Do LLMs suffer from Multi-Party Hangover? A Diagnostic Approach to Addressee Recognition and Response Selection in Conversations

Nicolò Penzo[1,2], **Maryam Sajedinia**[1,3], **Bruno Lepri**[1], **Sara Tonelli**[1], **Marco Guerini**[1]
[1] Fondazione Bruno Kessler, Italy
[2] University of Trento, Italy
[3] University of Turin, Italy
{npenzo,msajedinia,lepri,satonelli,guerini}@fbk.eu

Abstract

Assessing the performance of systems to classify Multi-Party Conversations (MPC) is challenging due to the interconnection between linguistic and structural characteristics of conversations. Conventional evaluation methods often overlook variances in model behavior across different levels of structural complexity on interaction graphs. In this work, we propose a methodological pipeline to investigate model performance across specific structural attributes of conversations. As a proof of concept we focus on Response Selection and Addressee Recognition tasks, to diagnose model weaknesses. To this end, we extract representative diagnostic subdatasets with a fixed number of users and a good structural variety from a large and open corpus of online MPCs. We further frame our work in terms of data minimization, avoiding the use of original usernames to preserve privacy, and propose alternatives to using original text messages. Results show that response selection relies more on the textual content of conversations, while addressee recognition requires capturing their structural dimension. Using an LLM in a zero-shot setting, we further highlight how sensitivity to prompt variations is task-dependent.

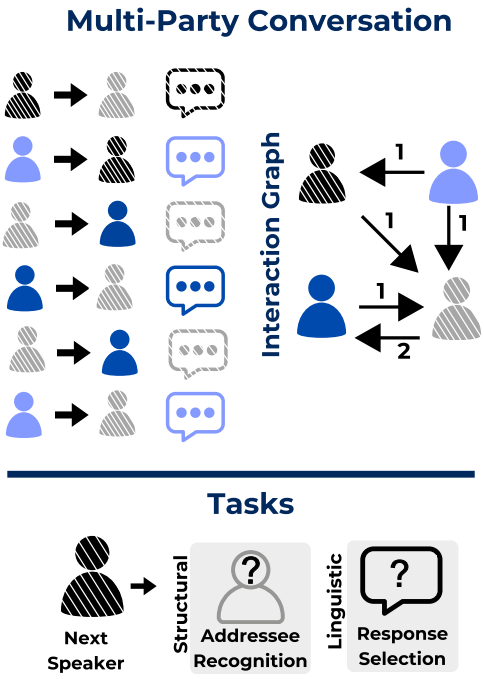

Figure 1: A graphical representation of the experiments. Each turn in a conversation includes a speaker, an addressee and a textual message. From the conversation, we extract the interaction graph to diagnose model capabilities by performing two tasks: addressee recognition and response selection.

1 Introduction

Multi-Party Conversations (MPCs) are multi-turn discussions involving more than two participants (Traum, 2003; Branigan, 2006), which are typical of online platforms such as Reddit or Twitter/X (Mahajan and Shaikh, 2021). Being able to capture the content of such discussions, when multiple users are involved and the conversation is composed of several turns, is a challenging task because both textual and structural information need to be modeled. Indeed, designing systems for MPC understanding is challenging not only because the *textual* dimension spans multiple turns, but also because we need to capture *structural* aspects, such as who writes to whom. Understanding how these two components should be integrated to classify MPC, and how effectively LLMs can contribute to this task, is still an open question.

In this paper, we examine the ability of an LLM to perform MPC classification tasks in a zero-shot setting as well as to capture relevant information from an existing conversation. Specifically, we address the tasks of *Response Selection* and *Addressee Recognition* and we use `Llama2-13b-chat` (Touvron et al., 2023) not only to classify the last turn of MPCs but also to summarise the previous conversation and to describe users, so that this information can be included in the prompts for zero-shot classification. Our choice of these two classifica-

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 11210–11233
November 12-16, 2024

tion tasks is based on the following key considerations: I. 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 structural and non-linguistic aspects; II. these tasks can be performed on any conversational corpus in any domain, without the need of manual annotation. This makes our framework widely applicable. In Figure 1 we report a graphical representation of an MPC, the pieces of information we retrieve and the tasks we perform.

Understanding the effects of conversation summarisation and user descriptions is important because they could make processing more efficient, replacing multiple turns with a more concise text representation. Furthermore, using summarisations 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).

Our research questions are therefore as follows:

> **RQ(1)**: How do LLMs perform in classification tasks involving MPCs in a zero-shot setting, using different input combinations to capture textual and structural information?
>
> **RQ(2)**: What is the model sensitivity to different prompt formulations when classifying MPCs?
>
> **RQ(3)**: How does structural complexity of the conversation affect classification performance?

To address **RQ(1)**, we evaluate `Llama2-13b-chat` on response selection and addressee recognition in a zero-shot scenario (Section 3). 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 and conversation flow. 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). We address also **RQ(2)** by designing prompts of different levels of verbosity for each combination and task. Finally **RQ(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 5). This allows us to analyse the connection between task scores and structural characteristics of MPCs.

