQ(3.1): Can LLMs be leveraged to generate large synthetic MPC datasets while maintaining compliance with predefined constraints on dialogue structure and participants' stance?

RQ(3.2): Which generation strategy (One-Long vs. Turn-by-Turn) produces higher-quality MPCs?

RQ(3.3): How can we effectively evaluate the variety and quality of the generated MPCs?

We test four popular LLMs and identify `Llama3.1` (Dubey et al., 2024) and `Qwen2.5` (Yang et al., 2024) as the best LLMs for complying the most with constraints. The Turn-by-Turn strategy seems to generate more constraint-compliant MPCs than the One-Long strategy. Moreover, the MPCs produced with a Turn-by-Turn approach exhibit greater lexical variability and semantic coherence. The generated MPCs also present a higher structural complexity than a widely-used corpus of "real" conversations (Ouchi and Tsuboi, 2016). Finally, a qualitative evaluation shows that LLMs can produce high-quality MPCs both in Turn-by-Turn strategy and One-Long strategy, rendering the choice of the LLM more important than the choice of generation strategy. The content of this chapter has been published in the paper *"Don't Stop the Multi-Party! On Generating Synthetic Written Multi-Party Conversations with Constraints"* in the Proceedings of the

Fortieth AAAI Conference on Artificial Intelligence (Penzo et al., 2026). The software to perform the MPC generation/evaluation and the MPCs that we generated are available on the dedicate Github repository at the link https://github.com/dhfbk/Constrained-SyntheticMPC.

5.1 Related Work on Generating Synthetic MPCs

Generating synthetic data to train machine learning models is a well-established strategy for improving performance on tasks affected by data scarcity (He et al., 2022). With the advent of LLMs, this approach has become even more accessible: models trained on massive amounts of data can now be leveraged to generate large-scale synthetic datasets, which can be used to train smaller and more accessible models among the public and research communities (Long et al., 2024; Kim et al., 2025).

While the generation of synthetic conversational data has been explored in various settings in recent years (Bao et al., 2023; Han et al., 2024; Suresh et al., 2025), scenarios involving multi-party conversations remained largely underexplored.

To the best of our knowledge, the only existing attempt at generating synthetic MPCs was made by Chen et al. (2023). However, their work primarily focused on conversations involving at most three participants, limiting the complexity of interactions. In contrast, our study explores the generation of MPCs with four or more participants, leading to more elaborate conversational dynamics. While this increased complexity allows for richer conversational structures, it also introduces a higher likelihood of generating errors, needing a rigorous evaluation process to assess the quality and consistency of the generated conversations.

Among the datasets presented in Section 2.3, again, the Ubuntu IRC corpus (Ouchi and Tsuboi, 2016) is the only dataset that we can use as a comparison for this work. Indeed, it is possible to retain from the initial set of 700 000 MPCs only those with the same number of messages, number of users and structural constraints as in our synthetic MPCs, obtaining a set of conversations with a size comparable to our synthetic datasets (details in Section 5.7).

5.2 Synthetic MPCs Generation

In Chapter 3, we focused on one-to-one interactions in which each reply was constrained to directly follow and address the immediately preceding turn. In Chapter 4, we relaxed this constraint by allowing replies to target non-adjacent turns, while still preserving a one-to-one interaction setting.

In this chapter, we take a further step forward by considering multi-party conversations as ordered sequences of turns in which each turn is characterized by three components: the speaker (*who* produces the turn), the message (*what* textual content is conveyed), and the addressees (*to whom* the turn is directed). This means that we are shifting from strict one-to-one replies to multi-addressees settings, better reflecting the complexity of real-world online discussions.

In this section, we first introduce the two generation strategies explored in our experiments (Section 5.2.1). We then describe the set of discussion topics selected for the synthetic MPCs (Section 5.2.2), and finally detail the constraints specified in the generation instructions (Section 5.2.3).

5.2.1 Generation Strategies

We test two strategies for generating multi-party conversations using instruction-based models. Our main goal is to determine whether LLMs behave differently when asked to generate an MPC as a unique narrative compared to acting as an interactive participant within the conversation. With this motivation, we use each LLM in the two following generation strategies.

One-Long generation strategy (OL). The LLM is prompted to generate the entire conversation in one pass. In this strategy, generation starts with a system input prompt that defines all the constraints and the task, asking then to generate the entire conversation. This strategy follows a one-step, long-generation process, based on a single input context.

Turn-by-Turn generation strategy (TT). Here the LLM is prompted to generate the conversation incrementally, provided the conversation history. The model is prompted multiple times to perform one of three tasks:

I. generate a speaker;

II. generate interactions between a speaker and addressees (given the candidate speakers/addressees);

III. generate a message (given the interaction).

The process begins with a system prompt specifying the constraints and these three tasks. The model is first prompted to generate each speaker and assign them a stance on a controversial topic. Then, the LLM generates a sequence of interactions and messages (one at a time), iteratively augmenting the conversation: this means that the context provided to the LLM increases monotonically in size with consecutive turns.

5.2.2 Topics

Progressive	**Conservative**
ban of targeted killing	allowance of targeted killing
ban of the death penalty	allowance of the death penalty
recognition of the right to abortion	ban of abortion
recognition of the right to euthanasia	ban of euthanasia
recognition of Palestinian state	non-recognition of Palestinian state
ban of mandatory military service	mandatory military service
ban of nuclear weapons	support for nuclear weapons
mandatory sex education in schools	optional sex education in schools
guarantee of online teaching	mandatory in-person teaching
fight to climate change	opposition to regulations for action on climate change
incentives for renewable energy	incentives for energy from fossil fuels
ban of facial recognition technology	incentives for facial recognition technology
incentives for AI research	opposition to AI research incentives
mandatory vaccination for children	optional vaccination for children
ban of animal testing	allowance of animal testing
incentives for organ donation	opposition to organ donation incentives
ban of racial profiling	allowance of racial profiling
incentives for immigration and asylum	support to immigration contrast and stricter asylum rules
universal healthcare	support to private healthcare
legalization of marijuana	ban of marijuana
legalization of same-sex marriage	ban of same-sex marriage
legalization of surrogate motherhood	ban of surrogate motherhood
programme for the reduction of the gender pay gap	increase of the gender pay gap in favor of men
limitation to gun ownership	right to unrestricted gun ownership
holocaust remembrance mandatory in schools	optional holocaust remembrance in schools
ban of zoos	support for zoos
protection of endangered species	opposition to endangered species protection
organization of pride parades	ban of pride parades
allowance of tattoos	ban of tattoos
cohabitation of couples before marriage	mandatory marriage before cohabitation
ban of arranged marriages	right to arranged marriages
US staying in NATO	US leaving NATO
Germany staying in EU	Germany leaving the EU
mandatory acceptance of mobile payments	ban of mobile payments
lowering university tuition fees	increase in university tuition fees
mandatory cameras on police officers	freedom of police officers to refuse cameras
freedom of blasphemy	punishment for blasphemy
legalization of adoption by same-sex couples	ban of adoption by same-sex couples

Table 5.1: List of topics, paired according to their Progressive and Conservative version.

To generate a controlled set of synthetic MPCs, we identify a set of controversial topics to encourage more polarized and clear statements from speakers based on their assigned stance. Specifically, following Li et al. (2024b), we select 38 topics (manually selected, freely available in the related Github repository[1]) and create two stance statements for each topic: one reflecting a *progressive* perspective and the other a *conservative* perspective. Specifically, we picked the most polarizing topics, in order to foster more clear-cut stances during the generation of MPCs. Moreover, the statements are created by avoiding potential biases such as framing statements negatively or using specific terms exclusively in one category. Finally, we instruct the LLMs to generate conversations based on each of the resulting 76 statements.

[1] `https://github.com/tianyi-lab/DEBATunE`

5.2.3 Conversation Constraints

To ensure that the generated conversations feature rich interaction patterns with diverse dynamics, we instruct the model to follow specific constraints, described in the system prompts created for each generation strategy (for details about how this was operationalized in prompts, see Appendix C.1).

Output Format. To enable automated analysis, the generated output must respect a structured `JSON` format with all the information needed. So, each generated MPC must be a dictionary with two main keys, namely `conversation` and `speakers`. The `conversation` field must include a list of dictionaries, each with specific fields such as `speaker`'s name, turn `message` and `addressees`, i.e. the list of participants in the conversation to whom the message is directed. The `speakers` field includes the speaker's `name` and the `stance` with respect to the conversation topic.

Interactions. These constraints refer to three requirements in the generated multi-party conversations:

I. all speakers appearing in the interactions must be present in the speakers' list (i.e., the LLM should not invent a new speaker half way through the conversation);

II. `addressees` must cover at least once also the role of `speaker`;

III. self-interactions, i.e. speakers sending a message to themselves, are not admitted.

Speaker's Contribution. All speakers in the `speakers` field must be authors of at least one turn in the conversation.

Number of Speakers. In order to enable complex interaction structures, each multi-party conversation must involve between 4 and 6 speakers.

Number of Messages. Each generated multi-party conversation must include 15 messages across all speakers, with a maximum of 50 words per message.

Speaker's Stance. We specify the exact number of speakers for each stance (e.g., 2 with the *pro* and 3 with *against* stance).

We additionally request that the first turn always addresses all participants: this ensures that the generated interaction graph is connected, as required for structural analysis (see Section 5.3.3).

5.3 Evaluation Framework

We design an evaluation framework aimed at assessing different aspects of the generated multi-party conversations. It is composed of four blocks, which we detail below.

5.3.1 Compliance with Constraints

The first dimension considered in the evaluation framework is to what extent the synthetic MPCs comply with the format and structural constraints given in the prompt. For each generated MPC, this framework must verify: (I.) the correctness of the *Output Format*; (II.) the correctness of the *Interactions*; (III.) the *Contribution* of each speaker; (IV.) the *Number of Speakers*; (V.) the *Number of Messages*; (VI.) the distribution of the *Stance of the Speakers*.

All the computed values must be compliant with the constraints presented in Section 5.2.3. Only for the Number of Messages, we relax the constraint by considering valid MPCs including less than 15 turns if they contain at least 2 messages per speaker (on average). After a manual check of the generated MPCs, we noticed that shorter or longer conversations may still represent high-quality data. Each value is computed separately and then used to identify how many multi-party conversations comply with *all* these constraints.

5.3.2 Analysis of Language Variability

A key risk for synthetic datasets is to suffer from low linguistic variability, due to repetitive examples obtained when using similar prompts (even if stochastic decoding is used), an issue already highlighted for dialogical settings (Occhipinti et al., 2024a). On the other hand, while generated MPCs should ideally be lexically rich, they should also be semantically coherent, i.e. different multi-party conversations about the same topic should exhibit a certain degree of semantic similarity. To control for these aspects, we compute the following three metrics.

Repetition Rate. Repetition Rate is a well-known metric from Bertoldi et al. (2013), which has already been used in synthetic conversational scenarios in the past (Bonaldi et al., 2022). It measures the rate of non-singleton n-grams within a cluster of MPCs. Repetition Rate is computed within each topic-based cluster by measuring repetitions across all MPCs in the cluster. The values are then averaged across clusters to assess the overall linguistic diversity.

String Similarity. Since Repetition Rate already reflects similarity across entire conversations, String Similarity is designed to penalize (with higher scores) pairs of conversations that contain also few highly similar turns, so promoting turn-level diversity. Specifically, given two conversations with a and b turns respectively, we compute string similarity scores for all $a \times b$ turn pairs. We then take the average of the top 5 scores as the String Similarity for that conversation pair. String similarity between pairs of turns is computed using thefuzz library[2] and is based on Lehvenstein distance. String Similarity is computed following an all-vs-all comparison strategy across every turn in all possible conversation pairs within the same topic. Then, the final topic-level value corresponds to the average across all conversation pair-scores of that topic and finally averaged across clusters.

Semantic Coherence. Semantic Coherence aims to measure how coherent the conversations are during all the turns. So, for each of the a turns in the first conversation, we keep the highest cosine similarity with any of the b turns in the second conversation, and vice versa. The final Semantic Coherence for that conversation pair score is the average of these $a + b$ scores. Semantic Coherence between pairs of turns is computed by first embedding each turn with `SentenceBERT-all-MiniLM-L6-v2` (Reimers and Gurevych, 2019) and then calculating pairwise cosine similarity. Again, Semantic Coherence is computed following an all-vs-all comparison strategy across every turn in all possible conversation pairs within the same topic. Then, the final topic-level value corresponds to the average across all conversation pair-scores of that topic finally averaged across clusters.

We provide representative examples of pairs of multi-party conversations with high Semantic Coherence and high String Similarity in Appendix C.2.

5.3.3 Interaction Structure Analysis

To describe and quantify the structural complexity of interactions in the generated MPCs, we compute the global network metrics and dyadic/tryadic metrics presented respectively in Section 2.4.2. A graphical overview of these structural metrics is reported in Figure 5.1.

Following Section 2.4.1, we represent MPC interactions with an *unweighted undirected graph* $G^{uw}_{ud}(C)$ and a *weighted directed graph* $G^{w}_{d}(C)$, where the weight of an edge corresponds to the number of messages sent in the direction of the edge.

For measuring the average activity of a speaker node u in the conversation, we compute the **Average Degree Centrality** $deg_{avg}(G^{uw}_{ud}(C))$, and the **Average Out-going**

[2] `https://github.com/seatgeek/thefuzz`

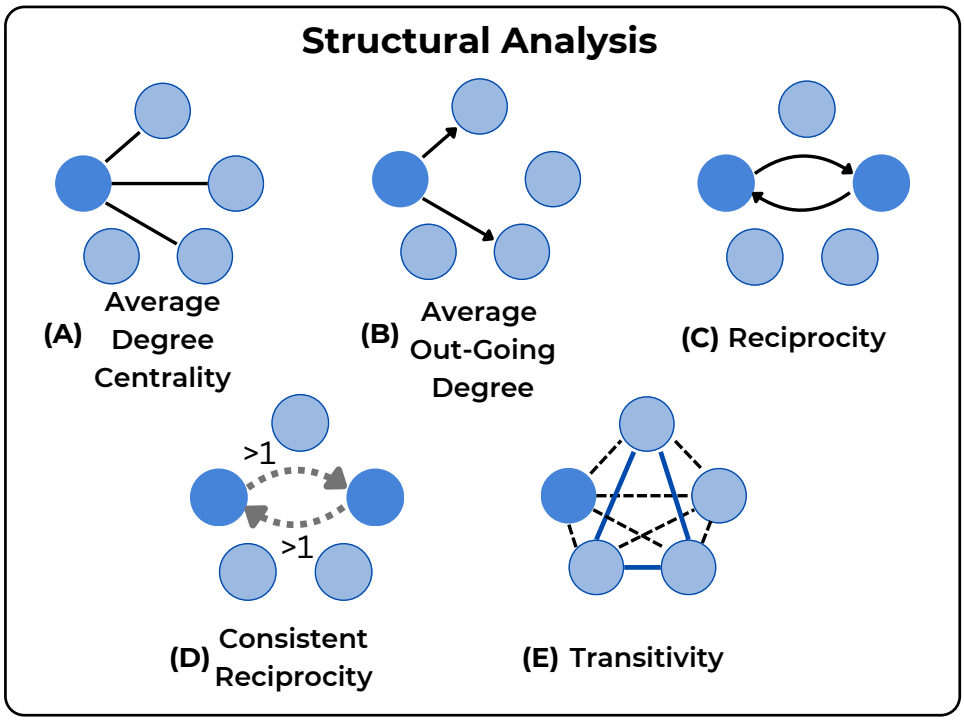

Figure 5.1: Overview of the structural metrics considered in our structural analysis.

Degree $outdeg_{avg}(G_d^w(C))$. Both averages are computed across all the nodes in the conversation and normalized according to their maximum possible values. Then, to quantify the presence of back-and-forth exchanges, we compute the **Reciprocity** $R(G_d^w(C))$, and the **Consistent Reciprocity** $R^w(G_d^w(C))$. Finally, to quantify how often speakers build "triads" of interactions, we compute the **Transitivity** $T(G_{ud}^{uw}(C))$. For all these metrics, higher values indicate more complex interactions in a conversation. Indeed, higher reciprocity (consistent or not) suggests more frequent back-and-forth exchanges. Again, higher average degree values means that speakers engage with more participants, while greater transitivity reflects denser connections, leading to the creation of more interconnected speaker groups (Pauksztat et al., 2011).

5.3.4 Qualitative Evaluation

As a final assessment, we evaluate synthetic MPCs qualitatively. We run both a small-scale human evaluation and an "LLM as a judge" assessment (Gu et al., 2024) for analyses on a larger scale.

We ask two expert human annotators and one LLM to rate a given MPC along the following dimensions (inspired by Chen et al., 2023) using a Likert Scale from 1 to 5:

I. *Naturalness*, i.e., the quality of the overall flow, tone, and word choice in the conversation;

II. *Argumentability*, i.e., how well the conversation presents reasoned and well-argued positions;

III. *Speaker's Stance Consistency*, i.e., whether all speakers maintain the stance assigned at the beginning of the conversation;

IV. *Speaker's Stance Evolution*, i.e., whether speakers demonstrate a realistic and logical evolution of their stance during the conversation or maintain their stance consistently;

V. *Addressee Correctness*, i.e., whether the assigned addressees align with the conversation context and are logically appropriate;

VI. *Addressee Preciseness*, i.e., whether addressees are precise and contextually appropriate (messages should target the smallest relevant group of individuals).

The full annotation guidelines are reported in Appendix C.4.

5.4 Experimental settings

To generate synthetic MPCs, we compare four different instruction-based models, chosen for their comparable parameter sizes and compatibility with the same prompt design. The models include `Llama3.1-8B-Instruct` (Dubey et al., 2024), `Qwen2.5-7B-Instruct` (Yang et al., 2024), `Ministral-8B-Instruct`[3], and `OLMo-2-7B-Instruct` (OLMo et al., 2024). For each generation strategy (One-Long or Turn-by-Turn, see Section 5.2.1) we develop three distinct system prompts combining a more or less schematic task description and different examples of the output format. For details we refer to Appendix C.1. For each combination of constraints, topic and system prompt, we generate 75 conversations to account for the potential variety of structures. In total we obtained 102 600 synthetic MPCs for each model and generation strategy. An example of generated MPC is reported in Figure 5.2.

5.5 Evaluation of Compliance with Constraints

We first address RQ(3.1), aimed at assessing whether synthetic MPCs can comply with the predefined constraints described in Section 5.3.1. The results of the analysis are

[3] `https://mistral.ai/news/ministraux/`

{"**conversation**": [
{"*id*": 1, "*speaker*": "Alex", "*message*": "We need to talk about climate change. What are your thoughts?",
"*addressee*": ["Emma", "Jack", "Lily", "Mike", "Oliver"]},
{"*id*": 2, "*speaker*": "Emma", "*message*": "I think we should invest more in renewable energy sources.", "*addressee*": ["Alex"]},
{"*id*": 3, "*speaker*": "Lily", "*message*": "Climate change is a serious issue, we need to take immediate action.",
"*addressee*": ["Mike", "Oliver"]},
{"*id*": 4, "*speaker*": "Jack", "*message*": "I agree with Emma, solar and wind power are the future.", "*addressee*": ["Alex", "Emma"]},
{"*id*": 5, "*speaker*": "Mike", "*message*": "I don't think we should make drastic changes, it's too expensive.",
"*addressee*": ["Lily", "Oliver"]},
{"*id*": 6, "*speaker*": "Oliver", "*message*": "We should focus on economic growth, not environmental issues.",
"*addressee*": ["Alex", "Emma", "Lily"]},
{"*id*": 7, "*speaker*": "Alex", "*message*": "Economic growth and environmental protection are not mutually exclusive.",
"*addressee*": ["Mike", "Oliver"]},
{"*id*": 8, "*speaker*": "Emma", "*message*": "We can create jobs in the renewable energy sector.", "*addressee*": ["Jack"]},
{"*id*": 9, "*speaker*": "Lily", "*message*": "We need to educate people about the consequences of climate change.",
"*addressee*": ["Mike"]},
{"*id*": 10, "*speaker*": "Jack", "*message*": "Renewable energy is becoming more efficient and affordable.", "*addressee*": ["Emma"]},
{"*id*": 11, "*speaker*": "Mike", "*message*": "But what about the cost of transitioning to new energy sources?", "*addressee*": ["Lily"]},
{"*id*": 12, "*speaker*": "Oliver", "*message*": "We should prioritize economic growth over environmental concerns.", "*addressee*": ["Alex"]},
{"*id*": 13, "*speaker*": "Alex", "*message*": "That's a short-sighted view, Oliver.", "*addressee*": ["Oliver"]},
{"*id*": 14, "*speaker*": "Emma", "*message*": "We can't afford to wait any longer to act on climate change.", "*addressee*": ["Lily", "Jack"]},
{"*id*": 15, "*speaker*": "Lily", "*message*": "Let's work together to find a solution.", "*addressee*": ["Alex", "Emma", "Jack", "Mike", "Oliver"]}
],
"**speakers**": [
{"*name*": "Alex", "*stance*": "positive"},
{"*name*": "Emma", "*stance*": "positive"},
{"*name*": "Jack", "*stance*": "positive"},
{"*name*": "Lily", "*stance*": "positive"},
{"*name*": "Mike", "*stance*": "negative"},
{"*name*": "Oliver", "*stance*": "negative"}
],
"**topic**": "fight to climate change"
}

Figure 5.2: Example of Synthetic Multi-Party Conversation

reported in Table 5.2. We compare the output generated by the four different LLMs, each following two strategies for generation (i.e. One-Long vs. Turn-by-Turn). We report the percentage of generated MPCs, out of the 102 600 in the initial set, that were generated in compliance with the given constraint.

This first evaluation shows that `Qwen2.5` is the best model to comply with the constraints, followed by `Llama3.1`. Focusing on the best generation strategy, 77.72% of the multi-party conversations generated by the former comply with all constraints, while for `Llama3.1` this percentage drops to 66.52%. `Ministral` and `OLMo2`, instead, fail to satisfy all constraints in the vast majority of generated conversations. Concerning the generation strategy, Turn-by-Turn generation is overall better at complying with almost all the constraints.

The constraints where most settings encounter significant challenges are the *Number of Speakers* and *Stance of the Speakers*. However, Turn-by-Turn seems to be able to mitigate these issues for LLMs except `Ministral`. Based on these findings, in the remainder of this work we will focus on `Llama3.1` and `Qwen2.5` and perform all analyses on the subset of MPCs that satisfy all constraints.

Model	**Llama3.1**		**Qwen2.5**		**Ministral**		**OLMo2**	
Generation strategy	*OL*	*TT*	*OL*	*TT*	*OL*	*TT*	*OL*	*TT*
Output Format	78.97	97.00	90.78	**99.58**	15.64	35.01	0.43	91.16
Interactions	78.91	93.49	90.72	**99.52**	15.61	13.18	0.43	70.82
Number of Messages	78.93	70.25	90.66	**99.57**	15.57	13.10	0.43	71.68
Number of Speakers	29.56	97.00	39.18	**99.57**	10.22	13.04	0.21	71.88
Stance of the Speakers	19.66	**96.81**	22.95	84.03	4.42	1.04	0.09	62.11
Contribution	72.87	**95.29**	84.80	90.43	15.53	18.20	0.16	30.08
All Constraints	15.16	66.52	20.32	**77.72**	4.34	0.87	0.04	19.39

Table 5.2: Number of generated MPCs that are compliant with each constraint (percentage on the full set of 102 600 generations) for each LLM and strategy (i.e. OL = One-Long generation, TT = Turn-by-Turn generation). The final percentage of MPCs (last row) is the percentage of generations that satisfy all constraints.

5.5.1 Effects of prompt formulation

Table 5.3 and Table 5.4 report the percentage of MPCs that are compliant with each constraint, as in Section 5.5, computed over the full set of 34 200 MPCs generated for each combination of System Prompt (as presented in Appendix C.1), model, and generation strategy. The constraints definitions follow those introduced in Section 5.2.3, and the results are broken down by both generation strategy and system prompt.

As the tables show, there is no universally best-performing prompt across models or strategies. Moreover, restricting the evaluation only to the best-performing prompt does not affect the ranking of model-strategy combinations discussed in Section 5.5. By doing so we would ignore the variability introduced by different prompt formulations.