The software to perform the experiments and the processed data are available on a dedicate Github repository.[1]

2 Related Work

Researchers have worked on MPC understanding tasks either by trying to model an entire conversation or by focusing on relations within the conversation (Gu et al., 2022b; Ganesh et al., 2023). Recent MPC understanding studies focus on response selection (RS) and addressee recognition (AR) tasks (Ouchi and Tsuboi, 2016; Zhang et al., 2018b) to compare different classification approaches. Indeed, RS is strictly related to textual (linguistic) information, while AR focuses on interaction information, thus permitting to analyse the performance of classification models from two different angles. However, both tasks can ideally benefit from cross-information between linguistic and interaction cues.

For both RS and AR, 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 GPT-4 (OpenAI et al., 2024) in MPCs, focusing only on the overall classification performance. Indeed, 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). Penzo et al. (2024) provided a first exploration of the role of conversation structure in stance detection, show-

[1] https://github.com/dhfbk/MPH

ing that it benefits classification only when large training data are available.

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.

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 RS and AR. The main difference is that, instead of focusing only on generic accuracy scores, we propose a diagnostic approach for evaluating models for MPC understanding. We use response selection and addressee recognition as proxy tasks and focus particularly on the contribution of structural information, by I. creating diagnostic datasets, each with a fixed number of users, and II. putting in relation the classification performance to specific network metrics (i.e., degree centrality and average outgoing weight of the speaker node).

3 Tasks

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 (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 (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, which introduces a user unseen in the conversation to check whether the classifier choice is fully random.

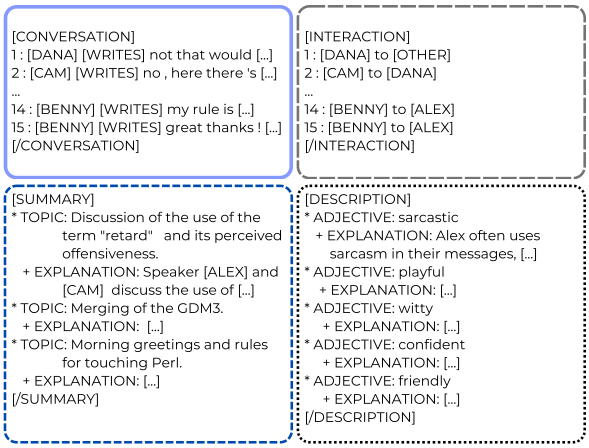

Figure 2: Example of the 4 possible conversation representations: I. Conversation Transcript (top left), II. Interaction Transcript (top right), III. Summary (bottom left) and IV. User Description (bottom right).

In both cases, the next speaker is given, and the classifier has to select what will be the content of the message (RS) or who will be the addressee (AR).

4 MPC Classification Workflow

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

4.1 Conversation Representation

The first step is modelling the input data to be included in the prompt used for classification. To analyse the contribution of contextual and structural information for RS and AR we identify four ways to model 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 structural 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 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 2.

These last two representations are meant to replace the actual discussion content, retaining only the most relevant information. This 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 (Penzo et al., 2024), so that it is not possible to identify which users are the same across different conversations (details in Appendix A).

4.2 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 and IV. addressee recognition. For creating prompts, we follow the guidelines provided by Meta[2].

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., 2023) of all candidate responses

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

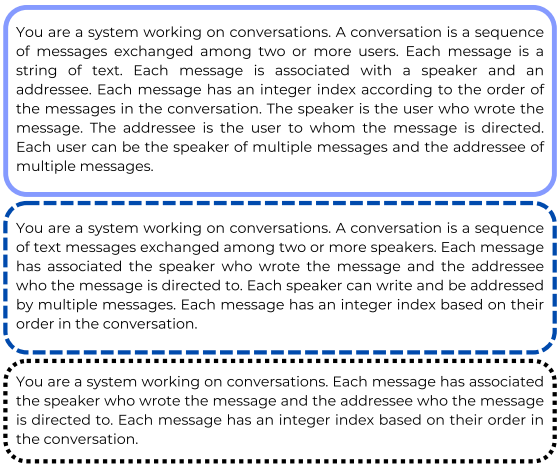

Figure 3: Example of the beginning of the system prompt in the three prompt schemes, from the most verbose (top) to the most concise (bottom).

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 word, whereas in our response selection task the candidates are sentences of variable length.

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]` $b\ r_i$

4.3 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 3 shows how the beginning of the system prompt changes across the three versions. More details and examples are provided in Appendix A.

5 Diagnostic Approach

To address the three research questions introduced in Section 1, 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 we report a schematic representation of our evaluation pipeline and the components involved.

5.1 Structural Information as Interaction Graph

Conversations present different structures depending on the discussion complexity and the speakers' involvement (Cogan et al., 2012; Coletto et al., 2017). To analyse the relation between interaction complexity and classification performance, we first identify two network metrics able to capture the structural complexity of MPCs. In the past, researchers have explored correlations between model performance and factors such as number of speakers and conversation length (Gu et al., 2023; Penzo et al., 2024), but to our knowledge network metrics to capture conversation complexity have never been considered before in this framework.

Given an MPC, its interaction graph can be modeled as an I. *unweighted undirected*; or II. *weighted directed* graph. In the unweighted undirected graph, each edge between two users simply indicates that they have interacted, without specifying the direction of the communication and the number of exchanged messages. The weighted directed graph instead includes directionality from the speaker to the addressee and a weight assigned to each edge, corresponding to the number of messages from the speaker to the addressee. From each conversation C we therefore extract both versions of the interaction graph, i.e. $G^{uw}_{ud}(C)$ and $G^{w}_{d}(C)$.

From the above interaction graph, we then derive two network metrics for the next speaker node, i.e. the degree centrality and the average weight of the outgoing edges (average outgoing weight). Specifically, the metrics are computed as follows:

Degree Centrality. Given an unweighted undirected graph $G^{uw}_{ud}(C) = (U, E)$, where U is the set of nodes and E is the set of edges, the degree centrality $deg(u)$ of a node $u \in U$ is the number of edges $e \in E$ incident in u, i.e. $e = (u, v)$ or $e = (v, u)$. In our setting, it represents the number of users the next speaker has interacted with.

Average Outgoing Weight. Given a weighted directed graph $G^{w}_{d}(C) = (U, E)$, where U is the set of nodes and E is the set of edges, the out-degree centrality $outdeg(u)$ of a node $u \in U$ is the number of outgoing edges $e \in E$ for which u is the originating node/speaker, i.e. $e = (u, v)$ and the weighted out-degree $outdeg_w(u)$ is the sum of the weights on such edges. So the average outgoing weight $w^{o}_{avg}(u) = outdeg_w(u)/outdeg(u)$ is the average weight of the edges $e \in E$ for which u is the originating node/speaker, i.e. $e = (u, v)$. In our setting, it represents the average number of messages sent to the users with whom the next speaker has interacted.

5.2 Diagnostic Datasets

To analyse the impact that different interaction structures have on RS and AR, 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 used 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 AR 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 *Ubuntu3/4/5/6*. 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 4.1 (details in Appendix A).

The resulting diagnostic datasets have respectively 1200, 635, 520 and 350 conversations. These

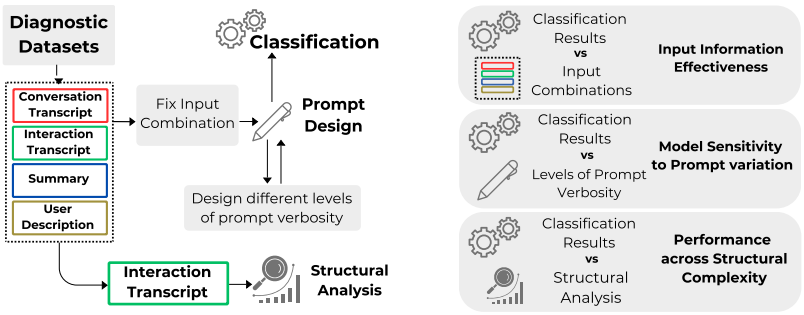

Figure 4: Schematic representation of our evaluation pipeline: on the left, the pipeline and the relation among the elements; on the right, the type of diagnostic evaluation we can perform.

datasets are used as test sets for evaluating RS and AR in a zero-shot setting.

6 Experiments

Given the four types of input presented in Section 4.1, 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+STRUCT); III. the interaction transcript and the conversation summary (STRUCT+SUMM); IV. the interaction transcript and the user descriptions (STRUCT+DESC); V. the interaction transcript, the conversation summary and the user descriptions (STRUCT+SUMM+DESC). For the AR task, we test a sixth combination, VI. STRUCT, which corresponds only to the interaction transcript. STRUCT is not relevant for RS since it does not include any linguistic information.

All combinations and prompt schemes are tested across the 4 diagnostic datasets.

7 Macro Results and Structural Evaluation

Macro-results on the best run.

In Figure 5, 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 the **AR task**, the number of classes (i.e. 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 label (see Section 3). For this reason, results across different Ubuntu subsets on AR should not be compared, and the lowest accuracy is achieved on Ubuntu 6, being its classification based on seven possible addressees.

In AR, the CONV+STRUCT and STRUCT combinations consistently perform best across all datasets. Instead, the CONV combination, serving as our 'text-only' baseline consistently shows the worst performance. 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+STRUCT) still ourperforms both alternatives.

In the **RS task**, the CONV and CONV+STRUCT combinations consistently perform the best across all datasets. Among the combinations with summary and/or user description, STRUCT+SUMM+DESC performs the best (in Ubuntu3/4) or extremely close to STRUCT+SUMM, which is the best in Ubuntu5/6. Finally, the STRUCT+DESC input combination yields the lowest classification performance for both tasks on all datasets, except for AR on Ubuntu3.

This analysis shows that the interaction transcript (i.e., the structural information) is fundamental for achieving the best result in AR. On the other hand, the conversation transcript is fundamental for achieving the best results in RS, which in fact is a more text-oriented task, based on information mainly available in the conversation itself. However, using summaries of conversation 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.