Model	**Llama3.1**						**Qwen2.5**					
Generation strategy	*OL*			*TT*			*OL*			*TT*		
System Prompt	I	II	III	I	II	III	I	II	III	I	II	III
Output Format	76.6	74.1	**86.1**	97.1	**97.3**	96.6	85.3	92.0	**95.1**	99.4	**99.7**	99.6
Interactions	76.6	74.1	**86.1**	93.5	**94.5**	92.4	85.2	91.9	**95.0**	99.3	**99.7**	99.6
Number of Messages	76.6	74.1	**86.1**	60.9	**76.9**	72.9	85.2	91.9	**94.9**	99.4	**99.7**	99.6
Number of Speaker	33.6	**34.1**	21.0	97.1	**97.3**	96.6	32.3	25.8	**59.5**	99.4	**99.7**	99.6
Stance of the Speakers	21.9	**22.4**	14.7	96.9	**97.1**	96.4	18.0	16.0	**34.9**	**86.4**	81.3	84.4
Contribution	73.0	67.4	**78.2**	94.8	**96.3**	94.7	79.9	**88.5**	86.0	89.8	**91.3**	90.2
All Constraints	**18.9**	18.0	8.6	57.1	**73.8**	68.6	15.5	14.1	**31.5**	**79.9**	75.4	77.9

Table 5.3: Percentage of generated MPCs (out of the full set of 34 200 generations) that are compliant with each constraint for each of the three prompt versions for `Llama3.1` and `Qwen2.5`, in both strategies (i.e. OL = One-Long generation, TT = Turn-by-Turn generation).

Model	**Ministral**						**OLMo2**					
Generation strategy	*OL*			*TT*			*OL*			*TT*		
System Prompt	I	II	III	I	II	III	I	II	III	I	II	III
Output Format	16.5	13.5	**16.9**	36.6	**37.7**	30.8	0.0	0.3	**1.0**	92.3	**93.3**	87.9
Interactions	16.5	13.5	**16.8**	**13.7**	12.2	13.6	0.0	0.3	**1.0**	**83.5**	67.9	61.1
Number of Messages	16.5	13.5	**16.8**	**13.7**	12.1	13.5	0.0	0.3	**1.0**	**84.3**	69.4	61.4
Number of Speakers	**14.1**	9.8	6.9	**13.6**	12.0	13.5	0.0	0.1	**0.6**	**84.0**	70.3	61.3
Stance of the Speakers	**6.1**	4.4	2.8	0.9	**1.2**	1.1	0.0	0.1	**0.2**	**75.8**	60.8	49.7
Contribution	16.5	13.4	**16.7**	**21.3**	16.6	16.8	0.0	**0.3**	0.2	27.9	28.6	**33.8**
All Constraints	**6.1**	4.3	2.7	0.7	**1.0**	0.9	0.0	**0.1**	**0.1**	**22.8**	17.8	17.6

Table 5.4: Percentage of generated MPCs (out of the full set of 34 200 generations) that are compliant with each constraint for each of the three prompt versions for `Ministral` and `OLMo2`, in both strategy (i.e. OL = One-Long generation, TT = Turn-by-Turn generation).

5.5.2 General Statistics of Final Set of Synthetic MPCs

In Figure 5.3, we report general statistics of the synthetic MPCs that satisfy the constraints described in Section 5.5. For each model–strategy combination, we present: (I.) the Empirical Cumulative Density Function (ECDF) of the average number of addressees per turn, both aggregated across all conversations and broken down by sub-categories (4, 5, and 6 users); (II.) the distribution of the number of users; (III.) the distribution of stance assignments; and (IV.) the distribution of the number of turns.

This analysis enables us to understand the characteristics and tendencies of conversations generated by each model-strategy combination. In particular, although the average number of addressees per turn is not pre-specified and the model is free to generate it, examining its distribution allows us to detect whether certain models consistently produce specific interaction patterns. Similarly, evaluating the number of users, stance assign-

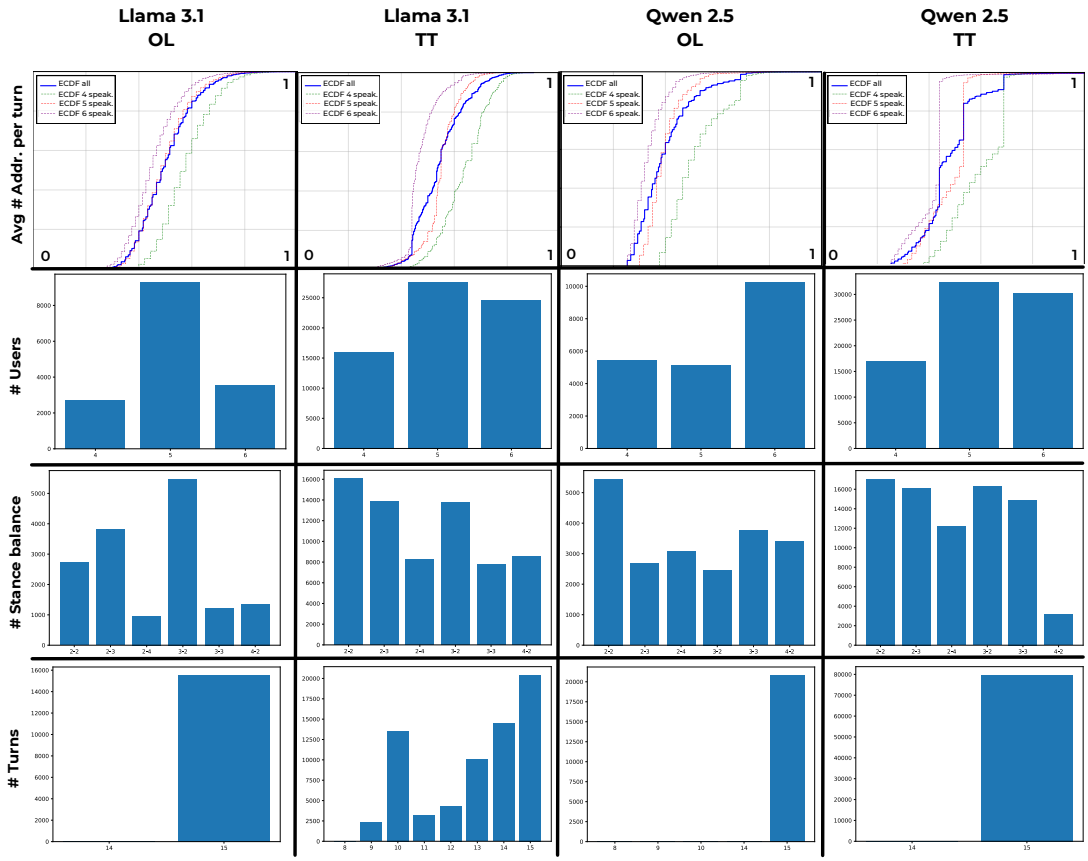

Figure 5.3: General statistics of the resulting MPC for `Llama3.1` and `Qwen2.5` on both generation strategies. The statistics reported are (from the top): (I.) average number of addressees per turn, (II.) number of users, (III.) stance assignment, (IV.) number of turns.

ments, and turns highlights potential strengths or weaknesses of each generation strategy under different constraints.

Most model–strategy combinations satisfy the strict requirement of producing exactly 15 messages. The main exception is `Llama3.1` using the Turn-by-Turn generation strategy, which frequently generates shorter conversations. For the other models, instances with lengths differing from 15 are minimal, appearing almost null in the figure. Regarding the number of users, although we requested a 6-user conversation in 50% of the prompts, the majority of generated MPCs (except those produced by `Qwen2.5`-OL) contain only 5 users. This suggests that many of the 6-user MPCs fail to meet at least one of the imposed constraints, likely due to the increased complexity of managing a larger number of participants. This interpretation is reinforced by the stance-distribution results: since each stance configuration was requested an equal number of times, the conditions with fewer valid generations correspond precisely to those involving 6 users.

Model	**Llama3.1**		**Qwen2.5**	
Gener. Strategy	*OL*	*TT*	*OL*	*TT*
Avg. # words	11.94	26.58	9.67	14.15
RepetitionRate (↓)	18.08	11.07	14.43	13.35
StringSimilarity (↓)	65.51	53.88	63.22	58.38
SemanticCoher. (↑)	0.606	0.636	0.588	0.604

Table 5.5: Results of language variability analysis.

With respect to the average number of addressees per turn, all model–strategy combinations produce a reasonably well-spread distribution across the possible values. The only exception is `Qwen2.5`-TT, which exhibits noticeable discontinuities ("jumps") in the distribution, suggesting the presence of specific recurrent interaction patterns.

5.6 Results of Language Variability

Table 5.5 summarizes the results of the analysis on language variability (as described in Section 5.3.2), aimed at assessing to what extent LLMs are able to generate MPCs that are lexically rich and consistent. The analysis shows that linguistic variability at surface level is lower when MPCs are generated in a single pass (One-Long generation) and also semantic coherence is lower compared to Turn-by-Turn generation for both models, i.e., `Llama3.1` and `Qwen2.5`. This is probably due to the fact that in Turn-by-Turn settings, the LLM is explicitly required to generate a turn by taking into account what immediately precedes it, building a coherent conversation step by step. `Llama3.1` generates less repetitive multi-party conversations at surface level, despite their turns being on average longer than `Qwen2.5`'s. Also semantic coherence is better for `Llama3.1` in all settings.

5.7 Results of Structure Analysis

We report in Figure 5.4 the results of the structural analysis for `Qwen2.5` and `Llama3.1` (with both generation strategies). Since one of our goals is to assess how synthetic MPCs compare to *real* MPCs in terms of interactional complexity, we perform the same structure analysis on our synthetic MPCs and on 13 714 MPCs extracted from the UbuntuIRC dataset (Ouchi and Tsuboi, 2016), a widely used corpus of conversations from an online forum about software issues and troubleshooting. This subset was extracted using the same strategy presented in Chapter 4 to obtain all non-overlapping conversations with 15 messages and 4, 5, or 6 speakers, ensuring that each conversation form a single connected-component (in terms of interaction graph). For each of the five network metrics introduced

in Section 5.3.3, we plot in Figure 5.4 the Empirical Cumulative Density Function (ECDF) obtained by analyzing synthetic MPCs with 4, 5 or 6 speakers (i.e. nodes) and on all generated MPCs, and we compare them with ECDF for UbuntuIRC.

For all metrics, higher values indicate more complex interactions. As shown by the median values, the UbuntuIRC dataset consistently exhibits lower values across all statistics. Compared to UbuntuIRC, speakers in our synthetic MPCs tend to interact with more participants. Also, pairs of speakers tend to have more back-and-forth dynamics and groups of speakers tend to be more interconnected. Additionally, in our dataset, the distribution of conversations with varying numbers of participants closely mirrors the overall average, with no notable deviations. This finding holds for all the model-strategy combinations and all metrics.

Regarding the average degree, almost all model–strategy combinations produce a median increase of approximately +87.5%, with the exception of `Qwen2.5`-OL, which shows a smaller increase of +67.5%. A similar pattern is observed across other network metrics, with `Qwen2.5`-OL consistently exhibiting smaller relative increases compared to the other combinations.

The transitivity metric provides particularly interesting insights. In the Ubuntu-IRC dataset, the transitivity curve saturates quickly, whereas in our generated conversations, values are pushed above 0.5. This is likely a result of allowing multiple addressees per turn, which naturally creates more triangles in the interaction network.

Despite this, multi-addressee interactions do not substantially increase the curve for consistent reciprocity. However, they do produce a more balanced distribution compared to the original Ubuntu-IRC dataset, indicating improved structural diversity in the generated conversations.

5.8 Qualitative Evaluation

The last analysis focuses on the quality of the generated conversations and is conducted both manually and automatically. Ideally, using LLM-as-a-judge would allow us to quickly evaluate all synthetic MPCs with limited effort. However, we need to assess the quality of this automatic multi-dimensional evaluation. So, we first select 96 MPCs (24 per model and generation strategy) via stratified sampling balanced across topic and stance. We then ask two human annotators with extensive experience in linguistic annotations to evaluate for each MPC the six dimensions described in Section 5.3.4, i.e.: (I.) *Naturalness*; (II.) *Argumentability*; (III.) *Speaker's Stance Consistency*; (IV.) *Speaker's Stance Evolution*; (V.) *Addressee Correctness*; (VI.) *Addressee Preciseness*.

The average values assigned to each dimension on a Likert scale between 1 (poor qual-

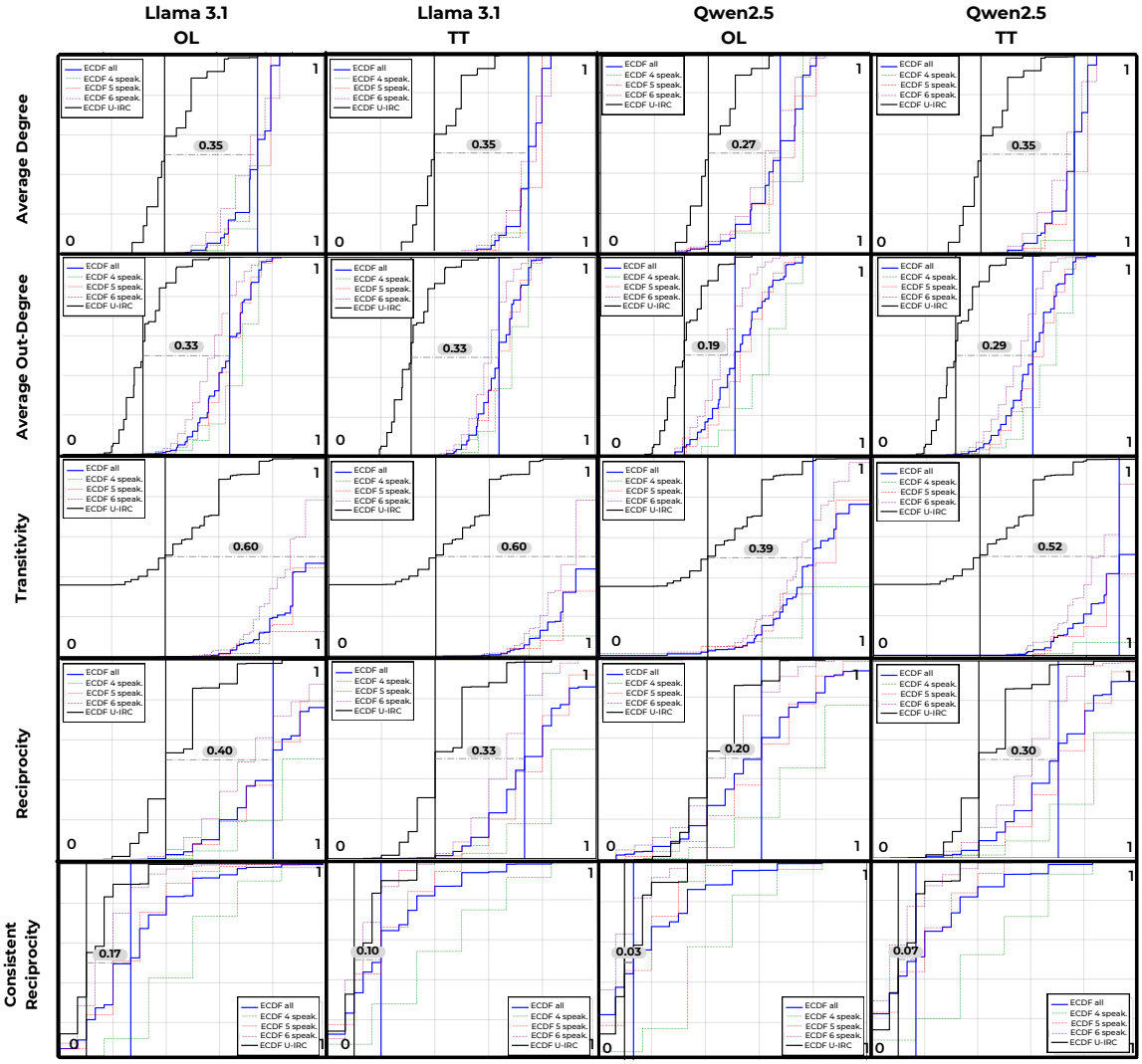

Figure 5.4: Empirical Cumulative Density Function (ECDF) of structural analysis on the synthetic MPCs from `Llama3.1` and `Qwen2.5`, with both generation strategies, i.e. One-Long and Turn-by-Turn generation. The statistics reported are (top to bottom): (I.) Average Degree Centrality, (II.) Average Out-Going Degree, (III.) Transitivity, (IV.) Reciprocity, (V.) Consistent Reciprocity. Average Degree Centrality and Average Out-Going Degree are normalized. In this way, all values on the vertical axis (density) and on the horizontal axis (value of the metric) are included between 0 and 1.

Model	**Llama3.1**		**Qwen2.5**	
Generation Strategy	*OL*	*TT*	*OL*	*TT*
Naturalness	**4.46**	4.29	4.33	4.00
Argumentability	3.98	**4.17**	3.83	3.52
Addressee Correctness	4.02	4.10	3.92	**4.21**
Addressee Preciseness	3.65	**3.94**	3.81	3.52
Stance Consistency	4.04	3.65	3.60	**4.21**
Stance Evolution	4.29	**4.73**	4.33	4.42

Table 5.6: Average results between the two human annotators on 96 MPCs (24 for each model-strategy combination).

ity) and 5 (perfect quality) on the 96 MPCs are reported in Table 5.6. We observe that all dimensions have been evaluated positively, especially *Naturalness* and *Speaker's Stance Evolution*. The most challenging dimension is *Addressee Preciseness*, which is the only dimension with an average score below 4 for all combinations. Neither of the two LLMs is consistently better and neither of the generation strategies (One-Long vs. Turn-by-Turn) is superior to the other with respect to all evaluation dimensions. The inter-annotator agreement, measured via Krippendorff's alpha (Krippendorff, 2011) and Spearman's correlation on all 96 MPCs, shows high agreement on the stance-based dimension, medium for addressee-based ones and lower for the content-based dimensions. We provide more details in Section 5.9.

We complement this manual evaluation with a large-scale automatic LLM-as-a-judge evaluation with OpenAI's o3-mini model.[4] We first assess whether it can be reliably used to evaluate all six dimensions above. We therefore launch LLM-as-a-judge on the same 96 MPCs which were manually evaluated and measure human-LLM agreement (full details in Table 5.8 in Section 5.9). While Spearman's correlation highlights a positive correlation between LLM and both human annotators on all dimensions except for *Addressee Preciseness*, Krippendorf's alpha results are less consistent. Only the *Speaker Stance Consistency*, i.e. whether the speakers comply with the assigned stance when entering the conversation, shows an extremely high agreement and correlation (Krippendorf's alpha 0.80, Spearman's correlation 0.76/0.78). We therefore carry out a large-scale evaluation only on the stance-based dimensions[5] using LLM-as-a-judge on 800 conversations (200 per model and generation strategy). Results are reported in Table 5.7 and, similar to the human evaluation, show that `Llama3.1` and `Qwen2.5` are comparable in terms of performance and that they are able to generate MPCs that present realistic evolution of

[4] `https://openai.com/index/openai-o3-mini/`

[5] We use LLM-as-a-judge also on the *Speaker's Stance Evolution*, where correlation was still highly statistical significant.

speakers' stance with both generation strategies.

Model	**Llama3.1**		**Qwen2.5**	
Generation Strategy	*OL*	*TT*	*OL*	*TT*
Stance Consistency	**4.15**	3.76	3.99	3.64
Stance Evolution	4.64	4.46	4.62	**4.68**

Table 5.7: Results with LLM as a judge on 800 MPCs (200 for each model-strategy combination) using a Likert Scale from 1 to 5.

5.9 Human and LLM-as-a-judge Agreement

As reported in Section 5.8, we compute Inter-Annotator Agreement (Krippendorf's alpha and Spearman's correlation) on a batch of 96 MPCs between the two human annotators and between each human annotator and the LLM (results in Table 5.8). Then, we compute Krippendorf's alpha among the three annotators together (the two humans and the LLM). For the score with highest Krippendorf's alpha, i.e., *Speaker Stance Consistency*, we employ LLM-as-a-judge on large scale. For completeness, we run LLM-as-a-judge also on the other stance-based dimension, i.e., *Speaker Stance Evolution*, where the correlation is still highly statistically significant, but with lower inter-annotator agreement (still positive).

The results highlight the high degree of subjectivity of some evaluation scores and provide insights into the generally low inter-annotator agreement (IAA) between LLM-as-a-judge and human annotators. Notably, all dimensions, except *Naturalness*, show statistically significant correlations between the two human annotators. Stance-based dimensions exhibit the highest IAA, followed by addressee-based and then stylistic ones. This pattern could be due to the annotators' differing backgrounds, one in philosophy, the other in computer engineering, which likely influenced their assessments of conversational quality (*Naturalness*) and argumentative strength (*Argumentability*). Agreement between LLM-as-a-judge and human annotators follows a similar trend, except for Addressee Preciseness and Stance Evolution, where agreement is markedly lower, suggesting that LLMs struggle particularly when assessing whether the set of addressees of a turn is too generic and whether the stance evolution is plausible (with respect to humans).

5.10 Discussion

The analyses from the previous sections allow us to address all three research questions.

Annotators	**H1-H2**		**H1-O3**		**H2-O3**		**H1-H2-O3**
Coefficients	*Kr.*	*Sp.*	*Kr.*	*Sp.*	*Kr.*	*Sp.*	*Kr.*
Naturalness	0.00	0.09	−0.24	0.05	0.16	0.26*	−0.01
Argumentability	0.23	0.36**	0.14	0.22*	0.29	0.31*	0.22
Addressee Correctness	0.37	0.37**	0.18	0.30*	0.30	0.42**	0.30
Addressee Preciseness	0.41	0.47**	−0.08	−0.12	−0.04	0.01	0.10
Stance Consistency	0.81	0.77**	0.78	0.76**	0.81	0.78**	0.80
Stance Evolution	0.49	0.57**	0.29	0.25*	0.27	0.42**	0.23

Table 5.8: Inter-annotator agreement (Krippendorf's alpha and Spearman's correlation) between LLM as a judge (O3) and Human experts (H1 and H2). For Spearman's correlation, (*) highlights that the correlation is statistically significant ($p < 0.05$). Instead, (**) corresponds to a correlation that is highly statistically significant ($p < 0.001$).

RQ(3.1) – Leveraging LLMs to generate large synthetic MPC datasets under constraints. With respect to RQ(3.1), targeting the possibility to generate synthetic MPCs following predefined constraints, our evaluation shows that models with comparable parameter sizes can yield very different performances. In this respect, `Qwen2.5` is by far the best performing LLM followed by `Llama3.1`. Indeed, it is able to generate 77.72% of MPCs compliant with *all* the constraints provided in the prompt. The reason behind this difference in performance cannot be clearly identified but it likely depends on the quality of pretraining data. Looking at other dimensions, however, there is no clear winner between `Qwen2.5` and `Llama3.1`. Although `Llama3.1` generates less repetitive and semantically more coherent MPCs, our qualitative evaluation does not favor either model.

RQ(3.2) – Best generation strategy. Regarding RQ(3.2), aimed at finding the best generation strategy between One-Long and Turn-by-Turn, we observe that generating MPCs in a Turn-by-Turn fashion is consistently better in terms of compliance with given constraints. This can be related to recent advancements in handling long contexts: generating shorter, multi-step outputs can be more precise and reduce errors compared to relying on a single, long-generation output. However, this advantage comes at the cost of longer computational times (in our experiments, Turn-by-Turn strategy took from 4 to 8 times more than One-Long, see Appendix C.3). Using Turn-by-Turn strategy reduces also the repetitiveness of MPCs while generating conversations that are more semantically coherent than One-Long. Our qualitative evaluation, in contrast, renders Turn-by-Turn and One-Long similarly viable.

RQ(3.3) – Evaluation pipeline and quality check To address RQ(3.3), concerning how we can effectively evaluate the quality of generated MPCs along different dimensions, we

present a framework composed by four evaluation blocks, each targeting a specific aspect of MPCs. Beside linguistic variety, coherence and qualitative dimensions such as naturalness and stance evolution, we introduce a novel assessment of the interactional structure of synthetic MPCs. We consider five network metrics and compute empirical cumulative density function to compare them with the same values calculated from real MPCs. We show that it is possible to steer the interactional structure in generated conversations, which paves the way to the large-scale creation of high-quality MPCs with much more complex interactions than what social media datasets offer.

5.11 Findings on Generating Conversations

Multi-party conversations are mostly studied based on social media data because of its abundance and accessibility. Due to platform constraints and inherently asynchronous communication, however, such datasets poorly reflect the structural diversity of natural MPCs.

In this chapter, we investigated the potential of generating varied multi-party conversations with LLMs, showing that (some) LLMs can generate MPCs that conform to structural constraints (e.g., number of speakers and their stances). Models such as `Llama3.1` and `Qwen2.5` can yield high-quality MPCs under varied constraints, both when prompted to (I.) generate the whole MPC at once or (II.) one turn at a time, given all preceding turns in context.

This makes LLMs suitable for synthesizing large-scale datasets for various types of conversations, addressing the diversity of real-world MPCs. Synthesized data can then be further leveraged to fine-tune smaller models for various discriminative tasks (e.g., next speaker or addressee prediction).