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

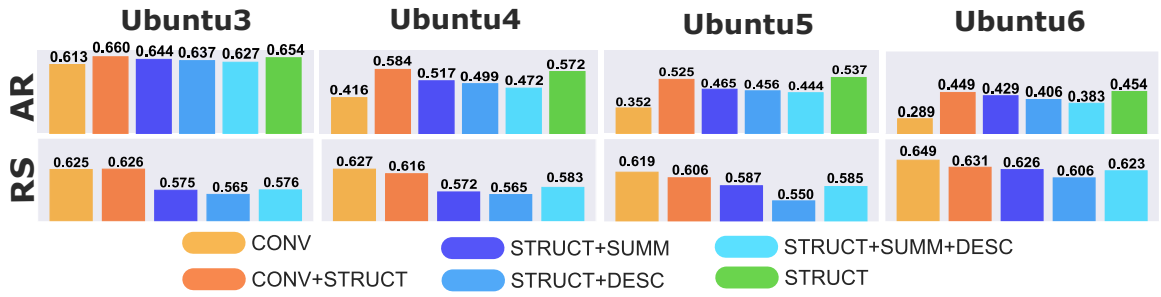

Figure 5: AR and RS macro-accuracy results (y axis), for each combination and for each dataset. The height of the columns represents the best macro result across the three prompt schemes. Note that for AR the number of classes on each Ubuntu subset changes, ranging from four (Ubuntu3) to seven (Ubuntu6), since the set of possible addressees includes the speakers involved in each conversation, plus the dummy label. For this reason, results across different Ubuntu subsets on AR should not be compared, and the lowest accuracy is achieved on Ubuntu6.

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.

COMBINAT.	T	U.3	U.4	U.5	U.6
CONV	AR	$2.7^{\diamond}$	0.6	$5.8^{\diamond}$	2.0
	RS	$0.8^{\diamond}$	$0.9^{\diamond}$	$0.8^{\diamond}$	0.6
CONV + STRUCT	AR	$7.1^{\diamond}$	$10.9^{\diamond}$	$4.5^{\diamond}$	$4.9^{\diamond}$
	RS	$0.4^{\diamond}$	0.6	1.1^{*}	0.8^{*}
STRUCT+ SUMM	AR	2.5^{*}	$6.5^{\diamond}$	3.6^{*}	6.7^{*}
	RS	0.8^{*}	1.0	1.7	$0.8^{\diamond}$
STRUCT+ DESC	AR	$2.7^{\diamond}$	$5.6^{\diamond}$	$4.8^{\diamond}$	$8.9^{\diamond}$
	RS	$2.1^{\diamond}$	0.9^{*}	1.3^{*}	1.6
STRUCT+ SUMM+DESC	AR	$1.5^{\diamond}$	$4.0^{\diamond}$	3.2^{*}	$6.0^{\diamond}$
	RS	$0.2^{\diamond}$	1.4	1.2	$0.6^{\diamond}$
STRUCT	AR	$5.6^{\diamond}$	$8.3^{\diamond}$	$8.5^{\diamond}$	$6.3^{\diamond}$

Table 1: Relative gap (%) between the best prompt result and the average, for each input combination and diagnostic dataset (UbuntuX is shortened as U.X), and for each task (i.e., AR and RS). We put a $\diamond$ when the best prompt is the verbose version, a $*$ when the medium version is the best and nothing when the best is the concise version.

In Table 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 AR task tend to be more sensitive to prompt formulation compared to the RS task, especially in the CONV+STRUCT combination. Indeed, the relative gap between the best run and the average results is remarkably larger for the AR task than for RS across all diagnostic datasets. Moreover, in the AR task, the STRUCT combination has similar prompt sensitivity to CONV+STRUCT.

Overall, we observe that for AR, where structural information is more relevant, classification performance tends to vary more with different prompt verbosity compared to RS, where linguistic information has a higher weight.

If we analyse 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 AR. For RS, instead, there is no evidence of benefit from using more or less verbose options.

Structural Evaluation. In Figure 6, we present how the best run for each input combination varies in relation to the two network metrics introduced in Section 5.1, i.e. degree centrality $deg(u)$ and average outgoing weight $w^{o}_{avg}(u)$, where u is the speaker node (across all 4 diagnostic datasets). More informally, $deg(u)$ (A in Figure 6, bottom) represents the number of users the next speaker has interacted with. Instead, $w^{o}_{avg}(u)$ (B in Figure 6, bottom) 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 AR task, combinations containing the interaction transcript (+STRUCT) 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 models (CONV+STRUCT and STRUCT) and others widens significantly at lower $deg(u)$ values. For example, while the STRUCT combination consistently ranks among the best in terms of macro-results, it is outperformed by (or comparable with) other combinations across all di-