Still, this work presents some limitations. First of all, we focused only on English language, and the topics we select are typical of US-centric polarized debates such as universal healthcare, right to abortion and death penalty. It is possible that precisely because of these divisive topics, speakers in generated MPCs were able to discuss in a consistent way with respect to the assigned stance. Moreover, in this first set of experiments, we generated MPCs with 4, 5 or 6 speakers, and with a maximum length of 15 turns. It may be worth investigating whether looser constraints, allowing more or less speakers, or longer and shorter conversations, can lead to the creation of more "natural" MPCs and whether the evaluation results would still hold.

One of the main reasons behind research on synthetic data is the need to comply with privacy concerns, especially when working with conversations extracted from social media (this is a common problem that we mentioned also in Chapter 3 and Chapter 4). Actu-

ally, generating synthetic MPCs could alleviate ethical issues related to sharing personal information online. This applies in particular to conversations about sensitive topics such as abortion or death penalty, like the ones that we generate in our experiments. Still, we acknowledge that the problem is not fully solved since basically all best performing LLMs are currently trained on social media data, and synthetic MPCs could include personal data as well Li et al. (2024c). Also, the creation of synthetic MPCs is not exempt from possible negative impact, for instance when used for training malicious agents in social conversation scenarios, but such drawback is applicable to the whole research field of synthetic corpora generation for conversational AI.

To conclude, generating high quality synthetic data is a promising direction but still in development. Indeed, even if from a research point of view we can be satisfied of finding LLMs and generation approaches that generates in quick way high amounts of good quality data, still we are far from reaching the level of real-world ones. A compromise would come with a Human-in-the-loop solution. In the next and last step of this research journey, we present a Human-AI collaborative platform for constructing high quality of linearized multi-party conversations.

Chapter 6

LLMberjack: Human-AI MPC creation

In the previous chapters we first analyzed whether interactional information can be useful and when, discovering that the amount of data used for training language models on downstream classification tasks is fundamental (Chapter 3). Then we addressed the lack of structural variety in MPC datasets and the weaknesses in evaluating MPC tasks, especially when it comes to interpret model's behavior in different interactional scenarios (Chapter 4). Finally, we showed how to overcome the problem of limited data and structural variety by synthesizing large amount of MPC data with LLMs in different generation strategies.

In the meanwhile, platforms such as X, Reddit and Kialo already provide a large amount of conversations, but in the form of *reply trees*, where each root-to-leaf path can be interpreted as a linearized MPC (Derczynski et al., 2017). In such cases, each node explicitly replies to its parent as in Chapter 3 (and occasionally to earlier messages in the thread), forming a clear, hierarchical conversational flow but lacking in most cases structures with multiple addressees.

Messaging platforms like Telegram and WhatsApp, instead, present inherently linear conversations that often contain overlapping or parallel sub-dialogues, frequently with multiple implicit addressees for each message. So, while representing examples of multi-party conversations, an annotation step would still be needed to make addressees explicit and enable modeling their complex conversation structures.

LLMs could be potentially used to address the lack of MPC datasets by generating synthetic dialogues. However, as shown in Chapter 5, although some LLMs can produce high-quality synthetic dialogues, they may still struggle with the generation of complex structures with multiple speakers.

A possible solution to create linearized multi-party conversations with overlapping or

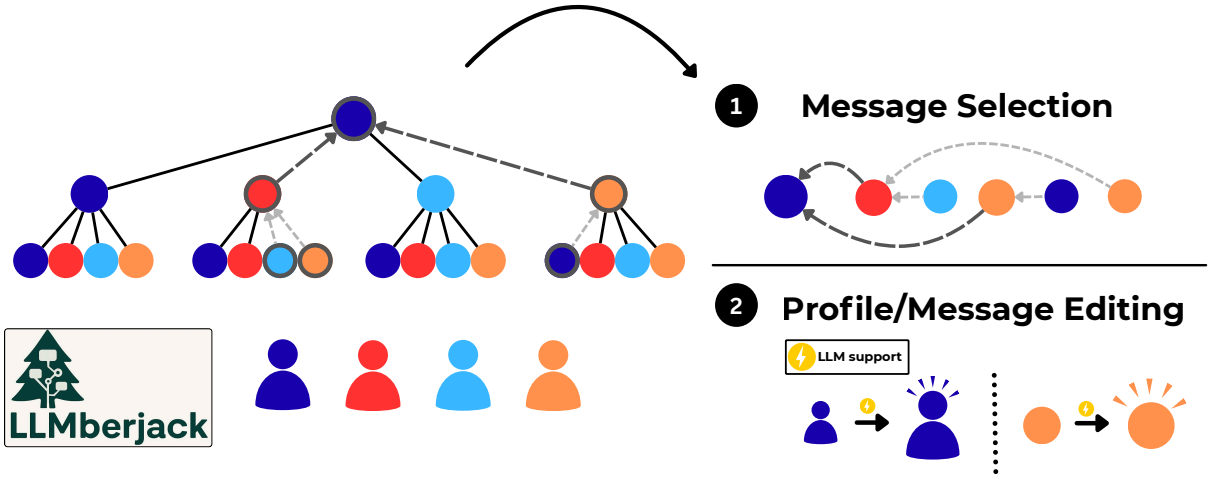

Figure 6.1: Overview of the LLMBERJACK platform. The interface integrates reply-tree visualization, message selection tools for building linearized multi-party conversations (1), and LLM-support for editing messages and speaker profiles (2).

parallel sub-dialogues starting from existing reply trees could come by "walking" on the tree following the explicit speaker–addressee relations. Human annotators could be involved in the task to modify or correct such existing conversations by editing messages or redefining addressee links, thereby enhancing both naturalness and interactional coherence. Furthermore, a single reply tree can yield several linearized MPCs, capturing potential conversation variations that result from different turn-taking orders. The task, however, would be quite complex and rather challenging for human annotators, even expert ones, if they are not provided with appropriate tools able to simplify and speed up the process.

In this chapter, we present LLMBERJACK, a Human-AI collaborative platform designed to create synthetic, thread-like multi-party conversations starting from existing reply trees. The platform provides an interface that allows annotators to "walk" through the tree, visualizing both the parent and child nodes for each message, thereby making selection decisions more context-aware.

Reply trees extracted from structured debate platforms like Kialo[1] or automatically generated may exhibit a style that is not fluent or natural enough. To enhance specific linguistic features or user traits, we implement in LLMBERJACK an LLM-assisted protocol that supports two key tasks beside tree editing: (I.) *user profiling*, i.e., the model generates a speaker profile based on the conversation content (or, in cases of limited data, from messages in the reply tree) and merges it with a pre-existing description; (II.) *mes-*

[1] https://www.kialo.com/

sage editing, i.e., the LLM refines a given message by considering the chat history and speaker profile. Human annotators then decide whether to accept, modify or reject the LLM's suggestion, ensuring the overall conversational quality.

We rigorously evaluate the impact of both tree visualization and LLM-assisted message editing involving four annotators. Results demonstrate that the quality of the resulting MPCs improves when tree visualization is available, and that LLMs can effectively support message editing, while also accelerating the annotation process.

The content of this chapter is available in the preprint *"LLMberjack: Guided Trimming of Debate Trees for Multi-Party Conversation Creation"* posted on ArXiv (Bottona et al., 2026).

LLMberjack is available on a dedicated Github repository[2]. The platform targets researchers from NLP and Social Sciences, helping them in the creation of high-quality MPCs with specific, pre-defined characteristics.

6.1 Related Work on Human-AI Synthetic Generation of Conversations

The limited availability of reliable multi-party conversation data with the desired level of structural and interactional detail suggests the need for alternative approaches. One promising direction is the use of *synthetic*, human-in-the-loop methods, which allow researchers to control conversational conditions while preserving human oversight, refinement, and interactional plausibility. This has been already tested in single-turn interactions (Fanton et al., 2021; Russo et al., 2023) and for multi-turn dialogues (Bonaldi et al., 2022; Occhipinti et al., 2024a), but not yet for multi-party settings.

Menini et al. (2025) introduced FirstAID, a platform designed to assist a human annotator in the synthetic creation of document-grounded dialogues among multiple participants, but the evaluation has been limited to 1-to-1 interactions. In literature, ConvoKit (Chang et al., 2020) is the most established toolkit for multi-party settings, which offers datasets and computational tools for the linguistic and structural analysis of multi-party conversations. Yet, despite these contributions, there is still no open-source platform that supports the *creation* with *human-AI refinement* of multi-party conversations from structured reply trees.

[2] https://github.com/LeonardoBottonaUniTn/demo_conv_creation/

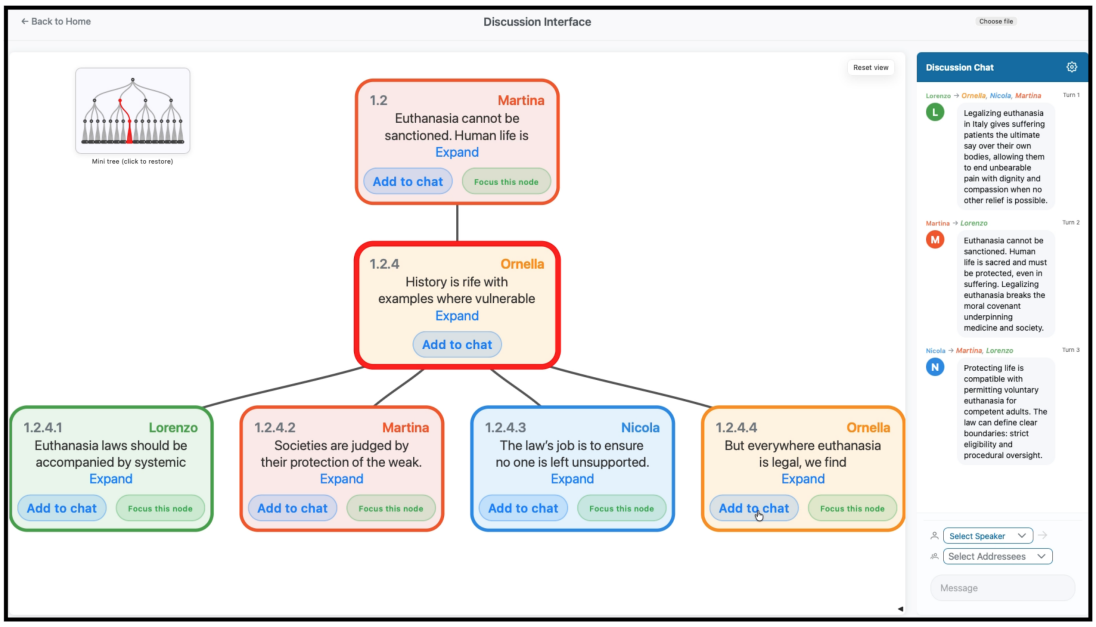

Figure 6.2: Screenshot of tree visualization for node 1.2.4 (left) and of the chat creation tab (right). Each node-box reports the speaker's name on the top-right corner, and a preview of the message in the center (expandable).

6.2 System Architecture

LLMberjack is designed to support the full workflow for transforming structured reply trees into coherent multi-party conversations. The system is organized into three main layers: (i.) a data-processing backend, (ii.) an interactive data manipulation interface, and (iii.) an export module.

6.2.1 Data Representation and Backend Processing

Tree-Centric Data Model. All discussion sources are represented as rooted reply trees. Each node corresponds to an individual message and stores author and text of message, and other existing platform-specific attributes, if any. Edges encode explicit reply-to relations.

Backend Services. The backend provides:

i. parsing routines that convert raw `json` dumps into the internal graph representation;

ii. subtree querying for efficient visualization and traversal;

III. file-management functionalities for uploading discussion files, performing LLM-assisted tree normalization when the structure is imperfect, and handling draft files containing partial or previously linearized conversations;

IV. LLM endpoints for message refinement and speaker profiling.

6.2.2 Interactive Data Manipulation Interface

The data manipulation environment is implemented using *Vue.js* and *D3.js* to provide real-time synchronization between the debate tree and the emerging linearized conversation.

Tree Visualization. Annotators are presented with an interactive view of the full debate tree featuring: (I.) a global structural visualization of the entire debate tree and a focused node view showing the selected node together with its parent and children; (II.) color-coded authors. We report a screenshot of the visualization in Figure 6.2.

This view facilitates the exploration of alternative conversational paths and supports informed linearization decisions.

Thread Construction. Annotators construct linear sequences of conversation turns by selecting messages from a given reply tree and placing them in a turn-by-turn order. The interface allows annotators to reorder messages, redefine addressee relations (for example by selecting multiple addressees for a turn) and enforce soft constraints (e.g., minimal edits, conversational plausibility).

6.2.3 LLM-Assisted Refinement Module

Speaker Profiling. Each user is associated with a speaker profile, either provided as input or assigned as a default version when unavailable (details in Appendix D.2). Upon request, the platform refines such profiles using an LLM. We use `Llama4-Maverick` (Meta, 2025) for all the LLM-assisted tasks, exploiting the `Groq` cloud platform[3] (prompts and generation details are reported in Appendix D.3). To construct or refine a profile, the model receives: (I.) the original speaker profile to be refined, and (II.) a set of selected messages from the speakers serving as contextual evidence (details in Appendix D.3). Based on this information, the LLM infers stylistic patterns and conversational temperament merging them into an updated profile.

[3] `https://groq.com/`

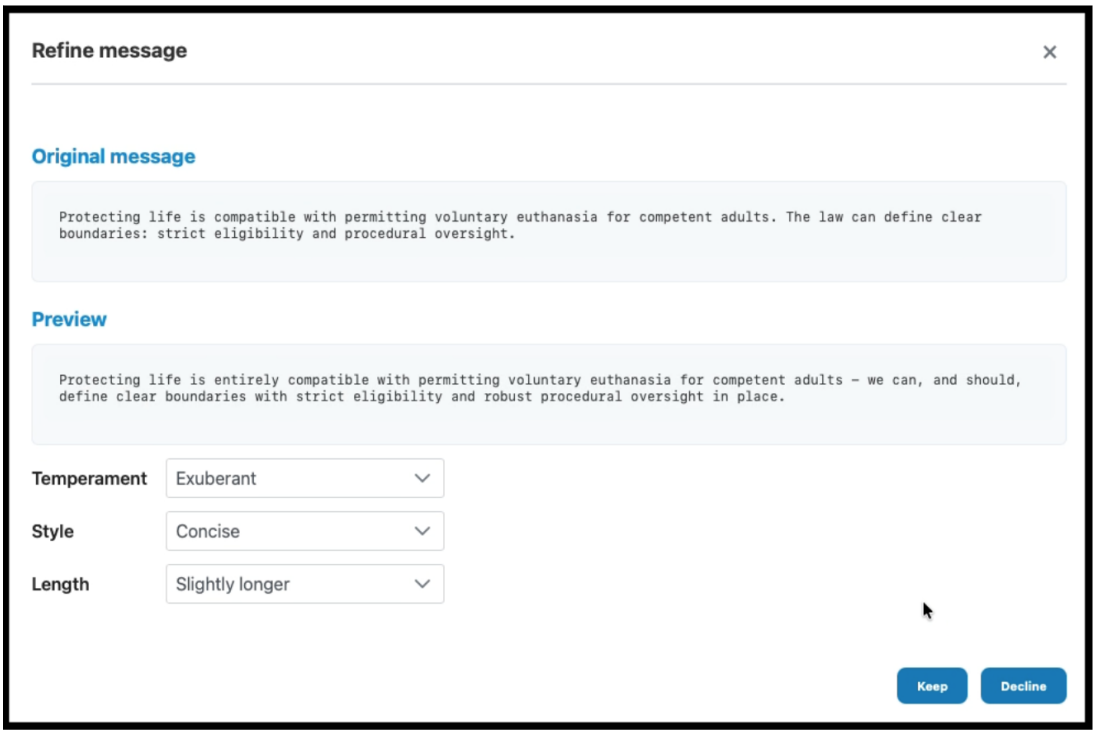

Figure 6.3: Screenshot of the LLM-assisted message refinement page.

Message Refinement. The LLM can refine individual messages under annotator's control. The model receives: (I.) the message to refine; (II.) the local conversational context, i.e., all messages appearing before the one being refined; (III.) the speaker profile; (IV.) the constraints set by the annotation protocol (style, length, temperament). Based on this information, the LLM generates a new, improved version of the original message.

The annotator can accept or modify the proposed revision, ensuring the final version remains coherent and free from hallucinations or stylistic drift. We report a screenshot of the message refinement page in Figure 6.3.

6.2.4 Data Export, Deployment and Availability

The system provides `json` export for the final conversations. The full platform is publicly available as open-source software via the dedicated GitHub repository. It can be deployed *locally*, for secure or small-scale dataset creation tasks. The repository includes installation scripts, configuration templates, and a demo instance, facilitating adaptation to diverse annotation protocols and datasets.

6.3 Evaluation of LLMberjack Features

We evaluated through human assessment the impact of two core features of LLMberjack: the impact of tree visualization on the creation of multi-party conversations and the LLM-assisted message editing. To isolate their effects, we split the analysis into two parts. Firstly, we performed the message selection task, starting from a reply tree, comparing conditions with and without tree visualization. Secondly, we edited a subset of messages, comparing scenarios with and without LLM support.

6.3.1 Creation of synthetic Reply-trees

As a preliminary step for our evaluation, we first generate synthetic reply-trees. Specifically, we first ask the LLM to define a set of m users and then generate iteratively the full debate tree. The process starts with one single initial message from a random user (i.e., the root of the discussion), followed by one reply from each participant (including the self-replies). This procedure is repeated recursively for each new node up to a specified depth d, resulting in a total of $n = \sum_{i=0}^{d-1} m^i$ messages. LLMs are generally proficient at producing coherent one-to-one replies that respect user profiles or conversational roles. From these generated reply trees, we make the annotators build linearized multi-party conversations.

We generated 4 synthetic reply trees using `GPT4.1`[4], each representing a complete debate with a depth of 4, and with exactly 4 users. Each reply tree is about a different topic. In each tree, every node receives exactly 4 replies (one from each user, including self-replies). For each topic, two speakers were assigned a pro stance and two a counter stance with respect to the topic. The selected topics are for the synthetic reply trees are:

T_1: Legalization of marijuana in Italy.

T_2: Legalization of euthanasia in Italy.

T_3: Introduction of a four-day work week in Italy.

T_4: Serie A clubs should promote more Italian players rather than foreign stars.

Since the annotators were Italian, these topics were chosen to reflect debates that are salient within the Italian sociopolitical context. Additionally, we generated a fifth synthetic reply tree on the topic *"Coca-Cola is better than Fanta"*. This tree, along with the multi-party conversations derived from it, was used as tutorial material to familiarize

[4]platform.openai.com/docs/models/gpt-4.1

annotators with the platform and the tasks, thereby minimizing platform-related issues during the actual annotation process.

The evaluation process consisted of two main steps: (I.) selection of messages to build a MPC starting from the synthetic reply tree, with and without tree visualization (Section 6.3.2); (II.) editing of the resulting MPC messages, with and without LLM support (Section 6.3.3).

6.3.2 Evaluating the Impact of Tree Visualization

Annotators were asked to create a multi-party conversation from a given synthetic reply tree by selecting a subset of nodes/messages. They were instructed to follow the rules below.

R_1: The opening message must be a general statement on the given topic addressed to everyone.

R_2: Each conversation should contain between 10 and 15 messages and should resemble the style of a typical Telegram chat.

R_3: Annotators may change or add addressee relations at their discretion but all users must contribute at least one turn. The tree structure serves as a suggestion rather than a strict constraint.

R_4: Annotators should perform only minimal, necessary edits, e.g., to correct errors or ensure conversational flow. Messages should not be edited to improve style or argumentative quality, which will be part of the second evaluation step (Section 6.3.3).

Annotators were asked to create 3 distinct MPCs from each tree, aiming for variation in content and interaction patterns across conversations.

Before starting the main task, each annotator was instructed to read all speakers' profiles carefully. Annotators completed the task under two different visualization conditions: option *w Tree*, which provided full access to the tree-structure visualization during MPC creation, and option *w/o Tree*, which presented all the messages as a single flat sequence without tree visualization. For each of the four synthetic reply trees, two annotators performed the task *w Tree* visualization and two *w/o Tree* visualization.

The assignment of tree–visualization pairs was counterbalanced across annotators so that all possible combinations were covered. This design reduces potential topic effects during evaluation and helps identify topics that may be inherently more challenging, while also minimizing annotator-specific variance in the quality assessment.

After the MPCs were created, two independent evaluators assessed their quality through pairwise comparisons of sets of conversations produced from the same reply tree, created *w Tree* or *w/o Tree* visualization, for a total of 16 pairs. Each comparison was performed along three dimensions:

1. **Naturalness of the conversation**, focusing on the coherence of the conversational flow, the plausibility of turn-by-turn progression, and the overall smoothness of the dialogue.

2. **Conversation Variability**, assessing whether the set of conversations derived from the same tree exhibited meaningful diversity in content, interaction patterns, and turn-taking structure.

3. **Participants' Engagement**, evaluating the degree to which the conversation goes beyond generic statements and displays targeted, socially meaningful exchanges. This includes the presence of distinctive interactional behaviors, user-specific styles, responsive turns that directly engage with previous messages, and interactional patterns that make the dialogue feel lively, purposeful, and contextually grounded.

For each dimension, evaluators indicated which conversation in the given pair they considered of higher-quality or whether the two were equivalent.

Quantitative Evaluation. In Table 6.1, we report evaluation results for the 3 dimensions above, the average turn-selection speed in terms of turns/minute (v_{turn}) and the inter annotator agreement using Weighted Cohen's kappa (κ_w). The results show an advantage for the *w Tree* condition over the *w/o Tree* setting (only for the Variability dimension there is a relative majority of ties). This advantage is particularly pronounced for the Naturalness dimension. Furthermore, the average speed increases by almost 25% *w Tree* visualization. Agreement ranges from 0.25 to 0.44, highlighting the subjectivity of the annotation task.

Qualitative Observations. We also collected all the feedback and comments provided by the evaluators during the sessions. They reported that conversations created with tree visualization tended to focus on fewer subtopics but developed them more deeply, exhibiting richer argumentative structure and stronger relational coherence across messages. On the contrary, conversations produced without tree visualization typically covered a broader range of aspects of the main topic but remained more superficial in their argumentative depth. In general, they confirmed the difficulty in identifying a version of higher quality

	Nat.	**Var.**	**Eng.**	$\mathbf{v_{turn}}$
w Tree	*65.62*	34.37	*49.99*	*1.82*
w/o Tree	28.13	21.88	28.13	1.46
tie	6.25	*43.75*	21.88	/
$\kappa_{\mathbf{w}}$	0.44	0.40	0.25	/

Table 6.1: Percentage of MPC comparisons where one setting (with or without tree visualization) was preferred over the others in terms of naturalness (Nat.), variability (Var.), and participants' engagement (Eng.). The last column reports the average turn-selection speed in turns/minute (v_{turn}). The final row shows inter-annotator agreement (weighted Cohen's κ_w).

than the other, since quality was generally high among all the given conversations. Annotators consistently reported that the tree visualization was substantially more helpful for the task. They appreciated the implicit "guidance" it provided, allowing them to make more confident and reliable choices, particularly about choosing the addressee(s). Annotators noted that the visualization would be even more advantageous in larger annotation rounds (more than 3), as it facilitates the identification of multiple plausible MPCs through different traverses from the same debate tree and reduces cognitive effort during the task.

6.3.3 Evaluating the Impact of LLM Support

In the second evaluation step, we aimed to assess the effect of LLM-assisted message editing compared to the editing without LLM support. 4 annotators refined a total of 8 conversations (two conversations for each topic). For each annotator–topic combination, one conversation was edited with LLM assistance and the other without. All four annotators worked on every conversation, and for each conversation, two used LLM support while the other two performed the task manually.

To ensure a controlled experimental setup and avoid fully rewriting the given conversations, each annotator was instructed to focus only on one speaker and to edit, if needed, only his/her messages throughout a given conversation. The editing should specifically involve *style*, *temperament*, and *length*.

After the MPCs were edited, two evaluators assessed their quality by comparing, for the same non-edited MPC, the conversations edited *w LLM* assistance against the versions refined without it (*w/o LLM*), for a total of 32 pairs. Each pair of conversations was evaluated along two dimensions:

1. **General turn quality**, considering both the coherence of each turn and its contribution to the conversation flow;

2. **Adherence to the refinement requirements**, evaluated across the three specified sub-dimensions: *length*, *temperament*, and *style*.

For each dimension, annotators indicated whether the *w LLM* support or *w/o LLM* editing was of better quality, or whether the two versions were considered equivalent.

The assignment of LLM-assisted versus non-assisted refinement was carefully counterbalanced: two couple of annotators (forming one pair) never used the LLM on the same conversation, while the other four possible annotator pairs shared the same setting in exactly half of the cases. This design allows us to evaluate the effect of LLM assistance while controlling for annotator-specific effects and overlapping refinements.