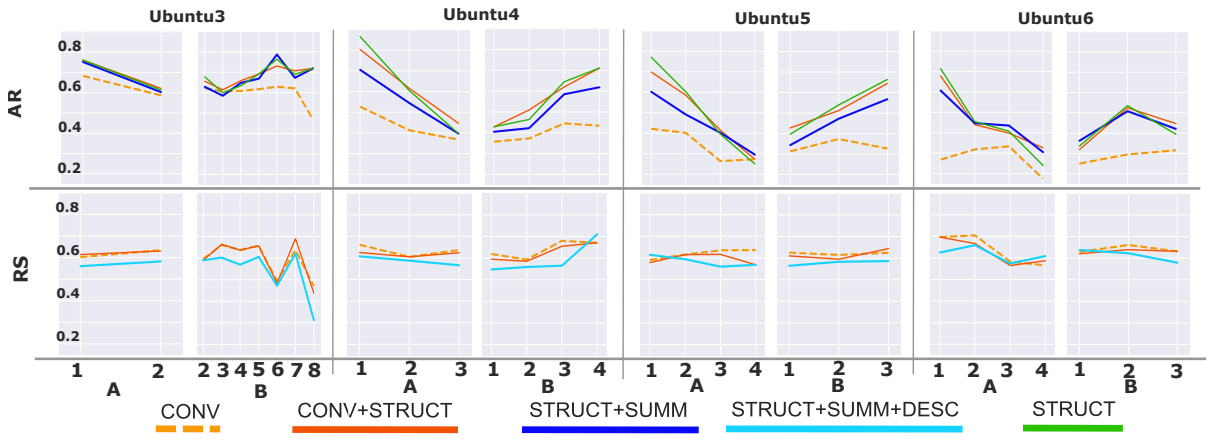

Figure 6: AR and RS accuracy results (y axis) for the different values of $deg(u)$ (A) and $w^o_{avg}(u)$ (B) of the speaker node u (x axis). We report the performance of the three best input combinations for each task, plus CONV in AR which serves as text-only baseline. $w^o_{avg}(u)$ is rounded at the closest integer number.

agnostic datasets as $deg(u)$ increases. This shows that using in the prompt STRUCT-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 spoke 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).

In the RS 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 RS task are not related to the structural dimension.

8 Discussion

Our comparative evaluation shows three main findings.

Input combination performance (RQ1). Regarding the best-performing combinations, in AR, STRUCT and CONV+STRUCT 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 RS, CONV and CONV+STRUCT consistently outperform other combinations, with 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. In the future, we plan to investigate this aspect with further analyses.

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 AR 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. Overall, user descriptions appear to be ineffective.

Prompt Verbosity (RQ2). Addressee recognition (AR), which benefits from structural information, shows greater sensitivity to prompts compared to response selection (RS), which is mostly a text-based task. This difference could be due to the similarity of RS to tasks used to pretrain LLMs. Indeed, RS 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 AR, probably because it helps the model in better capturing the structural information, which is crucial for this task. For RS, 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 (RQ3) Our structural analysis (Figure 6) reveals the limitations of relying solely on macro results, especially in the **AR task**.

In AR, if we consider the correlation between classification accuracy and $deg(u)$, especially in Ubuntu4/5/6, we observe that the performance gaps between the best input combination (i.e., STRUCT for Ubuntu3/4, CONV+STRUCT for Ubuntu5/6) and STRUCT+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 structural 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 STRUCT, 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 AR, 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 **RS 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 conversation 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 (AR and RS), we analysed 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 here.

9 Conclusions

In this study, 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 structure. 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 analyse how structural complexity interacts 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 MPCs, 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 AR and RS, 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 structural information in MPCs should be better analysed in current evaluations, merging contributions from NLP and the network science community.

10 Limitations

The findings presented in this work are based only on subsets of a single dataset, the Ubuntu Inter-

net Relay Chat corpus. This choice is due to the fact that many multi-party datasets lack sufficient variety in terms of structure and addressee labels, which are necessary for performing in-depth diagnostic analyses. Additionally, all the conversations in our dataset have the same length (15 turns), allowing us to exclude conversation length as a variable. During the development of this work, we took other datasets into account as possible candidates for our experiments, but when we analysed 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. However, for future research, it would be interesting to introduce other types of diagnostic datasets, for example extracted from different social media or dealing with a diverse set of topics. 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 generalisation of our approach.

11 Ethics Statement

In conducting this research, we prioritized the privacy and ethical management of data. Although the original dataset (UbuntuIRC) is freely distributed online and includes the original usernames, we chose to anonymize the data by replacing each username with a name from a set of ungendered names (details are provided in the Appendix A). This approach helps to protect the identity of the users and reduces potential biases associated with gendered names. Furthermore, we explored two alternative representations of the conversation transcripts to investigate the feasibility of working with different elements compared to the original messages. This is in line with current policies that encourage researchers to find methods that enhance user privacy and minimize biases. Future research could benefit from using summaries and/or user descriptions, which would allow the distribution of textual data while making it nearly impossible to train chatbots that could imitate specific users. Regarding reproducibility, our experiments were conducted in a zero-shot setting without fine-tuning. This ensures that our methodology can be replicated efficiently. Specifically, our experiments can be reproduced within a few hours on a single GPU with 48GB of VRAM and a batch size of one, using a model that is available online.

Acknowledgments

The work of BL was partially supported by the NextGenerationEU Horizon Europe Programme, grant number 101120237 - ELIAS and grant number 101120763 - TANGO. BL and ST were also supported by the PNRR project FAIR - Future AI Research (PE00000013). NP's activities are part of the network of excellence of the European Laboratory for Learning and Intelligent Systems (ELLIS).

References

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

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. 2017. Automatic controversy detection in social media: A content-independent motif-based approach. *Online Social Networks and Media*, 3-4:22–31.

Ananya Ganesh, Martha Palmer, and Katharina Kann. 2023. A survey of challenges and methods in the computational modeling of multi-party dialog. In *Proceedings of the 5th Workshop on NLP for Conversational AI (NLP4ConvAI 2023)*, pages 140–154, Toronto, Canada. Association for Computational Linguistics.

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 *IJCAI*, pages 5486–5493.

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.

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.

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.

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.