Each annotator refined two conversations for each given topic in a fixed order: first *without* LLM assistance, and then *with* LLM assistance followed by minimal human adjustments. This ordering was chosen to avoid potential bias introduced by prior exposure to LLM-refined content.

The platform has been designed to limit the annotators freedom on three dimension, with 5 options each:

1. **Length:** much shorter, slightly shorter, same length, slightly longer, much longer;
2. **Style:** sarcastic, aggressive, exuberant, cynic, detached;
3. **Temperament**: neutral, informal, expressive, concise, formal.

We asked the annotators to modify the message of only one precise speaker for each topic, and the same speaker for both MPCs from the same topic. Respectively we asked to make messages more: (I.) *aggressive*, *informal* and *much longer* for T_1; (II.) *exuberant*, *expressive* and *same length* for T_2; (III.) *cynical*, *concise* and *slightly shorter* for T_3; (IV.) *detached*, *formal* and *slightly longer* for T_4. The combination *sarcastic*, *neutral* and *much shorter* was used as "tutorial" setting.

Quantitative Evaluation. In Table 6.2, we report the evaluation results together with the average refinement velocity in terms of tokens[5]/second (v_{tokens}). Overall, the results show a clear advantage for the *w LLM* condition compared to the *w/o LLM* setting. The average refinement velocity indicates that LLM support speeds up the refinement process by approximately 83%. Agreement ranges from 0.36 to 0.58, highlighting also here the subjectivity of the annotation task, except for the Length dimension (which is intuitively more objective).

[5]Number of tokens of the final refined sentence

	Gen.	**Len.**	**Style**	**Temp.**	$\mathbf{v_{tokens}}$
w LLM	*64.06*	*57.81*	*64.06*	*56.25*	*0.86*
w/o LLM	17.19	4.69	25.00	31.25	0.47
tie	18.75	37.50	10.94	12.50	/
$\kappa_{\mathbf{w}}$	0.36	0.58	0.43	0.44	/

Table 6.2: Percentage of times one setting (with or without LLM support) was preferred over the other in terms of general quality (Gen.), length (Len.), style (Style), and temperament (Temp.), together with the average refinement speed in tokens/second (v_{tokens}). The final row reports inter-annotator agreement (weighted Cohen's κ_w).

Qualitative Observations. Feedback from the evaluators confirmed that annotators' experience with linguistic tasks has an important impact on the quality of refinements, regardless of whether LLM assistance is provided, particularly for dimensions such as *style* and *temperament*. Nonetheless, they consistently noted that LLM support is crucial when generating substantially longer messages, where manual refinement alone is often more challenging. Annotators agreed that the LLM is particularly helpful for reorganizing sentences rather than making minor additions or deletions, a crucial aspect for longer messages. At the same time, they noted that the LLM occasionally introduces repetitive interjections; still, with minimal human editing, these issues can be easily fixed.

6.4 Findings on LLMberjack

In this chapter, we presented LLMberjack, a Human-AI collaborative platform designed to generate synthetic thread-like multi-party conversations starting from tree-structured debates, with optional LLM support for message refinement. Our goal was to alleviate the scarcity of high-quality MPC datasets with well-controlled interactional and structural properties by providing annotators with an intuitive interface that supports more guided and more consistent decision-making. Our evaluations demonstrate that the platform effectively accelerates the overall creation workflow, both in message selection and in refinement, while also leading to conversations of higher quality.

This work concludes our research journey into the challenges that natural language processing tools present when dealing with multi-party conversations, with a particular emphasis on the interactional aspects. While the contributions of this thesis address several open questions, they also reveal new directions, challenges, and discussions that need to be explored. In the next and final chapter, we summarize the main findings of this work and outline possible future steps for research in this area, highlighting promising paths for advancing the modeling, evaluation, and generation of multi-party conversations.

Chapter 7

Conclusion

Multi-party conversations represent a particularly challenging and critical scenario in which future LLMs are likely to be employed extensively. While current LLMs have demonstrated strong performance across a wide range of NLP tasks, our findings throughout this thesis show that their effectiveness in multi-party conversational settings cannot be taken for granted. Beyond the increased linguistic complexity, multi-party conversations are characterized by rich interactional structures, multiple participants, and evolving conversational dynamics that add further dimensions to both modeling and evaluation. Moreover, the challenge is not only technical but also human-related: reliably assessing the naturalness, coherence, and pragmatic plausibility of a multi-party conversation and capturing human spontaneity in a way that is both realistic and controllable remains difficult. As highlighted by our experimental results, limitations in existing datasets and evaluation protocols further complicate this scenario, motivating the need for new methodologies to generate, analyze, and assess multi-party conversational data in a reproducible manner.

Need for large and diverse datasets. In Chapter 3 and Chapter 4, we showed that both the quantity of training data and its diversity in terms of structural complexity are fundamental for developing effective MPC models and for evaluating their performance in a reliable manner. In particular, adequate data coverage is required not only to achieve strong overall results, but also to meaningfully describe model behavior across conversations exhibiting different interactional patterns and varying levels of structural complexity. From the broader perspective of this thesis, these findings underscore that insufficient data or limited structural variety can lead to misleading conclusions about model capabilities: models may appear effective on average while failing in more complex or less common conversational scenarios. Moreover, our experiments highlight that using structural information and datasets of sufficient dimensions for training stabilizes performance across conversations with different numbers of turns and participants, demonstrating that

interaction-aware models benefit from the injection of textual and structural context. Consequently, a key conclusion is that both data quantity and structural representativeness are crucial not only for performance optimization but also for fair, informative, and generalizable evaluation of MPC systems. This insight motivates the subsequent work in the thesis on generating high-quality synthetic MPCs to supplement real-world datasets and ensure adequate coverage of diverse conversational structures.

Fair evaluation protocols. Again in Chapters 3 and 4, we demonstrated that relying solely on macro-level accuracy metrics obscures important variations in model performance and can lead to misleading conclusions about system effectiveness. In particular, relying only on aggregated evaluation scores mask the fact that models tend to behave very differently across conversations with varying structural complexity. As shown throughout the thesis, models that appear competitive under global metrics often struggle in conversations characterized by several turns, higher numbers of participants, or more complex interaction patterns, while the same models may perform well in simpler discussions.

From the broader perspective of this thesis, these findings highlight a fundamental limitation of current evaluation practices in MPC research: robustness cannot be reliably inferred from averaged scores alone. Without stratifying results by interactional properties, even with simple metrics like degree centrality, it becomes almost impossible to properly identify where contextual modeling truly provides benefits and where it fails. This limitation is particularly critical when assessing LLM-based systems, whose performance we showed to be highly sensitive to both prompt formulation and conversational structure.

Consequently, one of the key conclusions of this work is that fair evaluation of MPC systems must go beyond macro-level metrics and explicitly account for structural heterogeneity within datasets. Only by adopting diagnostic, structure-aware evaluation protocols we can meaningfully describe model strengths and weaknesses, and design models that generalize across the full spectrum of real-world multi-party conversational scenarios.

Creating data. To address the issues outlined above, in Chapter 5 we investigate the use of LLMs to rapidly generate large-scale synthetic datasets of multi-party conversations. Through a systematic comparison of multiple LLMs and generation strategies, we show that some model-strategy combinations are actually able to generate conversations that satisfy strict structural constraints, such as a predefined number of turns, number of participants, stance configuration, while showing at the same time varying addressee patterns, at a high rate. In particular, `Llama3.1` and `Qwen2.5` showed to be quite reliable models, especially in Turn-by-Turn generation strategy in terms of constraint compliance.

Also as regards linguistic variability, the Turn-by-Turn approach generated better conversations with both models. Instead, in terms of interactional structures, all model-strategy combinations (involving `Llama3.1` and `Qwen2.5`) generated conversations that, according to the structural analyses performed, provide more varied interaction patterns, with more "interesting" dynamics (like back-and-forth exchanges), compared to one of the widely used datasets for addressee-related tasks.

Despite these encouraging quantitative results, our experiments also reveal some limitations. Human evaluation indicates that most generated conversations are structurally valid, and their perceived coherence, pragmatic consistency, and degree of naturalness quite good (average over 3.5) but still not enough to be considered a valid replacement of natural conversations for plausible fine-tuning. Annotators often disagree on quality judgments, highlighting the inherent subjectivity and difficulty of manually evaluating MPC quality. These findings suggest that fully automatic generation alone is insufficient to guarantee high-quality conversational data, even when formal constraints are met.

To mitigate these limitations, in Chapter 6 we introduced LLMberjack, a Human–AI collaborative platform designed to support the creation of high-quality MPCs by combining structured human supervision with LLM-assisted generation and refinement. Experimental results from user studies showed that providing annotators with explicit visualizations of conversation structures (in the shape of reply trees) significantly improves their ability to reason about interactional dynamics and identify problematic turns. Moreover, LLM-assisted refinement of messages and speaker profiles reduced manual editing effort while preserving conversational coherence, leading to MPCs that are judged more realistic and consistent with respect to fully manual refinement. Overall, these results demonstrate that a hybrid Human–AI approach offers a more effective and controllable pathway for constructing synthetic MPC datasets than purely automated generation.

From an ethical and reproducibility perspective, synthetic MPCs offer important advantages over real-world social media data. They reduce privacy risks by avoiding the use of personal or sensitive user information, and they facilitate data sharing and reproducibility by enabling the release of fully open datasets that are not subject to platform policies, data deletions, or consent constraints. At the same time, our findings highlight that synthetic data should not be treated as a direct substitute for naturally occurring conversations: careful evaluation protocols and human oversight remain essential to prevent artifacts, biases, or overly regular interactional patterns from being introduced into downstream models.

Future challenges. Building on the findings of this thesis, we outline three key directions for future research that are crucial for advancing the modeling, evaluation, and generation

of multi-party conversations.

The first issue concerns the formal modeling of written multi-party conversations. In many social media platforms, conversational structure is partially imposed by platform constraints, often resulting in explicit one-to-one reply links while leaving addressees implicit or ambiguous. As a consequence, the observable textual sequence alone does not uniquely determine the underlying interaction structure. This ambiguity is further amplified in settings involving third-party annotation or synthetic data generation, where multiple interactional configurations may be equally plausible for the same set of messages. Still, different structural interpretations of the same textual content can all be coherent, but at the same time they can induce substantially different pragmatic meanings and discourse dynamics. Such problem poses a significant challenge for both modeling and evaluation. From a modeling perspective, it raises the question of which interactional assumptions should be encoded or learned by the system. From an evaluation perspective, it undermines the notion of a single "correct" conversational structure, making standard accuracy-based metrics insufficient or even misleading. Models may be penalized for producing alternative, yet valid, interpretations of conversational structure, while in evaluation benchmarks such data ambiguity risks to flag false errors. This highlights the need for evaluation protocols that explicitly account for structural variability and ambiguity, rather than assuming a fully specified and unambiguous conversational ground truth.

The second issue concerns how to effectively improve the synthetic generation of multi-party conversations to approach human-level quality and naturalness. In this thesis, we address this challenge by designing a Human-in-the-Loop platform that supports the creation and refinement of MPCs through structured human supervision combined with LLM-assisted generation. While this approach represents a meaningful step toward higher-quality synthetic data, it also exposes open challenges related to the fair and scalable evaluation of large volumes of Human–AI generated conversations. In particular, human assessment remains costly, subjective, and difficult to standardize across annotators and conversational settings. Looking forward, further improvements may arise from the integration of self-improving agent-based generation frameworks, in which models iteratively critique, revise, and adapt generated conversations, potentially reducing manual effort (Yuksekgonul et al., 2025; Wu et al., 2025).

The third issue concerns the effective use of synthetic data to fine-tune smaller models for multi-party conversation tasks. Indeed, generated conversations may exhibit systematic regularities, stylistic homogenization, or interactional biases introduced by the prompting formulation or the underlying LLM/training data, potentially leading smaller models to overfit synthetic patterns that do not fully reflect real-world conversational dynamics (Veselovsky et al., 2023; Møller et al., 2024). This distribution shift between

synthetic and naturally occurring multi-party conversations poses a significant challenge for generalization, particularly for tasks that are sensitive to interactional aspects.

From a broader perspective, addressing these issues is also central for promoting green NLP practices and responsible model deployment. While large-scale LLMs have proven highly effective across a wide range of tasks, their environmental impact grows substantially with model size, as their carbon footprint is strongly correlated with the number of parameters (Li et al., 2024a). Leveraging synthetic MPC data to train smaller, more efficient models offers a promising direction to mitigate this impact by reducing both training and inference costs. In addition to lowering energy consumption, smaller models are easier to reproduce, distribute, and deploy, thereby improving scalability and accessibility across research and real-world applications.

To conclude. Multi-party conversations are a complex and important domain, with strong implications for everyday communication and online social life. Through its empirical findings, this thesis shows that, while meaningful progress has been made, much work remains to be done. Many key questions about how to model, evaluate, and generate multi-party conversations are still open. Addressing these questions is not only a technical challenge, but also a social one, as future language technologies will increasingly influence public discussions, social interactions, and the way humans and AI communicate at scale.

Bibliography

Vibhor Agarwal, Sagar Joglekar, Anthony P Young, and Nishanth Sastry. 2022. Graphnli: A graph-based natural language inference model for polarity prediction in online debates. In *Proceedings of the ACM Web Conference 2022*, pages 2729–2737.

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2623–2631.

Atijit Anuchitanukul, Julia Ive, and Lucia Specia. 2022. Revisiting contextual toxicity detection in conversations. *J. Data and Information Quality*, 15(1).

Marianna Apidianaki. 2023. From word types to tokens and back: A survey of approaches to word meaning representation and interpretation. *Computational Linguistics*, 49(2):465–523.

Pablo Aragón, Vicenç Gómez, and Andreaks Kaltenbrunner. 2017. To thread or not to thread: The impact of conversation threading on online discussion. *Proceedings of the International AAAI Conference on Web and Social Media*, 11(1):12–21.

Selene Baez Santamaria, Helena Gomez Adorno, and Ilia Markov. 2024. Contextualized graph representations for generating counter-narratives against hate speech. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 7664–7674, Miami, Florida, USA. Association for Computational Linguistics.

Jun Bai, Xiaofeng Zhang, Chen Li, Hanhua Hong, Xi Xu, Chenghua Lin, and Wenge Rong. 2023. How to determine the most powerful pre-trained language model without brute force fine-tuning? an empirical survey. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5369–5382, Singapore. Association for Computational Linguistics.

Jianzhu Bao, Rui Wang, Yasheng Wang, Aixin Sun, Yitong Li, Fei Mi, and Ruifeng Xu. 2023. A synthetic data generation framework for grounded dialogues. In *Proceedings*

of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 10866–10882, Toronto, Canada. Association for Computational Linguistics.

Jason Baumgartner, Savvas Zannettou, Brian Keegan, Megan Squire, and Jeremy Blackburn. 2020. The pushshift reddit dataset. *Preprint*, arXiv:2001.08435.

Tilman Beck, Andreas Waldis, and Iryna Gurevych. 2023. Robust integration of contextual information for cross-target stance detection. In *Proceedings of the 12th Joint Conference on Lexical and Computational Semantics (*SEM 2023)*, pages 494–511, Toronto, Canada. Association for Computational Linguistics.

Nicola Bertoldi, Mauro Cettolo, and Marcello Federico. 2013. Cache-based online adaptation for machine translation enhanced computer assisted translation. In *Proceedings of Machine Translation Summit XIV: Papers*, Nice, France.

Helena Bonaldi, Sara Dellantonio, Serra Sinem Tekiroğlu, and Marco Guerini. 2022. Human-machine collaboration approaches to build a dialogue dataset for hate speech countering. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8031–8049, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Carlo Bonferroni. 1936. Teoria statistica delle classi e calcolo delle probabilita. *Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commericiali di Firenze*, 8:3–62.

Leonardo Bottona, Nicolò Penzo, Bruno Lepri, Marco Guerini, and Sara Tonelli. 2026. Llmberjack: Guided trimming of debate trees for multi-party conversation creation. *Preprint*, arXiv:2601.04135.

Holly Branigan. 2006. Perspectives on multi-party dialogue. *Research on Language and Computation*, 4:153–177.

Simon Martin Breum, Daniel Vædele Egdal, Victor Gram Mortensen, Anders Giovanni Møller, and Luca Maria Aiello. 2024. The persuasive power of large language models. *Proceedings of the International AAAI Conference on Web and Social Media*, 18(1):152–163.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language

models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Guido Caldarelli, Rocco De Nicola, Fabio Del Vigna, Marinella Petrocchi, and Fabio Saracco. 2020. The role of bot squads in the political propaganda on twitter. *Communications Physics*, 3(1):81.

Jean Carletta, Simone Ashby, Sebastien Bourban, Mike Flynn, Mael Guillemot, Thomas Hain, Jaroslav Kadlec, Vasilis Karaiskos, Wessel Kraaij, Melissa Kronenthal, Guillaume Lathoud, Mike Lincoln, Agnes Lisowska, Iain McCowan, Wilfried Post, Dennis Reidsma, and Pierre Wellner. 2005. The ami meeting corpus: a pre-announcement. In *Proceedings of the Second International Conference on Machine Learning for Multimodal Interaction*, MLMI'05, page 28–39, Berlin, Heidelberg. Springer-Verlag.

Galo Castillo-López, Gael de Chalendar, and Nasredine Semmar. 2025. A survey of recent advances on turn-taking modeling in spoken dialogue systems. In *Proceedings of the 15th International Workshop on Spoken Dialogue Systems Technology*, pages 254–271, Bilbao, Spain. Association for Computational Linguistics.

Santiago Castro. 2017. Fast Krippendorff: Fast computation of Krippendorff's alpha agreement measure. `https://github.com/pln-fing-udelar/fast-krippendorff`.

Souvic Chakraborty, Parag Dutta, Sumegh Roychowdhury, and Animesh Mukherjee. 2022. CRUSH: Contextually regularized and user anchored self-supervised hate speech detection. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1874–1886, Seattle, United States. Association for Computational Linguistics.

Jonathan P. Chang, Caleb Chiam, Liye Fu, Andrew Wang, Justine Zhang, and Cristian Danescu-Niculescu-Mizil. 2020. ConvoKit: A toolkit for the analysis of conversations. In *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 57–60, 1st virtual meeting. Association for Computational Linguistics.

Jonathan P. Chang and Cristian Danescu-Niculescu-Mizil. 2019. Trouble on the horizon: Forecasting the derailment of online conversations as they develop. 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 4743–4754, Hong Kong, China. Association for Computational Linguistics.

Serina Chang, Alicja Chaszczewicz, Emma Wang, Maya Josifovska, Emma Pierson, and Jure Leskovec. 2024. Llms generate structurally realistic social networks but overestimate political homophily. *Preprint*, arXiv:2408.16629.

Tyler A. Chang and Benjamin K. Bergen. 2024. Language model behavior: A comprehensive survey. *Computational Linguistics*, 50(1):293–350.

Kushal Chawla, Weiyan Shi, Jingwen Zhang, Gale Lucas, Zhou Yu, and Jonathan Gratch. 2023. Social influence dialogue systems: A survey of datasets and models for social influence tasks. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 750–766, Dubrovnik, Croatia. Association for Computational Linguistics.

Jiaao Chen and Diyi Yang. 2021. Structure-aware abstractive conversation summarization via discourse and action graphs. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1380–1391, Online. Association for Computational Linguistics.

Maximillian Chen, Alexandros Papangelis, Chenyang Tao, Seokhwan Kim, Andy Rosenbaum, Yang Liu, Zhou Yu, and Dilek Hakkani-Tur. 2023. PLACES: Prompting language models for social conversation synthesis. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 844–868, Dubrovnik, Croatia. Association for Computational Linguistics.

Ming-Bin Chen, Lea Frermann, and Jey Han Lau. 2025. WHoW: A cross-domain approach for analysing conversation moderation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2091–2126, Albuquerque, New Mexico. Association for Computational Linguistics.

Ta-Chung Chi and Alexander Rudnicky. 2022. Structured dialogue discourse parsing. In *Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 325–335, Edinburgh, UK. Association for Computational Linguistics.

Minje Choi, Luca Maria Aiello, Krisztián Zsolt Varga, and Daniele Quercia. 2020. Ten social dimensions of conversations and relationships. In *Proceedings of The Web Conference 2020*, WWW '20, page 1514–1525, New York, NY, USA. Association for Computing Machinery.

Alessandra Teresa Cignarella, Mirko Lai, Cristina Bosco, Viviana Patti, Rosso Paolo, and 1 others. 2020. Sardistance@ evalita2020: Overview of the task on stance detection in italian tweets. In *Proceedings of the Seventh Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2020)*, pages 1–10. Ceur.

Peter Cogan, Matthew Andrews, Milan Bradonjic, W. Sean Kennedy, Alessandra Sala, and Gabriel Tucci. 2012. Reconstruction and analysis of twitter conversation graphs. In *Proceedings of the First ACM International Workshop on Hot Topics on Interdisciplinary Social Networks Research*, HotSocial '12, page 25–31, New York, NY, USA. Association for Computing Machinery.

Mauro Coletto, Kiran Garimella, Aristides Gionis, and Claudio Lucchese. 2017a. Automatic controversy detection in social media: A content-independent motif-based approach. *Online Social Networks and Media*, 3-4:22–31.

Mauro Coletto, Kiran Garimella, Aristides Gionis, and Claudio Lucchese. 2017b. A motif-based approach for identifying controversy. In *Proceedings of the international AAAI conference on web and social media*, volume 11, pages 496–499.

Cristian Danescu-Niculescu-Mizil and Lillian Lee. 2011. Chameleons in imagined conversations: A new approach to understanding coordination of linguistic style in dialogs. In *Proceedings of the 2nd Workshop on Cognitive Modeling and Computational Linguistics*, pages 76–87, Portland, Oregon, USA. Association for Computational Linguistics.

Eustasio Del Barrio, Juan A Cuesta-Albertos, and Carlos Matrán. 2018. An optimal transportation approach for assessing almost stochastic order. *The Mathematics of the Uncertain: A Tribute to Pedro Gil*, pages 33–44.

Leon Derczynski, Kalina Bontcheva, Maria Liakata, Rob Procter, Geraldine Wong Sak Hoi, and Arkaitz Zubiaga. 2017. SemEval-2017 task 8: RumourEval: Determining rumour veracity and support for rumours. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 69–76, Vancouver, Canada. Association for Computational Linguistics.

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, Minneapolis, Minnesota. Association for Computational Linguistics.

Emily Dinan, Gavin Abercrombie, A Stevie Bergman, Shannon Spruit, Dirk Hovy, Y-Lan Boureau, and Verena Rieser. 2021. Anticipating safety issues in e2e conversational ai: Framework and tooling. *arXiv preprint arXiv:2107.03451*.

Yingtong Dou, Kai Shu, Congying Xia, Philip S Yu, and Lichao Sun. 2021. User preference-aware fake news detection. In *Proceedings of the 44th International ACM*

SIGIR Conference on Research and Development in Information Retrieval, pages 2051–2055.

Rotem Dror, Segev Shlomov, and Roi Reichart. 2019. Deep dominance - how to properly compare deep neural models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2773–2785, Florence, Italy. 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, and 516 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Esin Durmus, Faisal Ladhak, and Claire Cardie. 2019a. Determining relative argument specificity and stance for complex argumentative structures. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4630–4641, Florence, Italy. Association for Computational Linguistics.

Esin Durmus, Faisal Ladhak, and Claire Cardie. 2019b. The role of pragmatic and discourse context in determining argument impact. 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 5668–5678.

Mika Enomoto, Yasuharu Den, and Yuichi Ishimoto. 2020. A conversation-analytic annotation of turn-taking behavior in Japanese multi-party conversation and its preliminary analysis. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 644–652, Marseille, France. European Language Resources Association.

Lizhou Fan, Lingyao Li, Zihui Ma, Sanggyu Lee, Huizi Yu, and Libby Hemphill. 2024. A bibliometric review of large language models research from 2017 to 2023. *ACM Trans. Intell. Syst. Technol.*, 15(5).

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, Singapore. Association for Computational Linguistics.

Margherita Fanton, Helena Bonaldi, Serra Sinem Tekiroğlu, and Marco Guerini. 2021. Human-in-the-loop for data collection: a multi-target counter narrative dataset to fight online hate speech. In *Proceedings of the 59th Annual Meeting of the Association for*

Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 3226–3240, Online. Association for Computational Linguistics.

Diane Felmlee, Cassie McMillan, and Roger Whitaker. 2021. Dyads, triads, and tetrads: a multivariate simulation approach to uncovering network motifs in social graphs. *Applied network science*, 6(1):63.

Xiachong Feng, Xiaocheng Feng, Libo Qin, Bing Qin, and Ting Liu. 2021. Language model as an annotator: Exploring DialoGPT for dialogue summarization. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1479–1491, Online. Association for Computational Linguistics.

Emilio Ferrara, Onur Varol, Clayton Davis, Filippo Menczer, and Alessandro Flammini. 2016. The rise of social bots. *Commun. ACM*, 59(7):96–104.