Siwon Kim, Sangdoo Yun, Hwaran Lee, Martin Gubri, Sungroh Yoon, and Seong Joon Oh. 2023. Propile: Probing privacy leakage in large language models. In *Advances in Neural Information Processing Systems*, volume 36, pages 20750–20762. Curran Associates, Inc.

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.

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.

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.

Daniela Occhipinti, Serra Sinem Tekiroglu, and Marco Guerini. 2023. Prodigy: a profile-based dialogue generation dataset. *Preprint*, arXiv:2311.05195.

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, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar,

Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 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.

Nicolò Penzo, Antonio Longa, Bruno Lepri, Sara Tonelli, and Marco Guerini. 2024. 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.

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*.

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.

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, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 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.

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.

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.

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).

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.

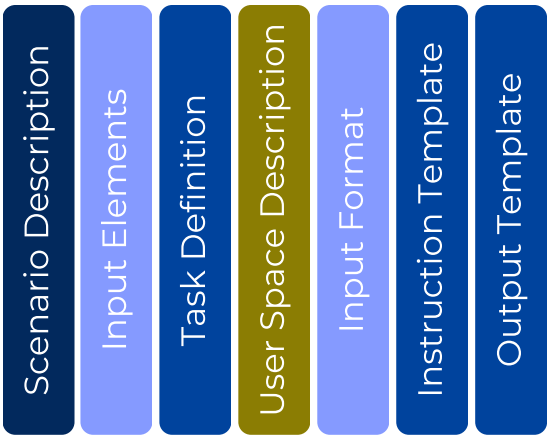

Figure 7: Graphical representation of the system prompt organization.

A Prompt schemes and combinations

In our experimental setup, we establish a fixed template for all system prompts, as shown in Figure 7, 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+STRUCT, STRUCT+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., AR, RS, summarisation, description); two sections, i.e. Input

Elements and Input Format, are constructed modularly based on the chosen input combination (e.g., CONV, STRUCT, SUMM, DESC). In Figure 8 we report how the different pieces of input information are related to each other.

There is an ongoing discussion about evaluating instruction-based models particularly considering the high sensitivity of their performance to different levels of prompt verbosity. 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.

We create three prompt schemes corresponding to three different levels of prompt verbosity. The first, reported in Figure 10 is extremely precise and detailed, the second (Figure 11) gives some concepts for granted and the third (Figure 12) is the most implicit. One clear example is in the Instruction Template for the AR task. It evolves from *"Write the user id of the addressee of the next message"*, to *"Write the addressee id of the next message"*, and finally to *"Write the next addressee id"*.

After creating the system prompt, we concatenate the input information and the instruction command, as shown in Figure 13 for generation tasks and in Figure 14 for classification tasks, to form the final prompt. An example of this process for the STRUCT+SUMM combination in the AR task is shown in Figure 15.

We identify the second dimension only for the generation of summary/user description task. Once we fix the prompt scheme, we test two output templates, as shown in Figure 16, while keeping all the other sections fixed. Since the results are similar (details in Section B.4) we mention only the first version for the arguments in the main body of the paper.

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.

B Task details and formalization

B.1 Formalization of an MPC

Given a conversation $C = (M, U)$, $M = \{m_1, m_2, ..., m_n\}$ is the set of chronologically ordered messages (message m_i appeared before m_j in the conversation if $i < j$) and $U = \{u_1, u_2, ..., u_p\}$ is the set of users occurred in C. Each message m_i is assigned a ordered pair (u_j, u_k) s.t. u_j is the speaker of m_i and u_k is the addressee of m_i, so $u_j = S(m_i)$ and $u_k = A(m_i)$.

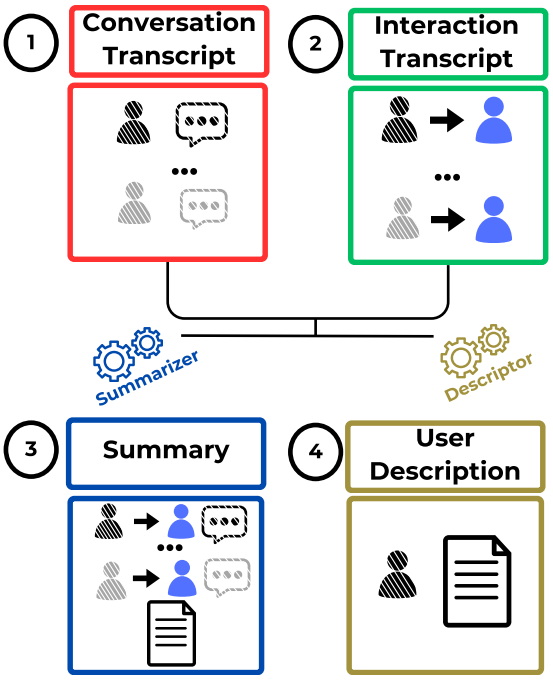

Figure 8: **Experimental setup**. First we create the conversation transcript (1) and the interaction transcript (2). From these, we extract the summary and the user description by using a specifically prompted LLM (3,4).

B.2 Classification using CPPL

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

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

according to the probability distribution of the model.

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

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

Analysing the correlation between the probability of the output target and the network metrics leads to the same conclusions obtained considering accuracy values.

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+STRUCT	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
STRUCT+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
STRUCT+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
STRUCT+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
STRUCT	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 2: Table of Accuracies in 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+STRUCT	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
STRUCT+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
STRUCT+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
STRUCT+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 3: Table of Accuracies in response selection across prompt schemes and input combinations

B.3 Results in detail

In this section we report all the results for each prompt scheme and combination, on both AR (Table 2) and RS (Table 3). It is the tabular version of the graphs in Figure 5, where only the best run is reported, and the most detailed version of Table 1, where we report the relative gap between the best run and the average among the 3 prompt schemes (from each cell, we extract only the gap).

B.4 Effect of different output templates in generating summary/user description prompt

In Figure 9 we report the results across diagnostic datasets and tasks for STRUCT+SUMM, STRUCT+DESC, and STRUCT+SUMM+DESC by averaging the results across the different prompt schemes but with the same output template. The average between the two different output templates does not differ much, with a maximum difference of 1.9% in the AR task-Ubuntu5 for STRUCT+SUMM+DESC, between the averages with same combination but different output templates. Nevertheless, we examine all combinations by averaging the results across the output template, obtaining results consistent with the ones presented in Section 7. We focus therefore on one output template for the sake of simplicity.

B.5 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[3]. We use Copilot[4] as a coding assistant and ChatGPT (OpenAI, 2022) as a writing assistant only to improve the style of the text.

[3] https://huggingface.co/meta-llama/Llama-2-13b-chat-hf

[4] https://github.com/features/copilot

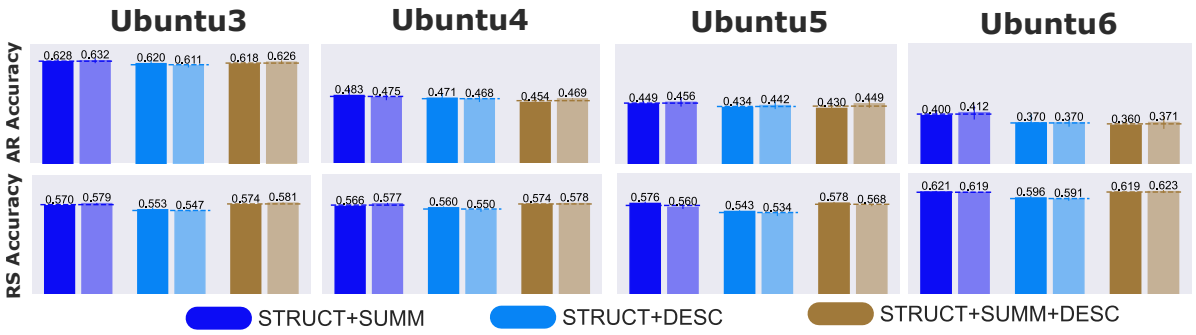

Figure 9: Results across the two generation of summary/user description output template.