Mauajama Firdaus, Hardik Chauhan, Asif Ekbal, and Pushpak Bhattacharyya. 2020. MEISD: A multimodal multi-label emotion, intensity and sentiment dialogue dataset for emotion recognition and sentiment analysis in conversations. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4441–4453, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Antigoni Founta, Constantinos Djouvas, Despoina Chatzakou, Ilias Leontiadis, Jeremy Blackburn, Gianluca Stringhini, Athena Vakali, Michael Sirivianos, and Nicolas Kourtellis. 2018. Large scale crowdsourcing and characterization of twitter abusive behavior. *Proceedings of the International AAAI Conference on Web and Social Media*, 12(1).

Shen Gao, Xin Cheng, Mingzhe Li, Xiuying Chen, Jinpeng Li, Dongyan Zhao, and Rui Yan. 2023. Dialogue summarization with static-dynamic structure fusion graph. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13858–13873, Toronto, Canada. Association for Computational Linguistics.

Kiran Garimella, Gianmarco De Francisci Morales, Aristides Gionis, and Michael Mathioudakis. 2018. Political discourse on social media: Echo chambers, gatekeepers, and the price of bipartisanship. In *Proceedings of the 2018 World Wide Web Conference*, WWW '18, page 913–922, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

David R. Gibson. 2003. Participation shifts: Order and differentiation in group conversation*. *Social Forces*, 81(4):1335–1380.

Genevieve Gorrell, Elena Kochkina, Maria Liakata, Ahmet Aker, Arkaitz Zubiaga, Kalina Bontcheva, and Leon Derczynski. 2019. SemEval-2019 task 7: RumourEval, determining rumour veracity and support for rumours. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 845–854, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

H. P. Grice. 1975. Logic and conversation. In Peter Cole and Jerry L. Morgan, editors, *Syntax and Semantics: Vol. 3: Speech Acts*, pages 41–58. Academic Press.

Jia-Chen Gu, Zhenhua Ling, Quan Liu, Cong Liu, and Guoping Hu. 2023. GIFT: Graph-induced fine-tuning for multi-party conversation understanding. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11645–11658, Toronto, Canada. Association for Computational Linguistics.

Jia-Chen Gu, Chao-Hong Tan, Chongyang Tao, Zhen-Hua Ling, Huang Hu, Xiubo Geng, and Daxin Jiang. 2022a. HeterMPC: A heterogeneous graph neural network for response generation in multi-party conversations. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5086–5097, Dublin, Ireland. Association for Computational Linguistics.

Jia-Chen Gu, Chongyang Tao, and Zhen-Hua Ling. 2022b. Who says what to whom: A survey of multi-party conversations. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 5486–5493. International Joint Conferences on Artificial Intelligence Organization. Survey Track.

Jia-Chen Gu, Chongyang Tao, Zhenhua Ling, Can Xu, Xiubo Geng, and Daxin Jiang. 2021. MPC-BERT: A pre-trained language model for multi-party conversation understanding. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3682–3692, Online. Association for Computational Linguistics.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, and 1 others. 2024. A survey on llm-as-a-judge. *arXiv preprint arXiv:2411.15594*.

Sharath Chandra Guntuku, David B Yaden, Margaret L Kern, Lyle H Ungar, and Johannes C Eichstaedt. 2017. Detecting depression and mental illness on social media: an integrative review. *Current Opinion in Behavioral Sciences*, 18:43–49.

Aric Hagberg, Pieter J. Swart, and Daniel A. Schult. 2008. Exploring network structure, dynamics, and function using networkx. Los Alamos National Laboratory (LANL), Los Alamos, NM (United States).

Ji-Eun Han, Jun-Seok Koh, Hyeon-Tae Seo, Du-Seong Chang, and Kyung-Ah Sohn. 2024. PSYDIAL: Personality-based synthetic dialogue generation using large language models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 13321–13331, Torino, Italia. ELRA and ICCL.

Xuanli He, Islam Nassar, Jamie Kiros, Gholamreza Haffari, and Mohammad Norouzi. 2022. Generate, annotate, and learn: Nlp with synthetic text. *Transactions of the Association for Computational Linguistics*, 10:826–842.

Dirk Hovy and Diyi Yang. 2021. The importance of modeling social factors of language: Theory and practice. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 588–602, Online. Association for Computational Linguistics.

Wenpeng Hu, Zhangming Chan, Bing Liu, Dongyan Zhao, Jinwen Ma, and Rui Yan. 2019. Gsn: A graph-structured network for multi-party dialogues. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 5010–5016. International Joint Conferences on Artificial Intelligence Organization.

Xueyu Hu, Tao Xiong, Biao Yi, Zishu Wei, Ruixuan Xiao, Yurun Chen, Jiasheng Ye, Meiling Tao, Xiangxin Zhou, Ziyu Zhao, Yuhuai Li, Shengze Xu, Shenzhi Wang, Xinchen Xu, Shuofei Qiao, Zhaokai Wang, Kun Kuang, Tieyong Zeng, Liang Wang, and 10 others. 2025. OS agents: A survey on MLLM-based agents for computer, phone and browser use. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7436–7465, Vienna, Austria. Association for Computational Linguistics.

Yilun Hua, Nicholas Chernogor, Yuzhe Gu, Seoyeon Jeong, Miranda Luo, and Cristian Danescu-Niculescu-Mizil. 2024. How did we get here? summarizing conversation dynamics. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1:*

Long Papers), pages 7452–7477, Mexico City, Mexico. Association for Computational Linguistics.

Zhaoheng Huang, Zhicheng Dou, Yutao Zhu, and Zhengyi Ma. 2022. MCP: Self-supervised pre-training for personalized chatbots with multi-level contrastive sampling. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1030–1042, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Chia-Chien Hung, Anne Lauscher, Dirk Hovy, Simone Paolo Ponzetto, and Goran Glavaš. 2023. Can demographic factors improve text classification? revisiting demographic adaptation in the age of transformers. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1565–1580, Dubrovnik, Croatia. Association for Computational Linguistics.

Léo Jacqmin, Lina M. Rojas Barahona, and Benoit Favre. 2022. "do you follow me?": A survey of recent approaches in dialogue state tracking. In *Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 336–350, Edinburgh, UK. Association for Computational Linguistics.

A. Janin, D. Baron, J. Edwards, D. Ellis, D. Gelbart, N. Morgan, B. Peskin, T. Pfau, E. Shriberg, A. Stolcke, and C. Wooters. 2003. The icsi meeting corpus. In *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03).*, volume 1, pages I–I.

Qi Jia, Yizhu Liu, Siyu Ren, Kenny Zhu, and Haifeng Tang. 2020. Multi-turn response selection using dialogue dependency relations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1911–1920, Online. Association for Computational Linguistics.

Natasa Jovanovic and Rieks op den Akker. 2004. Towards automatic addressee identification in multi-party dialogues. In *Proceedings of the 5th SIGdial Workshop on Discourse and Dialogue at HLT-NAACL 2004*, pages 89–92, Cambridge, Massachusetts, USA. Association for Computational Linguistics.

Daniel Jurafsky and James H. Martin. 2025. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models*, 3rd edition. Online manuscript released August 24, 2025.

Seungone Kim, Juyoung Suk, Xiang Yue, Vijay Viswanathan, Seongyun Lee, Yizhong Wang, Kiril Gashteovski, Carolin Lawrence, Sean Welleck, and Graham Neubig. 2025. Evaluating language models as synthetic data generators. In *Proceedings of the 63rd*

Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 6385–6403, Vienna, Austria. Association for Computational Linguistics.

Siwon Kim, Sangdoo Yun, Hwaran Lee, Martin Gubri, Sungroh Yoon, and Seong Joon Oh. 2023. Propile: probing privacy leakage in large language models. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Frederic Kirstein, Terry Ruas, Robert Kratel, and Bela Gipp. 2024. Tell me what I need to know: Exploring LLM-based (personalized) abstractive multi-source meeting summarization. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 920–939, Miami, Florida, US. Association for Computational Linguistics.

Filip Klubicka and Raquel Fernández. 2018. Examining a hate speech corpus for hate speech detection and popularity prediction. In *4REAL 2018 Workshop on Replicability and Reproducibility of Research Results in Science and Technology of Language*, page 9.

Klaus Krippendorff. 2011. Computing krippendorff's alpha-reliability. *Computing*, 1:25.

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

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Kornel Laskowski. 2010. Modeling norms of turn-taking in multi-party conversation. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 999–1008, Uppsala, Sweden. Association for Computational Linguistics.

Ran Le, Wenpeng Hu, Mingyue Shang, Zhenjun You, Lidong Bing, Dongyan Zhao, and Rui Yan. 2019. Who is speaking to whom? learning to identify utterance addressee in multi-party conversations. 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 1909–1919, Hong Kong, China. Association for Computational Linguistics.

Chia-Hsuan Lee, Hao Cheng, and Mari Ostendorf. 2021. Dialogue state tracking with a language model using schema-driven prompting. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4937–4949, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Sangmin Lee, Minzhi Li, Bolin Lai, Wenqi Jia, Fiona Ryan, Xu Cao, Ozgur Kara, Bikram Boote, Weiyan Shi, Diyi Yang, and 1 others. 2024. Towards social ai: A survey on understanding social interactions. *arXiv preprint arXiv:2409.15316*.

Baolin Li, Yankai Jiang, Vijay Gadepally, and Devesh Tiwari. 2024a. Sprout: Green generative AI with carbon-efficient LLM inference. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 21799–21813, Miami, Florida, USA. Association for Computational Linguistics.

Ming Li, Jiuhai Chen, Lichang Chen, and Tianyi Zhou. 2024b. Can LLMs speak for diverse people? tuning LLMs via debate to generate controllable controversial statements. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 16160–16176, Bangkok, Thailand. Association for Computational Linguistics.

Qinbin Li, Junyuan Hong, Chulin Xie, Jeffrey Tan, Rachel Xin, Junyi Hou, Xavier Yin, Zhun Wang, Dan Hendrycks, Zhangyang Wang, and 1 others. 2024c. Llm-pbe: Assessing data privacy in large language models. *arXiv preprint arXiv:2408.12787*.

Zekang Li, Jinchao Zhang, Zhengcong Fei, Yang Feng, and Jie Zhou. 2021. Conversations are not flat: Modeling the dynamic information flow across dialogue utterances. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 128–138, Online. Association for Computational Linguistics.

Jingcong Liang, Rong Ye, Meng Han, Ruofei Lai, Xinyu Zhang, Xuanjing Huang, and Zhongyu Wei. 2024. Debatrix: Multi-dimensional debate judge with iterative chronological analysis based on LLM. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 14575–14595, Bangkok, Thailand. Association for Computational Linguistics.

Yuxiao Lin, Yuxian Meng, Xiaofei Sun, Qinghong Han, Kun Kuang, Jiwei Li, and Fei Wu. 2021. BertGCN: Transductive text classification by combining GNN and BERT. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1456–1462, Online. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Adian Liusie, Potsawee Manakul, and Mark Gales. 2024. LLM comparative assessment: Zero-shot NLG evaluation through pairwise comparisons using large language models. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 139–151, St. Julian's, Malta. Association for Computational Linguistics.

Lin Long, Rui Wang, Ruixuan Xiao, Junbo Zhao, Xiao Ding, Gang Chen, and Haobo Wang. 2024. On LLMs-driven synthetic data generation, curation, and evaluation: A survey. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 11065–11082, Bangkok, Thailand. Association for Computational Linguistics.

Renze Lou, Kai Zhang, and Wenpeng Yin. 2024. Large language model instruction following: A survey of progresses and challenges. *Computational Linguistics*, 50(3):1053–1095.

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.

Bo-Ru Lu, Yushi Hu, Hao Cheng, Noah A. Smith, and Mari Ostendorf. 2022. Unsupervised learning of hierarchical conversation structure. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 5657–5670, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Zhenyi Lu, Wei Wei, Xiaoye Qu, Xian-Ling Mao, Dangyang Chen, and Jixiong Chen. 2023. Miracle: Towards personalized dialogue generation with latent-space multiple personal attribute control. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5933–5957, Singapore. Association for Computational Linguistics.

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

Khyati Mahajan, Sashank Santhanam, and Samira Shaikh. 2022. Towards evaluation of multi-party dialogue systems. In *Proceedings of the 15th International Conference*

on Natural Language Generation, pages 278–287, Waterville, Maine, USA and virtual meeting. Association for Computational Linguistics.

Khyati Mahajan and Samira Shaikh. 2021. On the need for thoughtful data collection for multi-party dialogue: A survey of available corpora and collection methods. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 338–352, Singapore and Online. Association for Computational Linguistics.

Khyati Mahajan and Samira Shaikh. 2024. Persona-aware multi-party conversation response generation. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 12712–12723, Torino, Italia. ELRA and ICCL.

Robert Malouf. 1995. Towards an analysis of multi-party discourse. *online), http://hpsg. stanford. edu/rob/talk/node2. html*.

John Mendonça, Alon Lavie, and Isabel Trancoso. 2024. On the benchmarking of LLMs for open-domain dialogue evaluation. In *Proceedings of the 6th Workshop on NLP for Conversational AI (NLP4ConvAI 2024)*, pages 1–12, Bangkok, Thailand. Association for Computational Linguistics.

Stefano Menini, Alessio Palmero Aprosio, and Sara Tonelli. 2021. Abuse is contextual, what about nlp? the role of context in abusive language annotation and detection. *Preprint*, arXiv:2103.14916.

Stefano Menini, Daniel Russo, Alessio Palmero Aprosio, and Marco Guerini. 2025. FirstAID: the first annotation interface for grounded dialogues. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 563–571, Vienna, Austria. Association for Computational Linguistics.

AI Meta. 2025. The llama 4 herd: The beginning of a new era of natively multimodal ai innovation. *https://ai. meta. com/blog/llama-4-multimodal-intelligence/, checked on*, 4(7):2025.

Erxue Min, Yu Rong, Yatao Bian, Tingyang Xu, Peilin Zhao, Junzhou Huang, and Sophia Ananiadou. 2022. Divide-and-conquer: Post-user interaction network for fake news detection on social media. In *Proceedings of the ACM web conference 2022*, pages 1148–1158.

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.

Anders Giovanni Møller, Arianna Pera, Jacob Dalsgaard, and Luca Aiello. 2024. The parrot dilemma: Human-labeled vs. LLM-augmented data in classification tasks. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 179–192, St. Julian's, Malta. Association for Computational Linguistics.

Van-Hoang Nguyen, Kazunari Sugiyama, Preslav Nakov, and Min-Yen Kan. 2020. Fang: Leveraging social context for fake news detection using graph representation. In *Proceedings of the 29th ACM international conference on information & knowledge management*, pages 1165–1174.

Daniela Occhipinti, Michele Marchi, Irene Mondella, Huiyuan Lai, Felice Dell'Orletta, Malvina Nissim, and Marco Guerini. 2024a. Fine-tuning with HED-IT: The impact of human post-editing for dialogical language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 11892–11907, Bangkok, Thailand. Association for Computational Linguistics.

Daniela Occhipinti, Serra Sinem Tekiroğlu, and Marco Guerini. 2024b. PRODIGy: a PROfile-based DIalogue generation dataset. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 3500–3514, Mexico City, Mexico. Association for Computational Linguistics.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, Nathan Lambert, Dustin Schwenk, Oyvind Tafjord, Taira Anderson, David Atkinson, Faeze Brahman, Christopher Clark, Pradeep Dasigi, Nouha Dziri, and 21 others. 2024. 2 olmo 2 furious. *Preprint*, arXiv:2501.00656.

OpenAI. 2022. Introducing chatgpt.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Hiroki Ouchi and Yuta Tsuboi. 2016. Addressee and response selection for multi-party conversation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural*

Language Processing, pages 2133–2143, Austin, Texas. Association for Computational Linguistics.

Pietro Panzarasa, Tore Opsahl, and Kathleen M Carley. 2009. Patterns and dynamics of users' behavior and interaction: Network analysis of an online community. *Journal of the American Society for Information Science and Technology*, 60(5):911–932.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, UIST '23, New York, NY, USA. Association for Computing Machinery.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are NLP models really able to solve simple math word problems? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2080–2094, Online. Association for Computational Linguistics.

Birgit Pauksztat, Christian Steglich, and Rafael Wittek. 2011. Who speaks up to whom? a relational approach to employee voice. *Social Networks*, 33(4):303–316.

John Pavlopoulos, Prodromos Malakasiotis, Juli Bakagianni, and Ion Androutsopoulos. 2017. Improved abusive comment moderation with user embeddings. In *Proceedings of the 2017 EMNLP Workshop: Natural Language Processing meets Journalism*, pages 51–55, Copenhagen, Denmark. Association for Computational Linguistics.

Nicolò Penzo, Antonio Longa, Bruno Lepri, Sara Tonelli, and Marco Guerini. 2024a. Putting context in context: the impact of discussion structure on text classification. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1793–1811, St. Julian's, Malta. Association for Computational Linguistics.

Nicolò Penzo, Maryam Sajedinia, Bruno Lepri, Sara Tonelli, and Marco Guerini. 2024b. Do LLMs suffer from multi-party hangover? a diagnostic approach to addressee recognition and response selection in conversations. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 11210–11233, Miami, Florida, USA. Association for Computational Linguistics.

Nicolò Penzo, Marco Guerini, Bruno Lepri, Goran Glavaš, and Sara Tonelli. 2026. Don't stop the multi-party! on generating synthetic written multi-party conversations with constraints. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(39):32701–32709.

Ildiko Pilan, Laurent Prévot, Hendrik Buschmeier, and Pierre Lison. 2024. Conversational feedback in scripted versus spontaneous dialogues: A comparative analysis. In *Proceedings of the 25th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 440–457, Kyoto, Japan. Association for Computational Linguistics.

Paloma Piot and Javier Parapar. 2025. Decoding hate: Exploring language models' reactions to hate speech. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 973–990, Albuquerque, New Mexico. Association for Computational Linguistics.

Soujanya Poria, Devamanyu Hazarika, Navonil Majumder, Gautam Naik, Erik Cambria, and Rada Mihalcea. 2019a. MELD: A multimodal multi-party dataset for emotion recognition in conversations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 527–536, Florence, Italy. Association for Computational Linguistics.

Soujanya Poria, Navonil Majumder, Rada Mihalcea, and Eduard Hovy. 2019b. Emotion recognition in conversation: Research challenges, datasets, and recent advances. *IEEE Access*, 7:100943–100953.

Liang Qiu, Yizhou Zhao, Weiyan Shi, Yuan Liang, Feng Shi, Tao Yuan, Zhou Yu, and Song-Chun Zhu. 2020. Structured attention for unsupervised dialogue structure induction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1889–1899, Online. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Zahra Rahimi and Diane Litman. 2020. Entrainment2vec: Embedding entrainment for multi-party dialogues. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8681–8688.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. 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 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Alan Ritter, Colin Cherry, and Bill Dolan. 2010. Unsupervised modeling of Twitter conversations. In *Human Language Technologies: The 2010 Annual Conference of the*

North American Chapter of the Association for Computational Linguistics, pages 172–180, Los Angeles, California. Association for Computational Linguistics.

Anindya Roy, Camille Guinaudeau, Hervé Bredin, and Claude Barras. 2014. TVD: A reproducible and multiply aligned TV series dataset. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 418–425, Reykjavik, Iceland. European Language Resources Association (ELRA).

Daniel Russo, Shane Kaszefski-Yaschuk, Jacopo Staiano, and Marco Guerini. 2023. Countering misinformation via emotional response generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 11476–11492, Singapore. Association for Computational Linguistics.

Harvey Sacks, Emanuel A Schegloff, and Gail Jefferson. 1974. A simplest systematics for the organization of turn taking for conversation. *Language*, 50(4):696–735.

Milene Santos Teixeira and Mauro Dragoni. 2022. A review of plan-based approaches for dialogue management. *Cognitive Computation*, 14(3):1019–1038.

Ashtosh Sapru and Hervé Bourlard. 2014. Detecting speaker roles and topic changes in multiparty conversations using latent topic models. In *Fifteenth Annual Conference of the International Speech Communication Association*.

Martin Saveski, Brandon Roy, and Deb Roy. 2021. The structure of toxic conversations on twitter. In *Proceedings of the Web Conference 2021*, pages 1086–1097.

Jost Schatzmann, Karl Weilhammer, Matt Stuttle, and Steve Young. 2006. A survey of statistical user simulation techniques for reinforcement-learning of dialogue management strategies. *The Knowledge Engineering Review*, 21(2):97–126.

Thomas Scialom, Serra Sinem Tekiroğlu, Jacopo Staiano, and Marco Guerini. 2020. Toward stance-based personas for opinionated dialogues. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2625–2635, Online. Association for Computational Linguistics.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. 2024. Quantifying language models' sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting. In *The Twelfth International Conference on Learning Representations*.

Iulian Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 30(1).

Chengcheng Shao, Giovanni Luca Ciampaglia, Onur Varol, Kai-Cheng Yang, Alessandro Flammini, and Filippo Menczer. 2018. The spread of low-credibility content by social bots. *Nature communications*, 9(1):4787.

Weizhou Shen, Siyue Wu, Yunyi Yang, and Xiaojun Quan. 2021. Directed acyclic graph network for conversational emotion recognition. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1551–1560, Online. Association for Computational Linguistics.

Amit Sheth, Valerie L Shalin, and Ugur Kursuncu. 2022. Defining and detecting toxicity on social media: context and knowledge are key. *Neurocomputing*, 490:312–318.

Chang Shu, Jiuzhou Han, Fangyu Liu, Ehsan Shareghi, and Nigel Collier. 2023. POSQA: Probe the world models of LLMs with size comparisons. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7518–7531, Singapore. Association for Computational Linguistics.

Kai Shu, Deepak Mahudeswaran, Suhang Wang, and Huan Liu. 2020. Hierarchical propagation networks for fake news detection: Investigation and exploitation. *Proceedings of the International AAAI Conference on Web and Social Media*, 14(1):626–637.

Kai Shu, Suhang Wang, and Huan Liu. 2019. Beyond news contents: The role of social context for fake news detection. In *Proceedings of the twelfth ACM international conference on web search and data mining*, pages 312–320.

Chenguang Song, Kai Shu, and Bin Wu. 2021. Temporally evolving graph neural network for fake news detection. *Information Processing & Management*, 58(6):102712.

Sanja Stajner and Seren Yenikent. 2020. A survey of automatic personality detection from texts. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6284–6295, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Hsuan Su, Jiun-Hao Jhan, Fan-yun Sun, Saurav Sahay, and Hung-yi Lee. 2021. Put chatbot into its interlocutor's shoes: New framework to learn chatbot responding with intention. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1559–1569, Online. Association for Computational Linguistics.

Jiuding Sun, Chantal Shaib, and Byron C Wallace. 2024. Evaluating the zero-shot robustness of instruction-tuned language models. In *The Twelfth International Conference on Learning Representations*.

Yang Sun, Nan Yu, and Guohong Fu. 2021. A discourse-aware graph neural network for emotion recognition in multi-party conversation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2949–2958, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Sathya Krishnan Suresh, Wu Mengjun, Tushar Pranav, and EngSiong Chng. 2025. DiaSynth: Synthetic dialogue generation framework for low resource dialogue applications. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 673–690, Albuquerque, New Mexico. Association for Computational Linguistics.

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.

Lin Tian, Xiuzhen Zhang, and Jey Han Lau. 2022. DUCK: Rumour detection on social media by modelling user and comment propagation networks. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4939–4949, Seattle, United States. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

David Traum. 2003. Issues in multiparty dialogues. In *Workshop on Agent Communication Languages*, pages 201–211. Springer.

Dennis Ulmer, Christian Hardmeier, and Jes Frellsen. 2022. deep-significance: Easy and meaningful signifcance testing in the age of neural networks. In *ML Evaluation Standards Workshop at the Tenth International Conference on Learning Representations*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

Veniamin Veselovsky, Manoel Horta Ribeiro, Akhil Arora, Martin Josifoski, Ashton Anderson, and Robert West. 2023. Generating faithful synthetic data with large language models: A case study in computational social science. *arXiv preprint arXiv:2305.15041*.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, and 16 others. 2020. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272.

Soroush Vosoughi, Deb K. Roy, and Sinan Aral. 2018. The spread of true and false news online. *Science*, 359:1146 – 1151.

Weishi Wang, Steven C.H. Hoi, and Shafiq Joty. 2020. Response selection for multi-party conversations with dynamic topic tracking. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6581–6591, Online. Association for Computational Linguistics.

Jimmy Wei, Kurt Shuster, Arthur Szlam, Jason Weston, Jack Urbanek, and Mojtaba Komeili. 2023. Multi-party chat: Conversational agents in group settings with humans and models. *Preprint*, arXiv:2304.13835.

Joseph Weizenbaum. 1966. Eliza—a computer program for the study of natural language communication between man and machine. *Commun. ACM*, 9(1):36–45.

Haoyang Wen, Zhenxin Xiao, Eduard Hovy, and Alexander Hauptmann. 2023. Towards open-domain Twitter user profile inference. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 3172–3188, Toronto, Canada. Association for Computational Linguistics.

Thomas P Wilson, John M Wiemann, and Don H Zimmerman. 1984. Models of turn taking in conversational interaction. *Journal of Language and Social Psychology*, 3(3):159–183.

Ludwig Wittgenstein. 1953. *Philosophical Investigations*. Macmillan, New York, NY, USA.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W White, Doug Burger, and Chi Wang. 2023. Autogen: Enabling next-gen llm applications via multi-agent conversation. *Preprint*, arXiv:2308.08155.

Tianhao Wu, Weizhe Yuan, Olga Golovneva, Jing Xu, Yuandong Tian, Jiantao Jiao, Jason E Weston, and Sainbayar Sukhbaatar. 2025. Meta-rewarding language models: Self-improving alignment with LLM-as-a-meta-judge. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 11548–11565, Suzhou, China. Association for Computational Linguistics.

Feng Xiachong, Feng Xiaocheng, and Qin Bing. 2021. Incorporating commonsense knowledge into abstractive dialogue summarization via heterogeneous graph networks. In *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, pages 964–975, Huhhot, China. Chinese Information Processing Society of China.

Chaojun Xiao, Jie Cai, Weilin Zhao, Biyuan Lin, Guoyang Zeng, Jie Zhou, Zhi Zheng, Xu Han, Zhiyuan Liu, and Maosong Sun. 2025. Densing law of llms. *Nature Machine Intelligence*, pages 1–11.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, and 43 others. 2024. Qwen2 technical report. *Preprint*, arXiv:2407.10671.

Ruichao Yang, Wei Gao, Jing Ma, Hongzhan Lin, and Zhiwei Yang. 2023. WSDMS: Debunk fake news via weakly supervised detection of misinforming sentences with contextualized social wisdom. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1525–1538, Singapore. Association for Computational Linguistics.

Ruoyao Yang, Wanying Xie, Chunhua Liu, and Dong Yu. 2019. BLCU_NLP at SemEval-2019 task 7: An inference chain-based GPT model for rumour evaluation. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 1090–1096, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Graph convolutional networks for text classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):7370–7377.

Steve Young, Milica Gašić, Blaise Thomson, and Jason D. Williams. 2013. Pomdp-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5):1160–1179.

Mingzhi Yu, Emer Gilmartin, and Diane Litman. 2019. Identifying personality traits using overlap dynamics in multiparty dialogue. *arXiv preprint arXiv:1909.00876*.

Xinchen Yu, Eduardo Blanco, and Lingzi Hong. 2022. Hate speech and counter speech detection: Conversational context does matter. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5918–5930, Seattle, United States. Association for Computational Linguistics.

Mert Yuksekgonul, Federico Bianchi, Joseph Boen, Sheng Liu, Pan Lu, Zhi Huang, Carlos Guestrin, and James Zou. 2025. Optimizing generative ai by backpropagating language model feedback. *Nature*, 639(8055):609–616.

Justine Zhang, Jonathan Chang, Cristian Danescu-Niculescu-Mizil, Lucas Dixon, Yiqing Hua, Dario Taraborelli, and Nithum Thain. 2018a. Conversations gone awry: Detecting early signs of conversational failure. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1350–1361, Melbourne, Australia. Association for Computational Linguistics.

Rui Zhang, Honglak Lee, Lazaros Polymenakos, and Dragomir Radev. 2018b. Addressee and response selection in multi-party conversations with speaker interaction rnns. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

Wenjie Zheng, Jianfei Yu, Rui Xia, and Shijin Wang. 2023. A facial expression-aware multimodal multi-task learning framework for emotion recognition in multi-party conversations. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15445–15459, Toronto, Canada. Association for Computational Linguistics.

Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2020. Graph neural networks: A review of methods and applications. *AI open*, 1:57–81.

Kaimin Zhou, Chang Shu, Binyang Li, and Jey Han Lau. 2019. Early rumour detection. 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 1614–1623, Minneapolis, Minnesota. Association for Computational Linguistics.

Ke Zhou, Marios Constantinides, Luca Maria Aiello, Sagar Joglekar, and Daniele Quercia. 2021. The role of different types of conversations for meeting success. *IEEE Pervasive Computing*, 20(4):35–42.

Xuhui Zhou, Zhe Su, Tiwalayo Eisape, Hyunwoo Kim, and Maarten Sap. 2024. Is this the real life? is this just fantasy? the misleading success of simulating social interactions with LLMs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 21692–21714, Miami, Florida, USA. Association for Computational Linguistics.

Pengcheng Zhu, Wei Zhou, Kuncai Zhang, Yuankai Ma, and Haiqing Chen. 2023. Robust learning for multi-party addressee recognition with discrete addressee codebook. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 571–578, Toronto, Canada. Association for Computational Linguistics.

Caleb Ziems, William Held, Omar Shaikh, Jiaao Chen, Zhehao Zhang, and Diyi Yang. 2024. Can large language models transform computational social science? *Computational Linguistics*, 50(1):237–291.

Appendix A

Modeling the conversational context

A.1 Training Details and training pipeline

Here we report all the hyperparameters used in the experiments presented in Chapter 3.

We exploit Optuna (Akiba et al., 2019) for hyperparameter search, using a grid search for: (I.) the learning rate, with a uniform probability between the values $7.5 \cdot 10^{-6}$, $1.0 \cdot 10^{-5}$, $2.5 \cdot 10^{-5}$, $5 \cdot 10^{-5}$, $7.5 \cdot 10^{-5}$; (II.) the dropout applied between the layers of the MLP, with values 0.25 and 0.5. We use batch size $b = 32$ and weight decay $w_d = 10^{-4}$ in the `RoBERTa` components. In SDK dataset, we use unweighted Cross Entropy loss both in the training and in the validation phase, since the imbalance is negligible.

For the final evaluation, we fix the hyperparameters and then we perform 10 runs, changing each time the random seed. Then we keep the 5 best runs in validation, in order to exclude possible "outlier" runs due to initialization problems. We compute the average and standard deviation of the test results on these 5 best runs.

We perform backpropagation on the full structure of the model, without freezing any layer. As previously stated, our experiments keep always the same model, just changing the input. We use early stopping for model selection with patience $p = 2$ epochs for the SDK dataset (Section 3.5 and Section 3.5.2) and $p = 5$ epochs for the SQDC dataset (Appendix 3.6.1). In the SDK dataset, each epoch corresponds to a training epoch on a sample of the training set, which is around half of the total training set, in order to speed up computation and generalization. We test also the usage of the full training set in each epoch, but the results remain comparable.

For all the experiments we use a single A40 GPU with 48GB Memory. All the experimental code is developed in PyTorch. It requires around 33 minutes of computation for each epoch (training phase plus validation phase).

Category	**Model**	**LR**	**DO**
Dummy	Majority	/	/
	Random	/	/
Baselines	Single	$7.5 \cdot 10^{-6}$	0.5
	Pair	$7.5 \cdot 10^{-6}$	0.25
Contextual	TC	$7.5 \cdot 10^{-6}$	0.25
	TC + T	$7.5 \cdot 10^{-6}$	0.25
	TC + U	$1.0 \cdot 10^{-5}$	0.25
	TC + U + T	$7.5 \cdot 10^{-6}$	0.25

Table A.1: Training hyperparameters for SDK dataset. LR column reports the Learning Rate and DO column reports the dropout value in the MLP component.

Category	**Model**	**LR**	**DO**
Dummy	Majority	/	/
	Random	/	/
Baseline	Single	$5.0 \cdot 10^{-5}$	0.25
	Pair	$2.5 \cdot 10^{-5}$	0.25
Cont.	TC	$1.0 \cdot 10^{-5}$	0.25
	TC + T	$1.0 \cdot 10^{-5}$	0.5
	TC + U	$2.5 \cdot 10^{-5}$	0.25
	TC + U + T	$2.5 \cdot 10^{-5}$	0.5

Table A.2: *SQDC - Challenge.* Training hyperparameters for SQDC dataset, on the original split given for the challenge. LR column reports the Learning Rate and DO column reports the dropout value in the MLP component.

Category	**Model**	**LR**	**DO**
Dummy	Majority	/	/
	Random	/	/
Baseline	Single	$5.0 \cdot 10^{-5}$	0.25
	Pair	$2.5 \cdot 10^{-5}$	0.25
Contextual	TC	$2.5 \cdot 10^{-5}$	0.5
	TC + T	$7.5 \cdot 10^{-6}$	0.25
	TC + U	$1.0 \cdot 10^{-5}$	0.25
	TC + U + T	$1.0 \cdot 10^{-5}$	0.5

Table A.3: *SQDC - New split.* Training hyperparameters for SQDC dataset, with our new split to obtain complex structures even in training. LR column reports the Learning Rate and DO column reports the dropout value in the MLP component.

Category	**Model**	**LR**	**DO**
Dummy	Majority	/	/
	Random	/	/
Baseline	Single	$1.0 \cdot 10^{-5}$	0.5
	Pair	$7.5 \cdot 10^{-6}$	0.5
Contextual	TC	$7.5 \cdot 10^{-6}$	0.25
	TC + T	$1.0 \cdot 10^{-5}$	0.5
	TC + U	$1.0 \cdot 10^{-5}$	0.5
	TC + U + T	$2.5 \cdot 10^{-5}$	0.25

Table A.4: *SQDC - Binary.* Training hyperparameters for SQDC dataset, with our new split to obtain complex structures even in training, for the binary task to detect Stance class vs No Stance Class. LR column reports the Learning Rate and DO column reports the dropout value in the MLP component.

Category	**Model**	**LR**	**DO**
Dummy	Majority	/	/
	Random	/	/
Baselines	Single	$1.0 \cdot 10^{-5}$	0.5
Contextual	TC	$7.5 \cdot 10^{-6}$	0.5
	TC + T	$1.0 \cdot 10^{-5}$	0.5
	TC + U	$7.5 \cdot 10^{-6}$	0.5
	TC + U + T	$7.5 \cdot 10^{-6}$	0.25

Table A.5: *ContextAbuse.* Training hyperparameters for ContextAbuse dataset. LR column reports the Learning Rate and DO column reports the dropout value in the MLP component.

Appendix B

Modeling the conversational components

B.1 Technical details

For our experiments we use a single A40 GPU with 48GB Memory. With such GPU, it is possible to use `Llama2-13b-chat` only in inference with a batch size of 1. We used the `Llama-2-13b-chat-hf` version provided by HuggingFace[1]. We use Copilot[2] as a coding assistant.

B.2 Prompt designs

In this section, we report all three prompt variants (i.e., *verbose*, *medium*, and *concise*) for each prompt component used in the classification and generation experiments presented in Chapter 4 and described in Section 4.3.3.

Figure B.1 presents the *Scenario Description* and *User Description*, which remain constant across all input–task combinations. Figure B.2 and Figure B.3 report the *Input Elements* and *Input Format* sections, which vary depending on the input configuration. Figures B.4, B.5, and B.6 illustrate the *Task Definition*, *Instruction Template*, and *Output Template*, respectively; these sections are task-dependent.

Figure B.7 shows the prompt used for generation tasks, which always includes both the *Conversation Transcript* and the *Interaction Transcript* as input and varies according to the specific generation task. Figure B.8 presents the classification prompt, which depends on both the (classification) task and the selected input information. In Figure B.9, we report the two output templates evaluated in the analysis discussed in Section 4.6.3.

[1] `https://huggingface.co/meta-llama/Llama-2-13b-chat-hf`

[2] `https://github.com/features/copilot`

Finally, Figure B.10 provides a complete example of a prompt used for the addressee recognition task, combining *Interaction Transcript* and *Summary* as input information.

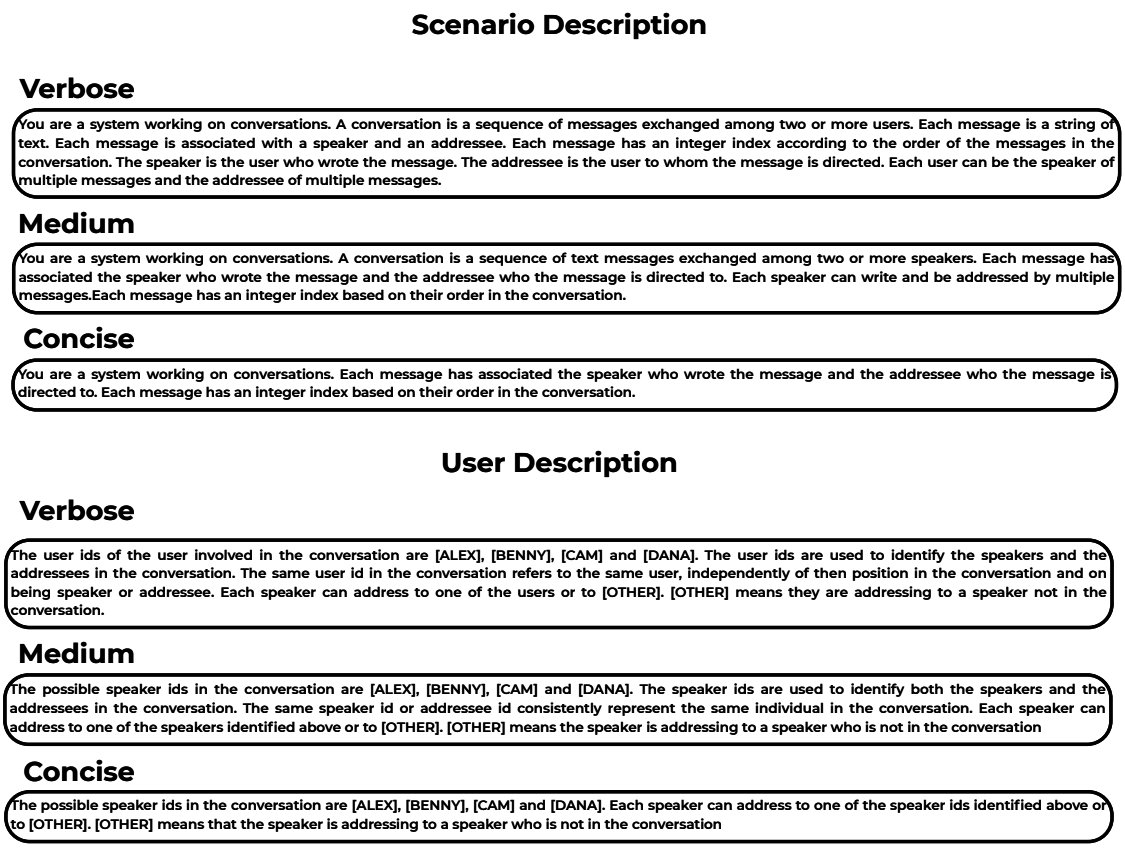

Figure B.1: Scenario Description and User Description sections of the prompts. These components are shared across all tasks and input configurations and define the conversational setting and the participants involved.

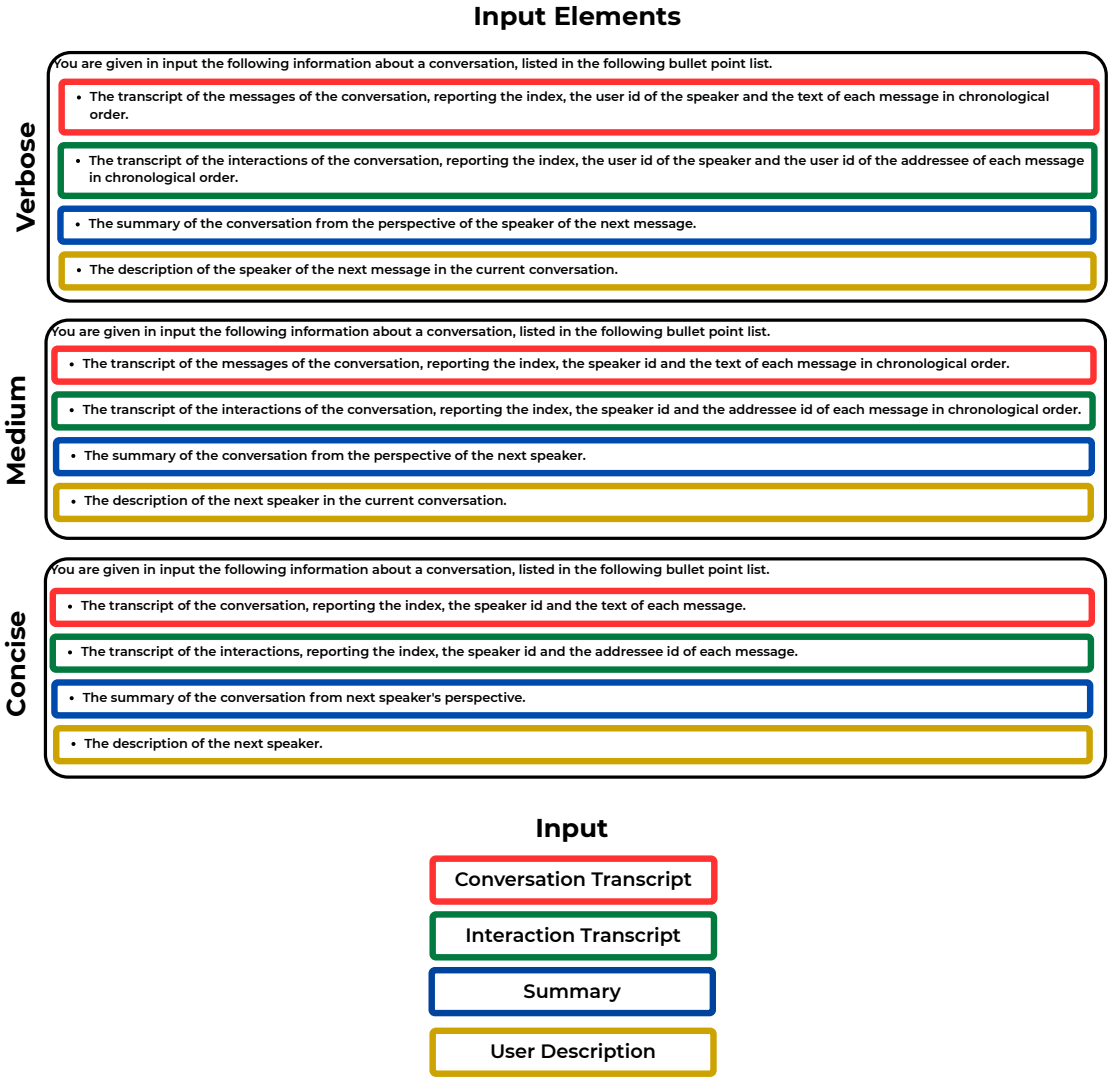

Figure B.2: Input Elements sections of the prompts. This sections specify which input information is provided to the model, therefore varying across different input configurations.

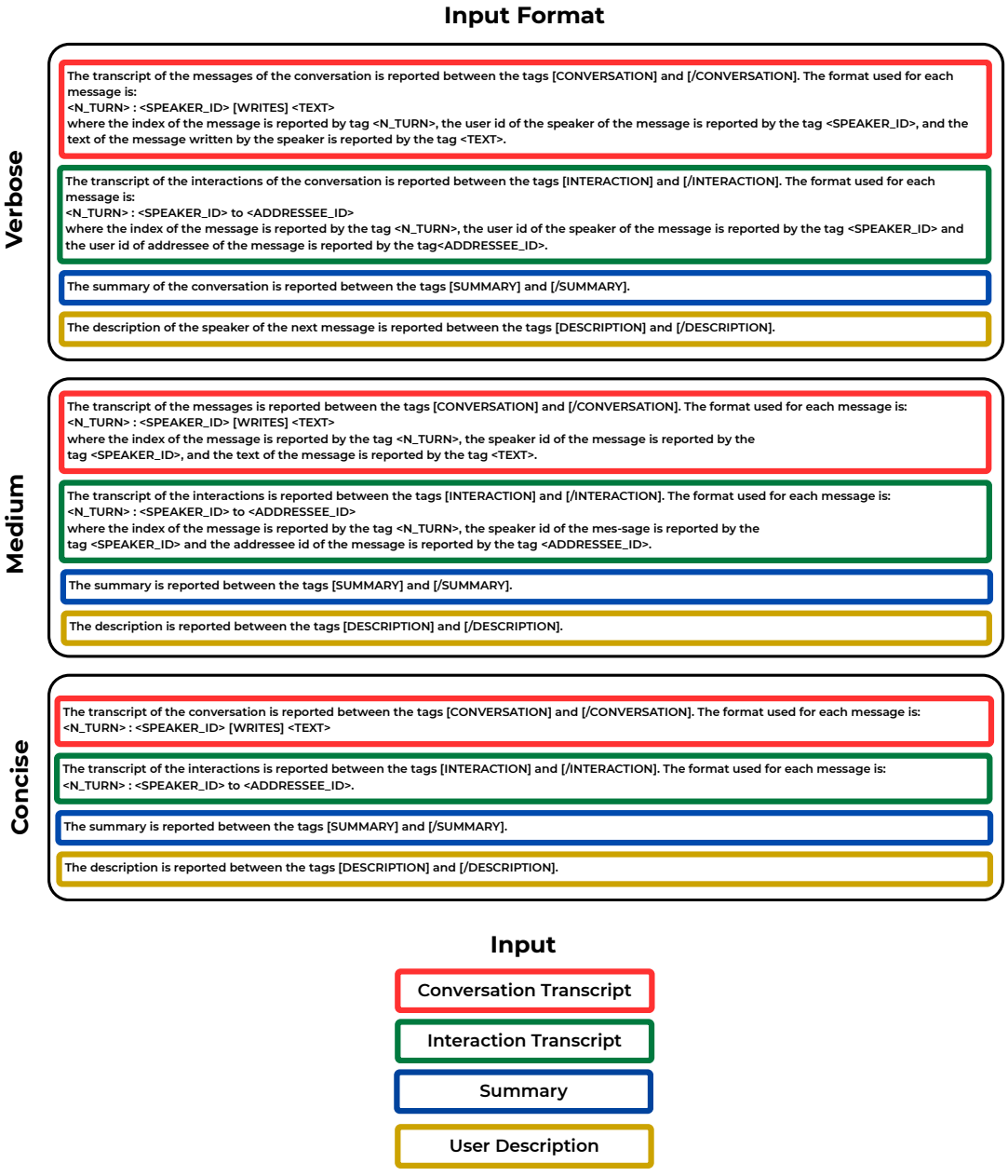

Figure B.3: Input Format sections of the prompts. This section specify how input information is structured and given to the model, therefore varying across different input configurations.

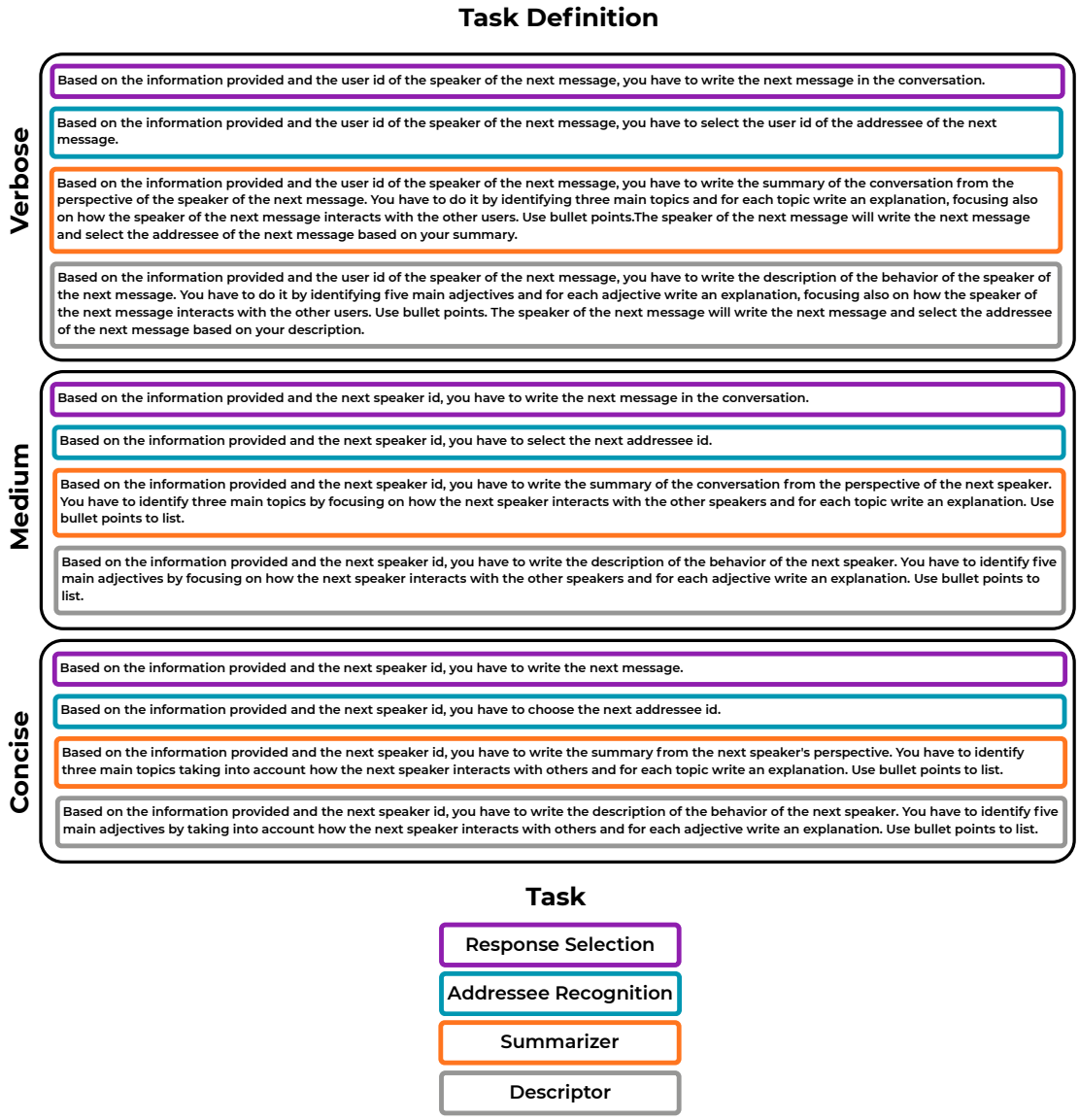

Figure B.4: Task Definition section of the prompt. This component describes the objective of the task to be performed (e.g., addressee recognition or response selection) and is task-dependent.

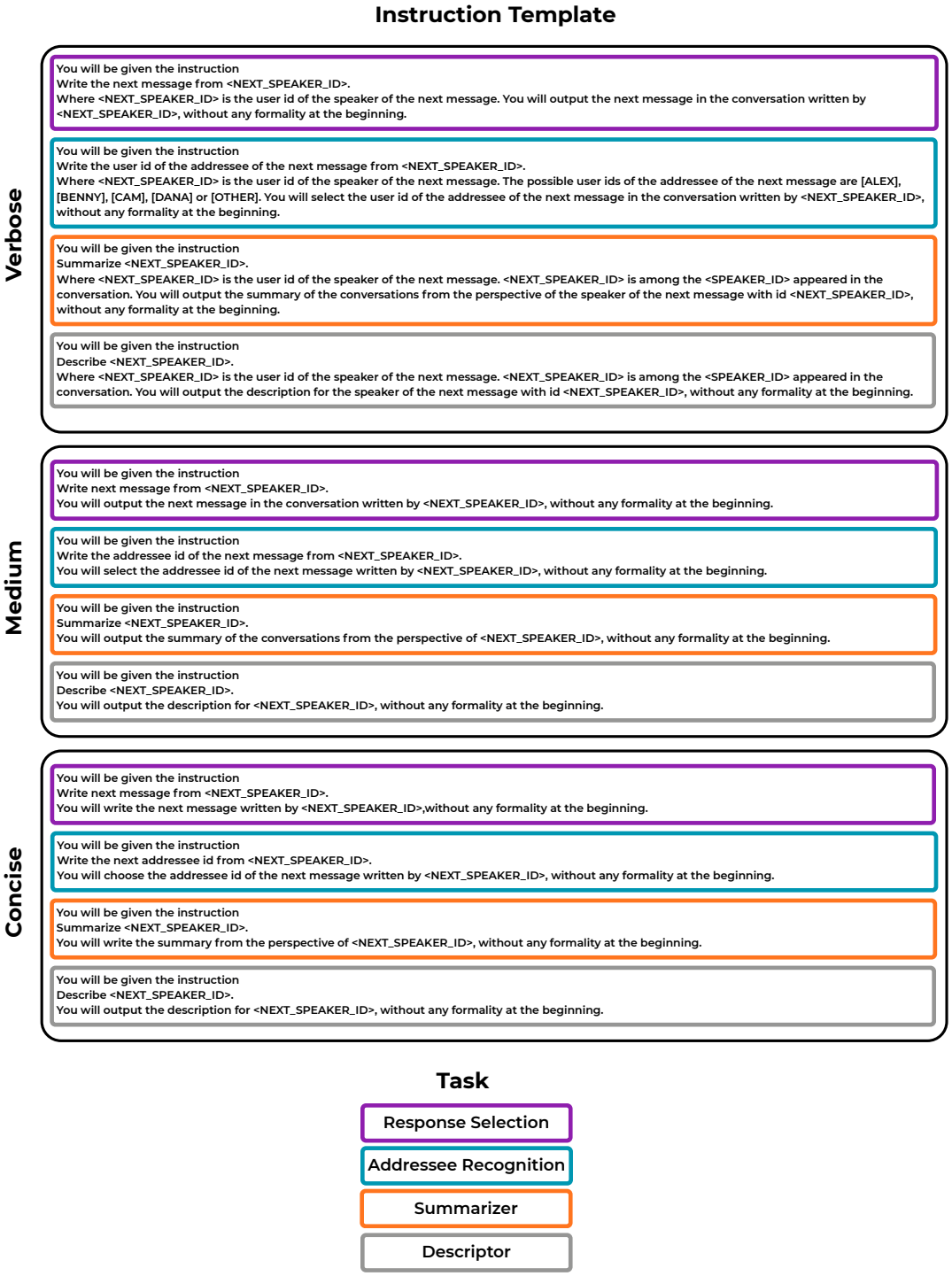

Figure B.5: Instruction Template section of the prompt. This section provides explicit instructions guiding the model on how to perform the task.

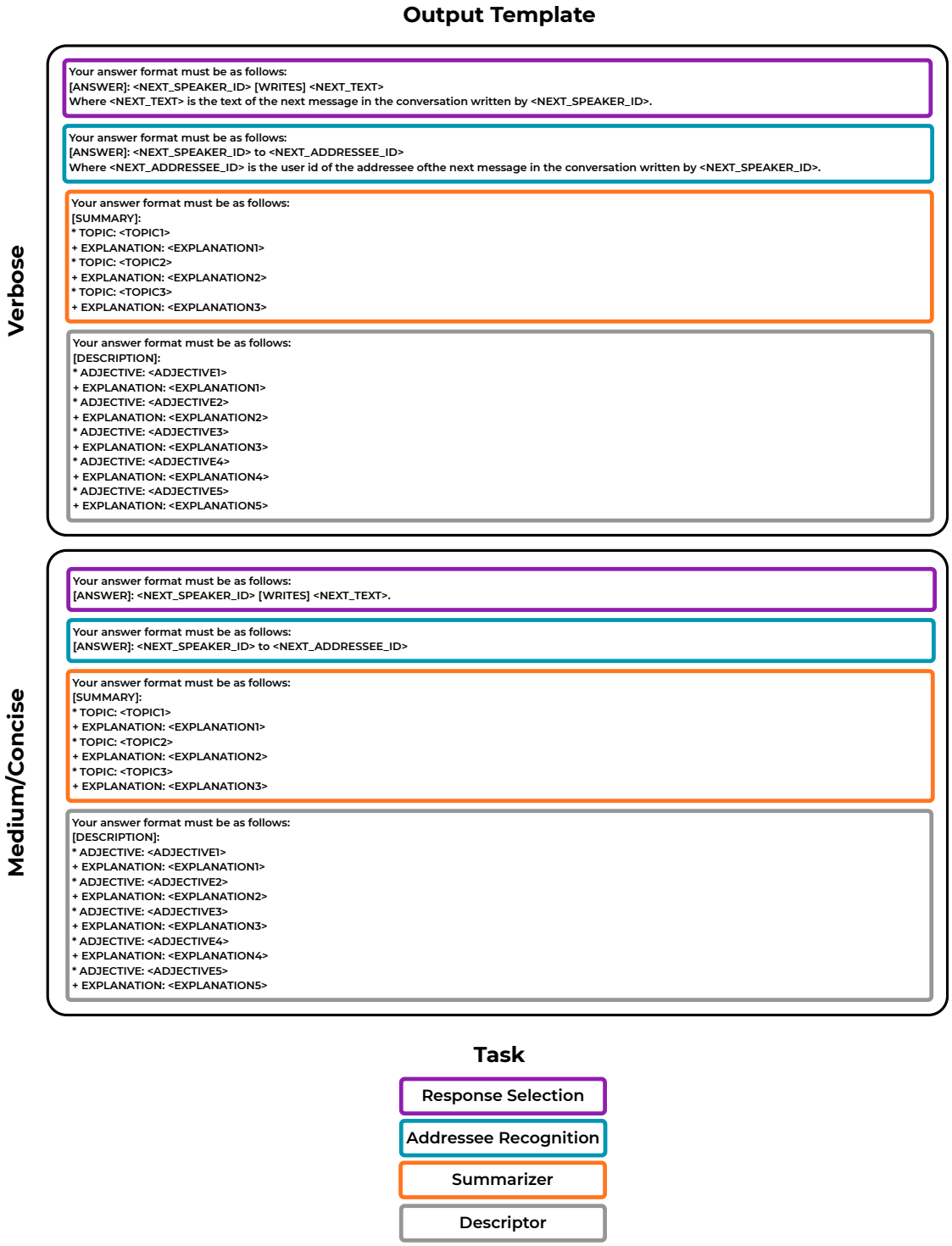

Figure B.6: Output Template section of the prompt. This component constrains the format of the model's response and defines the expected structure of the output for each task.

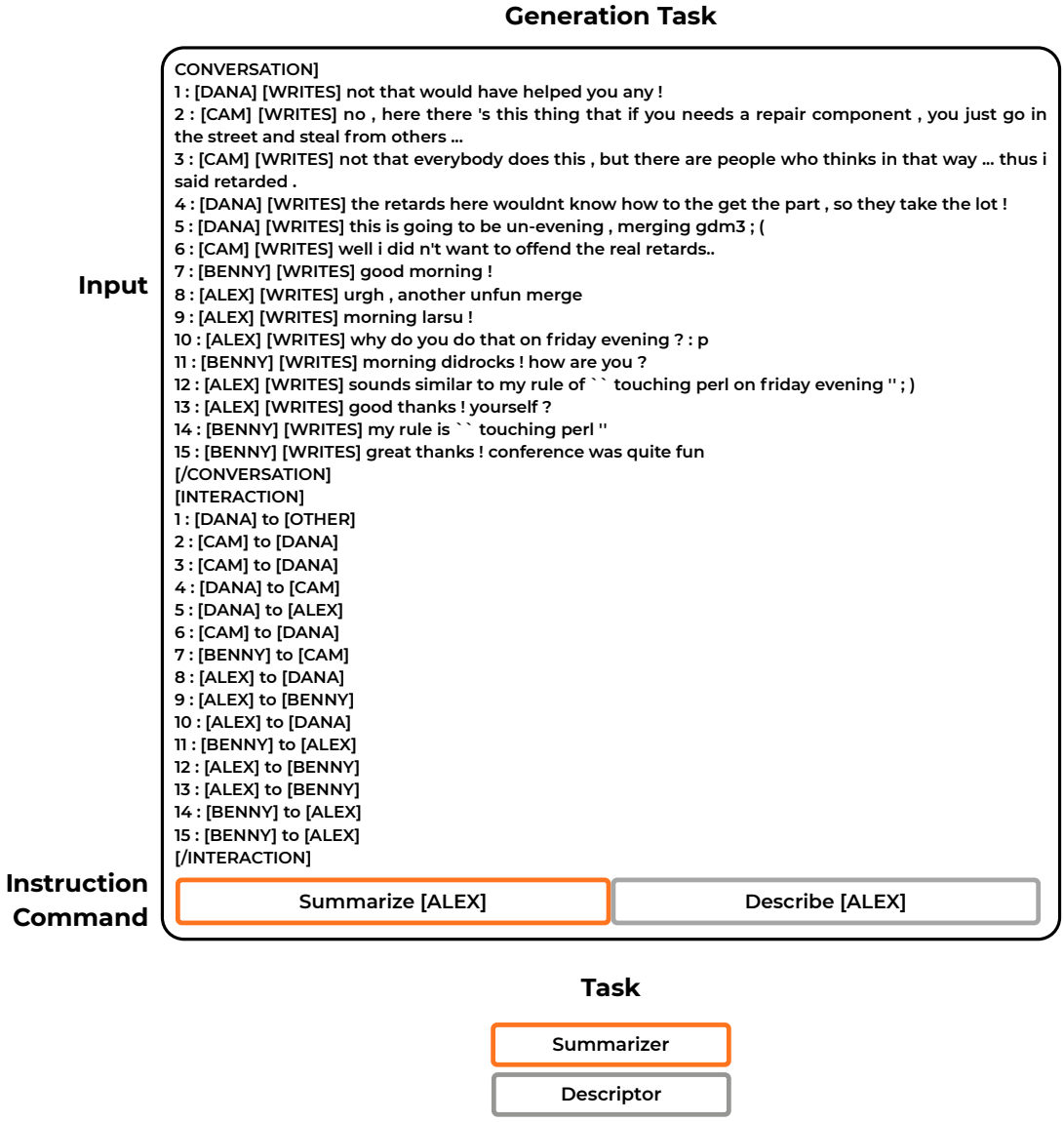

Figure B.7: Prompt used for generation tasks. The prompt always includes the Conversation Transcript and the Interaction Transcript as input and is adapted to the specific generation task.

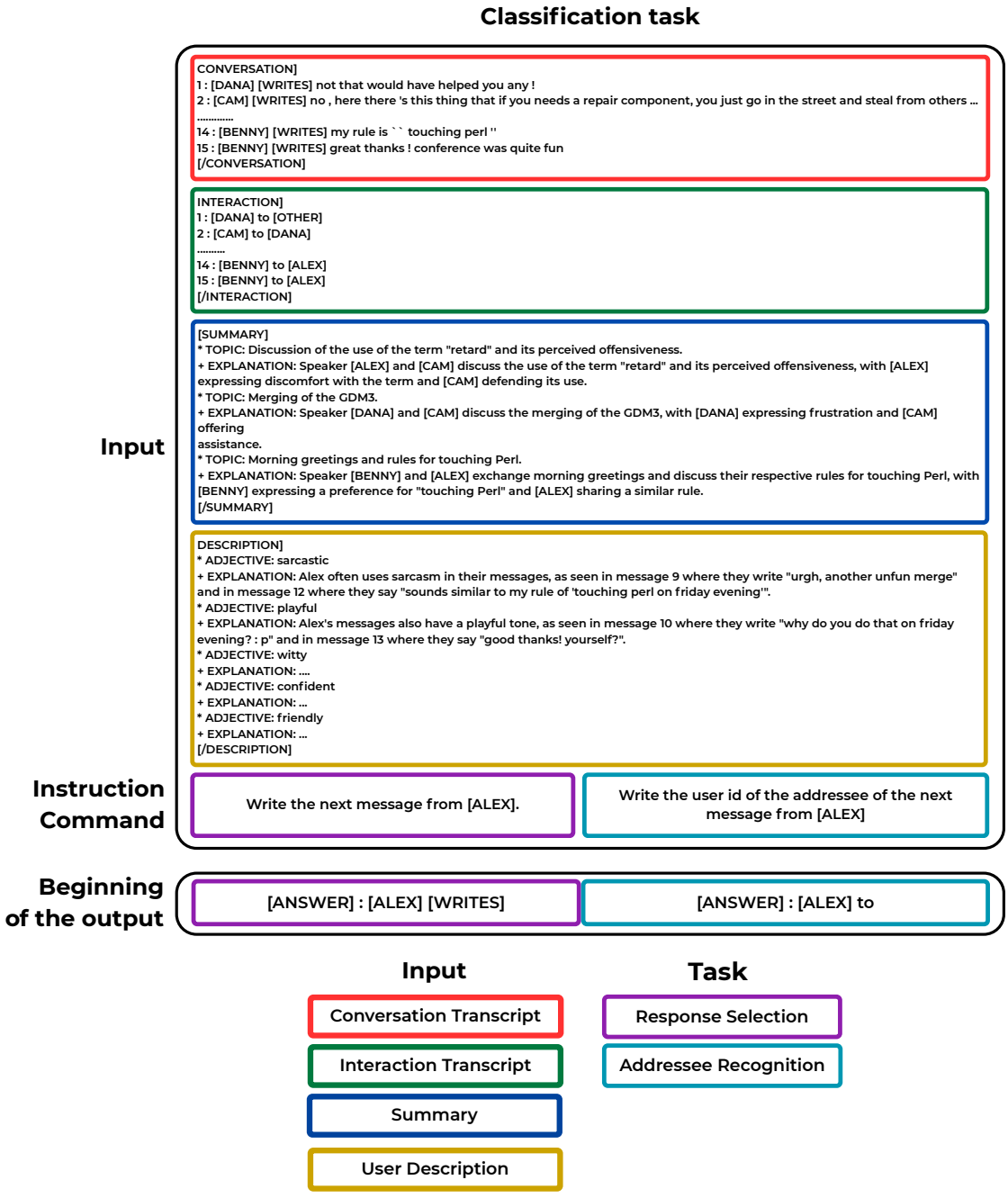

Figure B.8: Prompt used for classification tasks. The prompt varies depending on both the task (i.e., addressee recognition or response selection) and the selected input information.

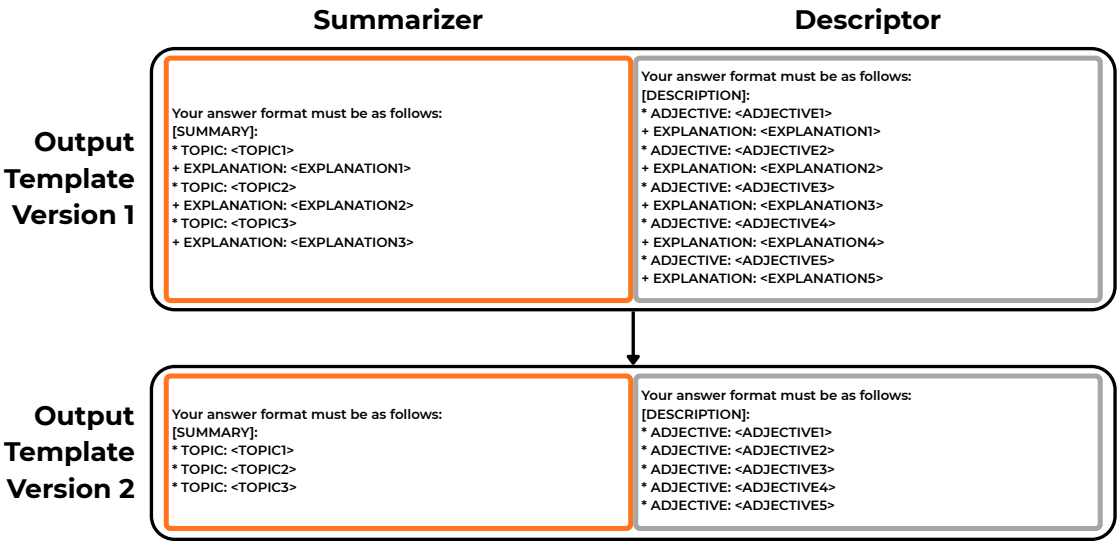

Figure B.9: Comparison of the two output templates evaluated for generation tasks, with and without explicit explanation fields.

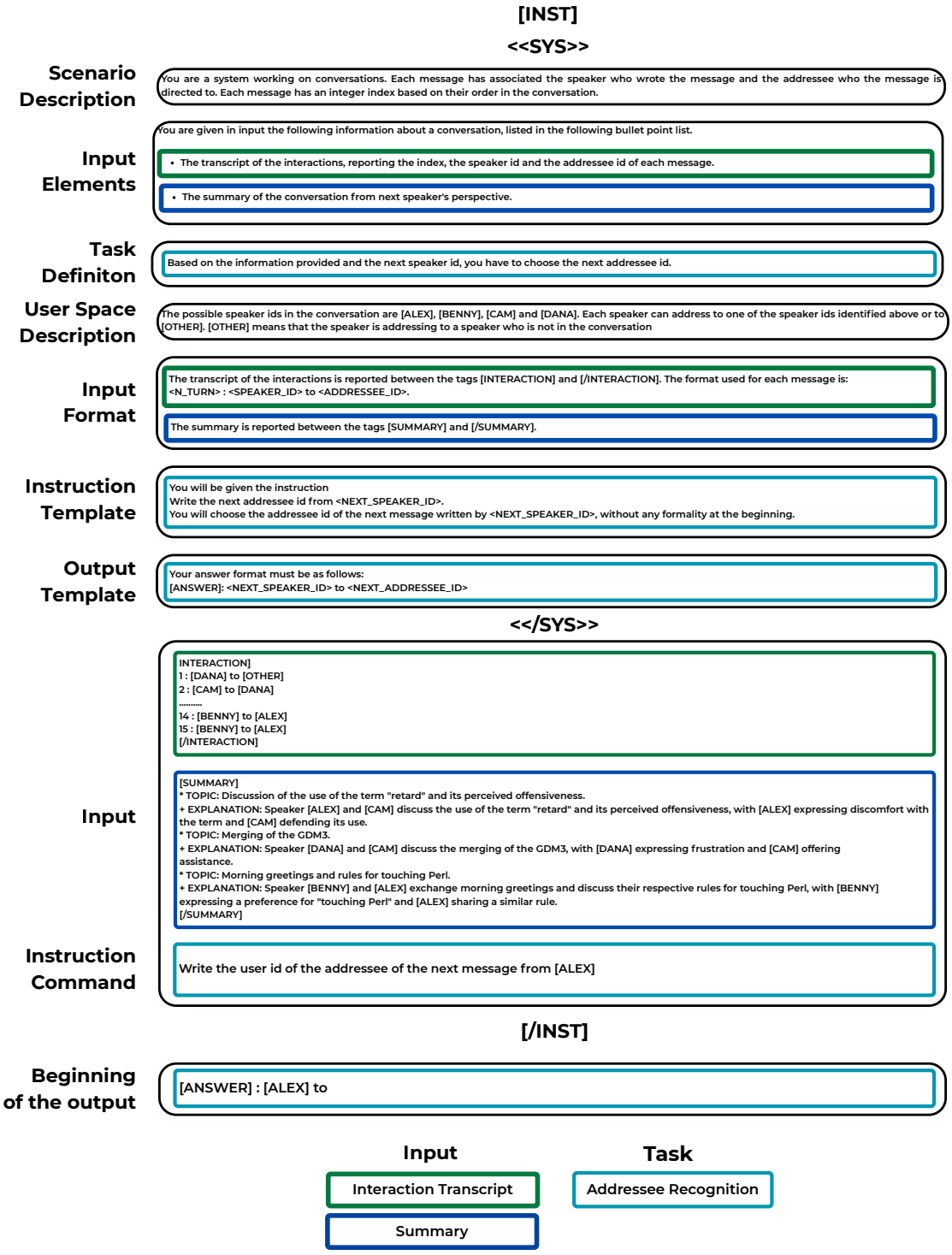

Figure B.10: Example of a complete prompt for the addressee recognition task, using Interaction Transcript and Conversation Summary as input information.

Appendix C

Generating Conversations

C.1 System prompts

We report in Figure C.1 and Figure C.2 respectively the two versions of the `Task Description` and the two versions of the `Output Format` (same structure, just different example) for the One-Long (OL) generation strategy. In Figure C.3 and Figure C.4 we report the same for the Turn-by-Turn (TT) generation strategy.

For both strategies, the three System Prompts are obtained by concatenating: (I.) `Task Description 1` + `Output Format 1`; (II.) `Task Description 2` + `Output Format 1`; (III.) `Task Description 1` + `Output Format 2`. In the System Prompt, the assignment of stance among the speakers has 6 possible distributions (pro-topic vs counter-topic): 2 vs 2, 3 vs 2, 2 vs 3, 2 vs 4, 3 vs 3, 4 vs 2.

C.2 Conversations with high Similarity scores between conversations.

In Figure C.5 and Figure C.6 we report two pairs of conversations generated with `Llama3.1`, One-Long strategy (i.e., the combination with highest Repetition Rate). We report in Figure C.5 the pair of conversations with highest Semantic Coherence (equal to 0.85) and in Figure C.6 the pair of conversations with highest String Similarity (equal to 99.4) from the pool of MPCs generated by the `LLama3.1`-OL combination. We observe that, despite the very high similarity scores, the two conversations are still clearly different, ensuring a good variability in conversations generated with the same prompt.

C.3 Technical Report

All the experiments have been run on Ampere A40 GPUs, which present 48GB of VRAM. We used the vLLM library (Kwon et al., 2023) for speeding up the inference time, namely the version 0.6.6. In Table C.1 we report the links to the models and the repositories we used. As hyperparameters, we use temperature 0.7, mixed top p and top k decoding with $p = 0.9$ and $k = 40$.

In Table C.2 we report the computational time to generate the full amount of 102 600 synthetic MPCs for each model-generation strategy combination. `OLMo2` is the model requiring more time (close to `Ministral` in One-Long strategy and definitely higher in Turn-by-Turn). In terms of strategies, as expected, the Turn-by-Turn is the one requiring more time, going from $\times 4.05$ times more in `Ministral` to $\times 7.82$ times more in `Qwen2.5`.

In order to compute the structural metrics, we used the tools from the `NetworkX`[1] library (Hagberg et al., 2008). Instead, for computing the Krippendorf-alpha we used the implementation from Castro (2017) for interval data. For the Spearman's correlation, we used the SciPy library (Virtanen et al., 2020). For the human evaluation, we used Argilla[2] for creating the annotation platform and we used the User Interface provided on HuggingFace spaces.[3] For the LLM-as-a-judge evaluation, we employed the OpenAI API for reasoning models.[4] We also used the official guide to perform prompt refinement.[5] Our results can be rerun by paying less than $20.00. We used both ChatGPT[6] and Copilot[7] for help in the coding process.

C.4 Guidelines for Human Evaluation

For human annotation, we designed a careful documentation to be used as guidelines. We report in Figure C.7 the introduction, in Figure C.8 the Platform Description (with a screenshot of the view), in Figure C.9 the Scores Description. One annotator performed his task as part of an internship, while the second annotator is regularly employed at the authors' institution. The effort required, on average, 2.5 hours for a batch of 32 items.

[1] `https://networkx.org/documentation/stable/`
[2] `https://github.com/argilla-io/argilla/`
[3] `https://huggingface.co/argilla`
[4] `https://platform.openai.com/docs/guides/text-generation`
[5] `https://platform.openai.com/docs/guides/prompt-generation`
[6] `https://openai.com/index/chatgpt/`
[7] `https://github.com/features/copilot`

Model	Repository
`Llama-3.1-8B`	`https://huggingface.co/meta-llama/Llama-3.1-8B`
`Qwen2.5-7B-Instruct`	`https://huggingface.co/Qwen/Qwen2.5-7B-Instruct`
`Ministral-8B-Instruct`	`https://huggingface.co/mistralai/Ministral-8B-Instruct-2410`
`OLMo2-7B-Instruct`	`https://huggingface.co/allenai/OLMo-2-1124-7B-Instruct`

Model	Context length
`Llama-3.1-8B`	128k
`Qwen2.5-7B-Instruct`	131k
`Ministral-8B-Instruct`	131k
`OLMo2-7B-Instruct`	4k

Table C.1: Upper - repositories of the model used for generating the synthetic MPCs. Lower - context length of each model.

Model	**Llama3.1**	**Qwen2.5**	**Ministr.**	**OLMo2**
OL	1485	1128	2553	2446
TT	11540	8824	10335	16580

Table C.2: Computational time for each model-strategy combination (in minutes).

OL - Task Description 1

You are a helpful and honest assistant. Please, respond concisely and truthfully.

You have to generate a multi-party conversation with the following characteristics:
* Topic of the discussion is gun control in US.
* Among 5 speakers.
* 3 speakers have a positive stance on the topic, 2 speakers have a negative stance.
* 15 messages in total.
* Avoid consecutive messages from the same speaker.
* Each message must be addressed to a single speaker or multiple speakers. Ensure to have a variety in the number of addressees among the messages.
* The first message is addressed to everyone.
* Each message has a maximum of 50 words.

For each message, you have to generate both the speaker and the addressee. The addressee cannot be the speaker.

OL - Task Description 2

You are a helpful and honest assistant. Please, respond concisely and truthfully.

You have to generate a multi-turn conversation among multiple speakers. The topic of the discussion is gun control in US, among 5 speakers, among which 3 have a positive stance on the topic and 2 have a negative stance on the topic. The conversation will have 15 messages in total.
Please Avoid consecutive messages from the same speaker.

Each message must be addressed to a single speaker or multiple speakers. Ensure to have a variety in the number of addressees among the messages. The first message is addressed to everyone. Ensure to have a variety in the number of addressees among the messages. Each message has a maximum of 50 words.

For each message, you have to generate both the speaker and the addressee. The addressee cannot be the speaker.

Figure C.1: The two versions of `Task Description` for the One-Long generation strategy

OL- Output Format 1

```
The output must follow the json format:
{ "conversation": [
  {
      "id" : 1,
      "speaker": speaker,
      "message": message,
      "addressee": [addressee1, addressee2]
  },
  {
      "id" : 2,
      "speaker": speaker,
      "message": message,
      "addressee": [addressee1]
  },
  {
      "id" : 3,
      "speaker": speaker,
      "message": message,
      "addressee": [addressee1, addressee2, addressee3]
  }
],
"speakers:[
  {
      "name": speaker1,
      "stance": stance_speaker_1,
  },
  {
      "name": speaker2,
      "stance": stance_speaker_2,
  },
  {
      "name": speaker3,
      "stance": stance_speaker_3,
  },
  {
      "name": speaker4,
      "stance": stance_speaker_4,
  }
]
}

The "stance" can be only "positive" or "negative".

You will be given the command:

Generate a conversation.

Be concise.
Don't give further details.
```

OL- Output Format 2

```
The output must follow the json format:
{ "conversation": [
  {
      "id" : 1,
      "speaker": speaker,
      "message": message,
      "addressee": [addressee1, addressee2]
  },
  {
      "id" : 2,
      "speaker": speaker,
      "message": message,
      "addressee": [addressee1]
  },
  {
      "id" : 3,
      "speaker": speaker,
      "message": message,
      "addressee": [address1, addressee2, addressee3]
  },
  {
      "id" : 4,
      "speaker": speaker,
      "message": message,
      "addressee": [addressee1, addressee2]
  },
  {
      "id" : 5,
      "speaker": speaker,
      "message": message,
          "addressee": [addressee1, addressee2, addressee3,
addressee4]
  },
  {
      "id" : 6,
      "speaker": speaker,
      "message": message,
      "addressee": [addressee1]
  }
],
"speakers:[
  {
      "name": speaker1,
      "stance": stance_speaker_1,
  },
  {
      "name": speaker2,
      "stance": stance_speaker_2,
  },
  {
      "name": speaker3,
      "stance": stance_speaker_3,
  },
  {
      "name": speaker4,
      "stance": stance_speaker_4,
  },
  {
      "name": speaker5,
      "stance": stance_speaker_5,
  },
  {
      "name": speaker6,
      "stance": stance_speaker_6,
  }
]
}
The "stance" can be only "positive" or "negative".

You will be given the command:

Generate a conversation.

Be concise.
Don't give further details.
```

Figure C.2: The two examples of `Output Format` for the One-Long generation strategy

TT - Task Description 1

You are a helpful and honest assistant. Please, respond concisely and truthfully.

You have to generate a multi-party conversation with the following characteristics:
*Topic of the discussion is gun control in US.
*Among 5 speakers.
*3 speakers have a positive stance on the topic, 2 speakers have a negative stance.
* 15 messages in total.
* Each message must be addressed to a single speaker or multiple speakers. Ensure to have a variety in the number of addressees among the messages.
* The first message is addressed to everyone.
* Each message has a maximum of 50 words.

You will be given three possible tasks:

* Generate a speaker;

* Choose the next turn interactions;

* Generate the next message.

TT - Task Description 2

You are a helpful and honest assistant. Please, respond concisely and truthfully.

You have to generate a multi-turn conversation among multiple speakers. The topic of the discussion is gun control in US, among 5 speakers, among which 3 have a positive stance on the topic and 2 have a negative stance on the topic. The conversation will have 15 messages in total.
Please Avoid consecutive messages from the same speaker.

Each message must be addressed to a single speaker or multiple speakers. Ensure to have a variety in the number of addressees among the messages. The first message is addressed to everyone. Ensure to have a variety in the number of addressees among the messages. Each message has a maximum of 50 words.

You will be given three possible tasks:

* Generate a speaker;

* Choose the next turn interactions;

* Generate the next message.

Figure C.3: The two versions of `Task Description` for the Turn-by-Turn generation strategy

TT- Output Format 1

When you will be asked to generate a speaker, you have to generate a speaker name and its stance, following the json format:

```
{
    "name": speaker,
    "stance": stance_speaker,
}
```

Ensure to respect the number of speakers with positive and negative stance.

The "stance" can be only "positive" or "negative".

When you will be asked to choose the next turn interactions, you have to choose the next speaker of the next turn and the next addressees of the next turn among the previously generated speakers, following the json format:

```
{
    "speaker": speaker,
    "addressee": [addressee1, addressee2]
}
```

When you will be asked to generate the next message, you have to generate the next message based on the interactions decided in the previous message and consistent with the user stance, following the json format:

```
{
    "message": message,
}
```

Be concise.
Don't give further details.

TT- Output Format 2

When you will be asked to generate a speaker, you have to generate 1) a speaker name and 2) its stance, which can only be positive or negative, following the json format:

```
{
    "name": speaker_name,
    "stance": speaker_stance,
}
```

Ensure to respect the number of speakers with positive and negative stance.

When you will be asked to choose the next turn interactions, you will output the speaker of the next turn and the addressees of the next turn among the previously generated speakers, following the json format:

```
{
    "speaker": speaker_name,
    "addressee": [addressee1, addressee2, addressee3]
}
```

When you will be asked to generate the next message, you have to output the message based on the interactions decided in the previous message and consistent with the user stance, following the json format:

```
{
    "message": message,
}
```

Be concise.
Don't give further details.

Figure C.4: The two examples of `Output Format` for the Turn-by-Turn generation strategy

High Semantic Coherence (0.85)

I think zoos are essential for conservation and education. What do you think?

I'm not sure about zoos. They can be inhumane and don't always prioritize animal welfare.

I agree with Speaker1. Zoos help protect endangered species and raise awareness about conservation.

I've seen zoos provide a safe space for injured animals to recover and thrive. They're not just for entertainment.

But don't zoos contribute to animal exploitation? Shouldn't we focus on preserving animals in their natural habitats?

I think zoos can be a starting point for people to learn about wildlife and develop a love for nature.

That's a great point, Speaker6. Zoos can inspire a sense of wonder and responsibility towards the natural world.

But what about the ethics of keeping wild animals in captivity? It's just not natural.

Zoos also support breeding programs for endangered species, which can help increase populations and eventually release animals back into the wild.

I agree with Speaker3. Zoos can be a vital tool in conservation efforts, especially for species that are critically endangered.

But what about the conditions in some zoos? Aren't they often subpar and inhumane?

Not all zoos are created equal. Some are well-designed and prioritize animal welfare.

I think zoos can be a valuable resource for education and conservation, but we need to ensure they're done responsibly.

I still have concerns about the ethics of zoos, but I'm willing to consider the benefits.

Let's focus on supporting responsible zoos that prioritize animal welfare and conservation.

I think zoos play a vital role in conservation and education.

I disagree, zoos are a form of animal captivity and can be inhumane.

Zoos also support research and breeding programs for endangered species.

But what about the ethics of keeping wild animals in captivity for our entertainment?

Zoos provide a safe environment for both animals and visitors, promoting education and appreciation.

Zoos also contribute to the conservation of species through sustainable practices.

I still think zoos are a form of exploitation, prioritizing human interests over animal welfare.

Some zoos are doing great work, but we need to ensure their standards are high and consistent.

Agreed, better regulation is necessary to ensure zoos are serving the greater good.

I'd like to see more transparency and accountability from zoos regarding animal care.

I think zoos can be a valuable resource for scientific research and education.

We should support zoos that prioritize animal welfare and conservation.

I still believe zoos are not the answer to conservation and education.

Let's focus on supporting zoos that are making a positive impact.

Agreed, let's promote zoos that prioritize animal welfare and conservation.

Figure C.5: Conversations with highest Semantic Coherence from `Llama3.1`-OL.

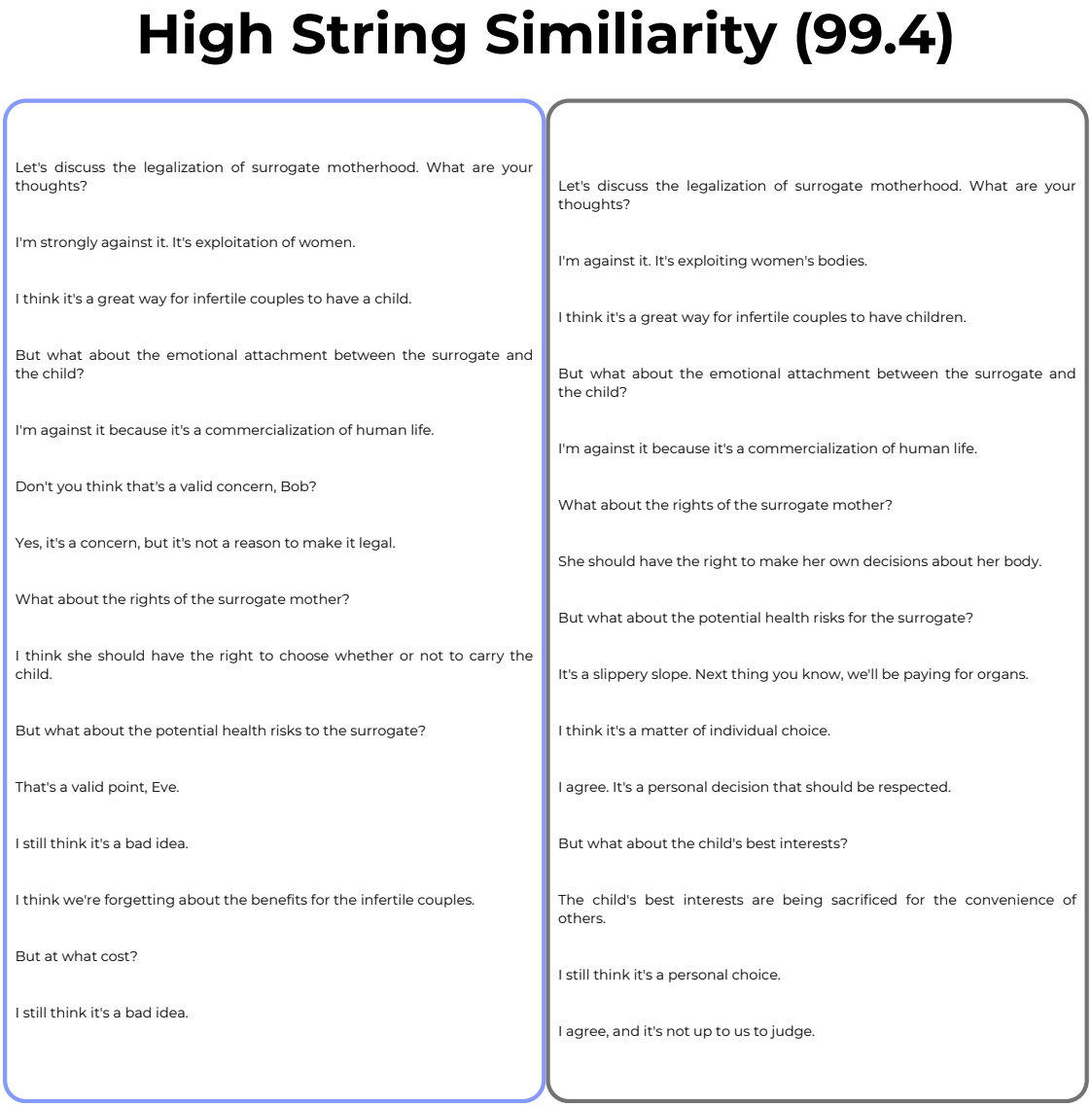

Figure C.6: Conversations with highest String Similarity from `Llama3.1`-OL.

Annotation Guidelines for Multi-Party Conversations
You will annotate multi-party conversations involving from **4 to 6 users**, each discussing a specific topic. The **topic** and each user's **initial stance** will be provided.

Key Concepts to Understand

1. Addressee
The addressee refers to the **intended recipient** of a message in the conversation.
An addressee can be:

- **Directly and explicitly addressed** (e.g., a user's name or a clear reference is mentioned).
- **Implied but inferred from context** (e.g., the message implicitly responds to someone's argument).
- **Multiple users at once** (e.g., the message addresses the group).

Important: The addressee is distinct from an "**overhearer**": an overhearer is a participant who reads the message but is **not the intended recipient**.
Note: Simply reporting the name of someone (e.g., "I agree/disagree with X") does not automatically mean that **X is the addressee**. Always consider the **context** to identify the correct addressee.

2. Stance
The stance reflects the **speaker's opinion about the topic**, i.e. they can either support or counter a topic.

- Example 1:
  - Topic: ban of mandatory military service
  - Pro: support the ban of mandatory military service
  - Con: counter the ban of mandatory military service
- Example 2:
  - Topic: mandatory military service
  - Pro: support the mandatory military service
  - Con: counter the mandatory military service
- Speakers **can change** their stance during the conversation, but such changes must arise logically through the course of argumentation and counterargumentation.

Illogical or abrupt stance changes that lack reasoning within the discussion are not acceptable

Figure C.7: Overview of guidelines for human evaluation

How Conversations Are Presented

1. **Chronological Order**
- The conversation will be displayed in the order the messages were exchanged.

2. **Topic Information**
- Before the first message, the **topic of the conversation** will be shown for context.

3. **Message Display**
- Each message appears in a **colored box** with the **speaker's name** clearly labeled.
- On the **right side of each message box**, the **addressees** of the message are listed. These addressee names will also appear in colored boxes corresponding to their assigned colors.

4. **Speaker Color Coding**
- Each speaker has a **consistent color** assigned to all their messages and addressee labels.
- **Hot colors** (e.g., red, orange) indicate speakers with an initial **countertopic** stance (opposition to the topic)
- **Cold colors** (e.g., blue, green) indicate speakers with an initial **protopic stance** (support for the topic).
- Remember, the stance should evolve logically through the conversation, but the color assigned will remain the same.

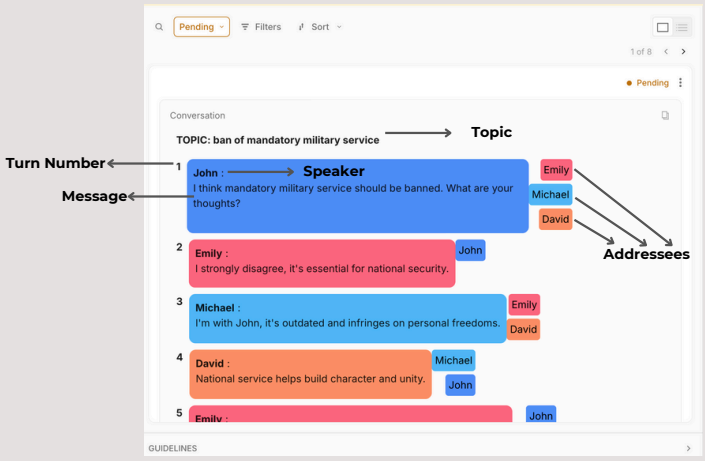

Figure C.8: Platform description from human evaluation guidelines

Annotation Task Overview
You will evaluate several aspects of a **multi-party conversation** based on the definitions provided. Each aspect will be rated on a specific scale.

Evaluation Criteria
1. Naturalness
- **Definition**: Assess the overall quality of the conversation's flow, tone, and word choice.
- **Scale**:
  - **1**: Completely unnatural – disjointed, incomprehensible.
  - **2**: Mostly unnatural – awkward, robotic, hard to follow.
  - **3**: Moderately natural – clear but with noticeable unnatural elements.
  - **4**: Mostly natural – fluent, minor issues, nearly native-like.
  - **5**: Fully natural – perfect flow, tone, and word choice, native-level..

2. Argumentability
- **Definition**: Rate how well the conversation presents reasoned and well-argued positions.
- **Scale**:
  - **1**: Total lack of arguments – The conversation has no argumentative character.
  - **2**: Insufficient arguments – Mostly empty and generic assertions are presented.
  - **3**: Mediocre arguments – Arguments are few or poorly reasoned, and many messages lack argumentative depth.
  - **4**: Good arguments – Messages generally express sufficiently reasoned positions, with rare exceptions.
  - **5**: Excellent arguments – All messages present well-reasoned arguments.

3. Speaker Stance Consistency
- **Definition**: Evaluate whether all speakers maintain the stance assigned at the beginning of the conversation. If the stance is uncertain, it does not mean the speaker is inconsistent with their assigned stance.
- **Scale**:
  - **1**: Completely inconsistent – all speakers fail to start with their assigned stance.
  - **2**: Mostly inconsistent – frequent misalignment with assigned stances at the start.
  - **3**: Moderately consistent – some speakers begin with their assigned stance, but others deviate.
  - **4**: Mostly consistent – rare deviations, most speakers start with their assigned stance.
  - **5**: Fully consistent – all speakers clearly and strictly start with their assigned stance.

4. Speaker Stance Evolution
- **Definition**: Assess whether each speaker demonstrates a realistic and logical evolution of their stance during the conversation or keep logically their stance. If stance doesn't change at all, the evolution is considered totally realistic.
- **Scale**:
  - **1**: Completely illogical – all stance changes are unrealistic, inconsistent, or poorly justified.
  - **2**: Mostly illogical – frequent unrealistic or weakly justified changes in stance.
  - **3**: Moderately plausible – some stance changes are logical, others feel forced or unclear.
  - **4**: Mostly plausible – most stance changes are logical, with a few minor inconsistencies.
  - **5**: Fully plausible – all stance changes are realistic, logical, and follow a natural progression.

5. Addressee Correctness
- **Definition**: Evaluate whether the assigned addressees align with the conversation context and are logically appropriate. A correct addressee is one that fits the intended recipient(s) based on the content of the message.
- **Scale**:
  - **1**: Completely Incorrect - addressees are random, irrelevant, or inappropriate.
  - **2**: Mostly Incorrect - frequent misalignments with the intended context or message.
  - **3**: Moderately Correct - some addressees are appropriate, but others are misaligned or unclear.
  - **4**: Mostly Correct - addressees are appropriate in most cases, with only minor errors.
  - **5**: Fully Correct - all addressees are logical and perfectly aligned with the conversation context.

6. Addressee Preciseness
- **Definition**: Assess whether addressees are precise and contextually appropriate. Messages should target the smallest relevant group or individuals, avoiding unnecessary use of "everyone" unless truly relevant to all.
- **Scale**:
  - **1**: Completely Imprecise - addressees are mostly irrelevant or overly broad, frequently defaulting to "everyone."
  - **2**: Mostly Imprecise - addressees are often too broad, with few attempts at precision.
  - **3**: Moderately Precise - addressees are sometimes appropriate but often lack precision, with overuse of "everyone."
  - **4**: Mostly Precise - addressees are usually precise, with only minor lapses in targeting.
  - **5**: Fully Precise - addressees are always the most precise and contextually appropriate, using "everyone" only when necessary.

Figure C.9: Description of the scores from the human evaluation guidelines

Appendix D

LLMberjack: Human-AI MPC creation

D.1 System Architecture

LLMberjack adopts a client–server design. The front-end (Vue.js + D3.js) handles visualization and user interaction, while a Python backend manages data structures, annotation logic, and controlled LLM calls. Components communicate through a RESTful API.

D.2 Data and File Management

Discussion files are represented as rooted trees whose nodes store message text, author metadata, and parent/child links. The system supports two file types: *discussion files* (full debate trees) and *draft files* (partially or fully linearized conversations). If a discussion file has an imperfect or noisy structure, users may invoke an LLM-assisted normalization step that reconstructs missing or inconsistent reply relations. When the `users` section is missing or incomplete, the system automatically extracts all speakers from the debate tree and regenerates the `users` list, assigning each participant a default profile with the description "This is a telegram user".

D.3 LLM integration

LLM calls follow fixed templates. For speaker profiling, the model receives the speaker profile to refine and a set of selected messages from the speakers serving as contextual evidence. Such contextual evidence corresponds either to the speaker's messages from the

emerging linearized conversation (if at least three messages from the speaker are written) or all nodes authored by that speaker in the original reply tree. For message refinement, the LLM is given the message to edit, the speaker profile, and the local conversational context, i.e., all turns preceding the one being refined.

For **tree-structure normalization**, we use a fully deterministic configuration (temperature = 0.0, top-p = 0.7, max tokens = 8192), ensuring stable, reproducible JSON reconstruction aligned with the expected schema. For **speaker-profile generation**, we adopt a more expressive setting (temperature = 1.2, top-p = 0.9, max tokens = 2048) to allow stylistic variability when synthesizing biographical descriptions. For **message refinement**, we employ a moderately stochastic configuration (temperature = 0.7, top-p = 0.9, max tokens = 512), balancing stylistic flexibility with semantic faithfulness to the draft. All calls use the same model (`Llama4-Maverick`) and a fixed seed (42). Complete templates and parameter settings are available in the project repository (see the `llm_calls.py` module).