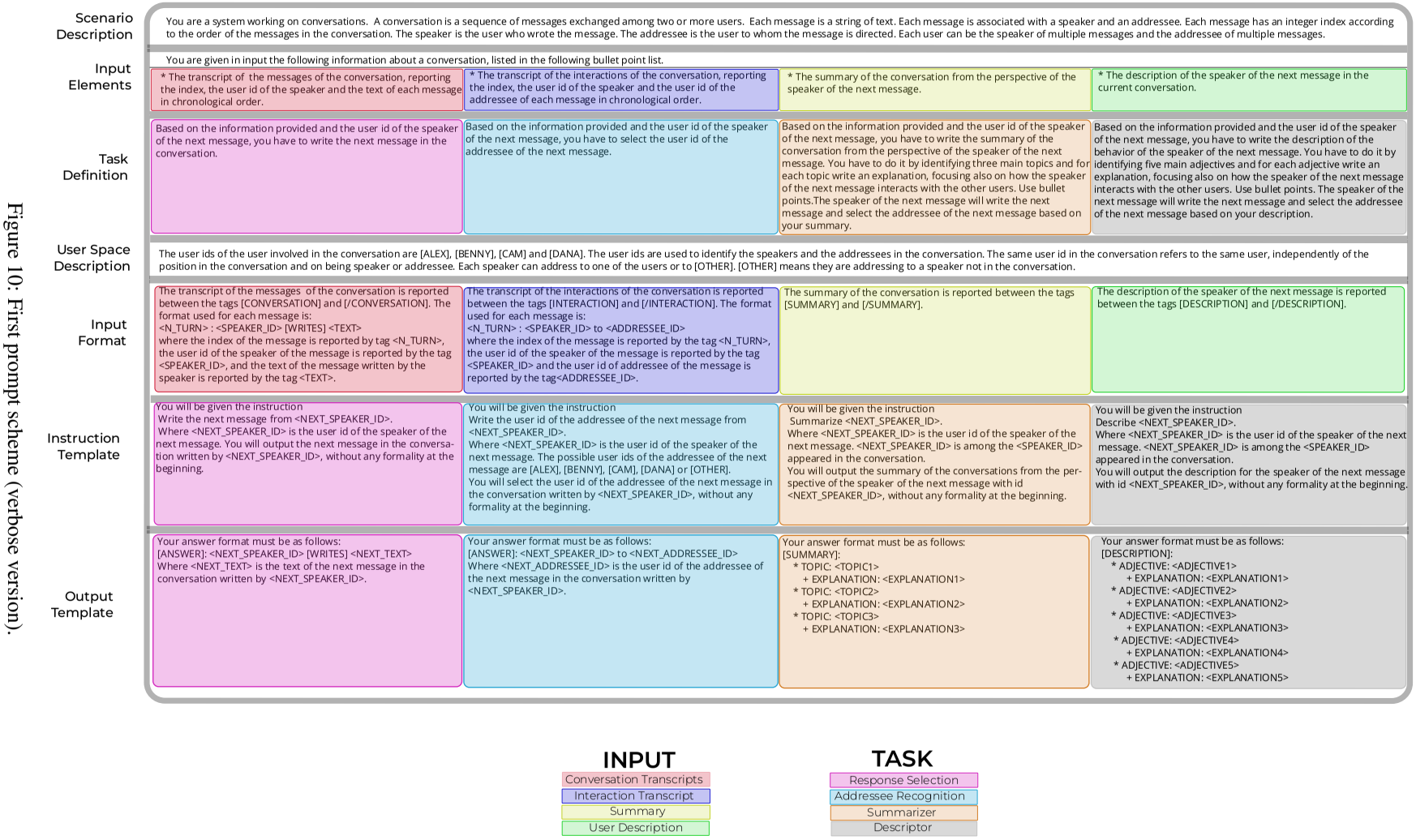

Figure 10: First prompt scheme (verbose version).

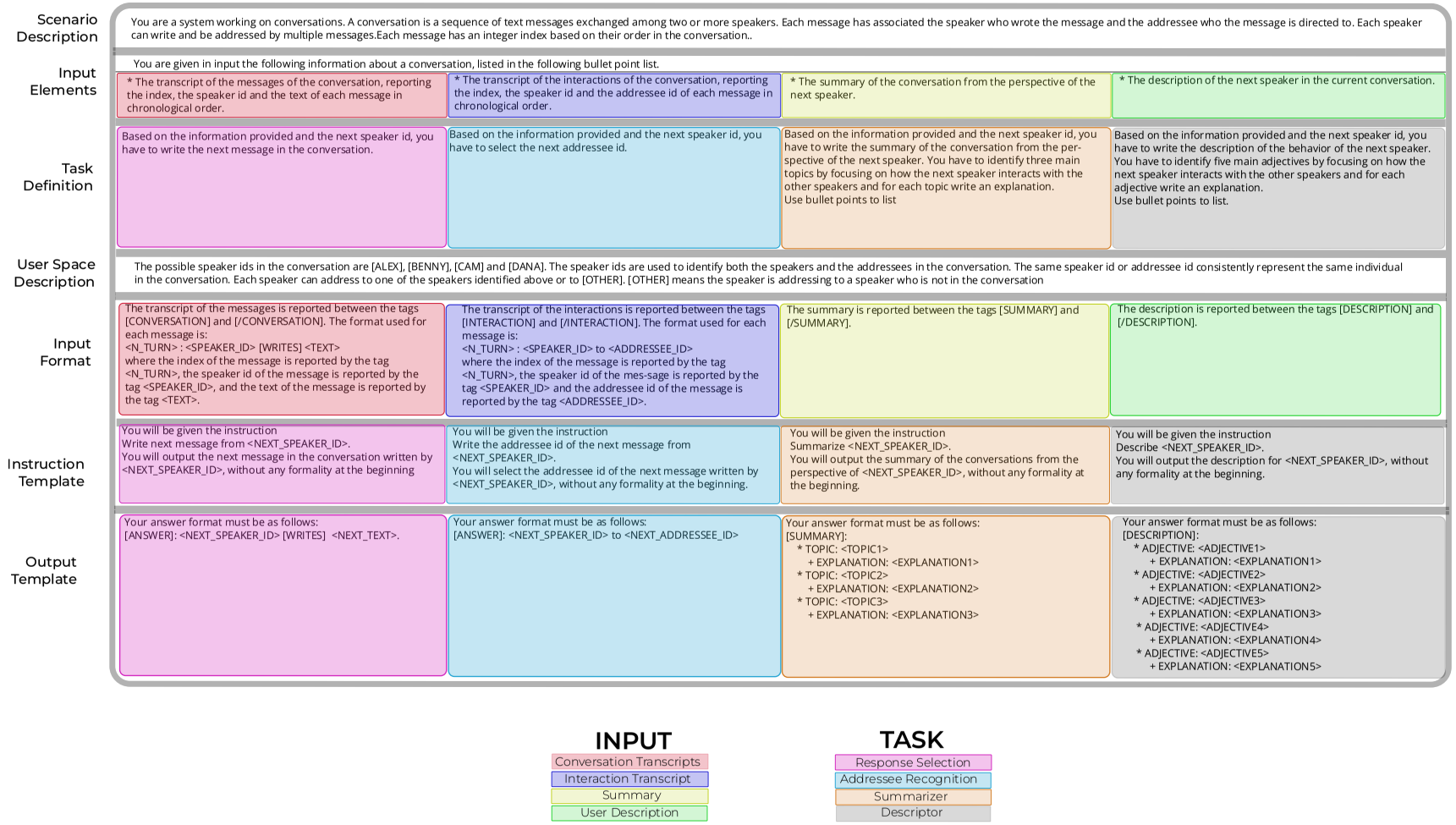

Figure 11: Second prompt scheme (medium version).

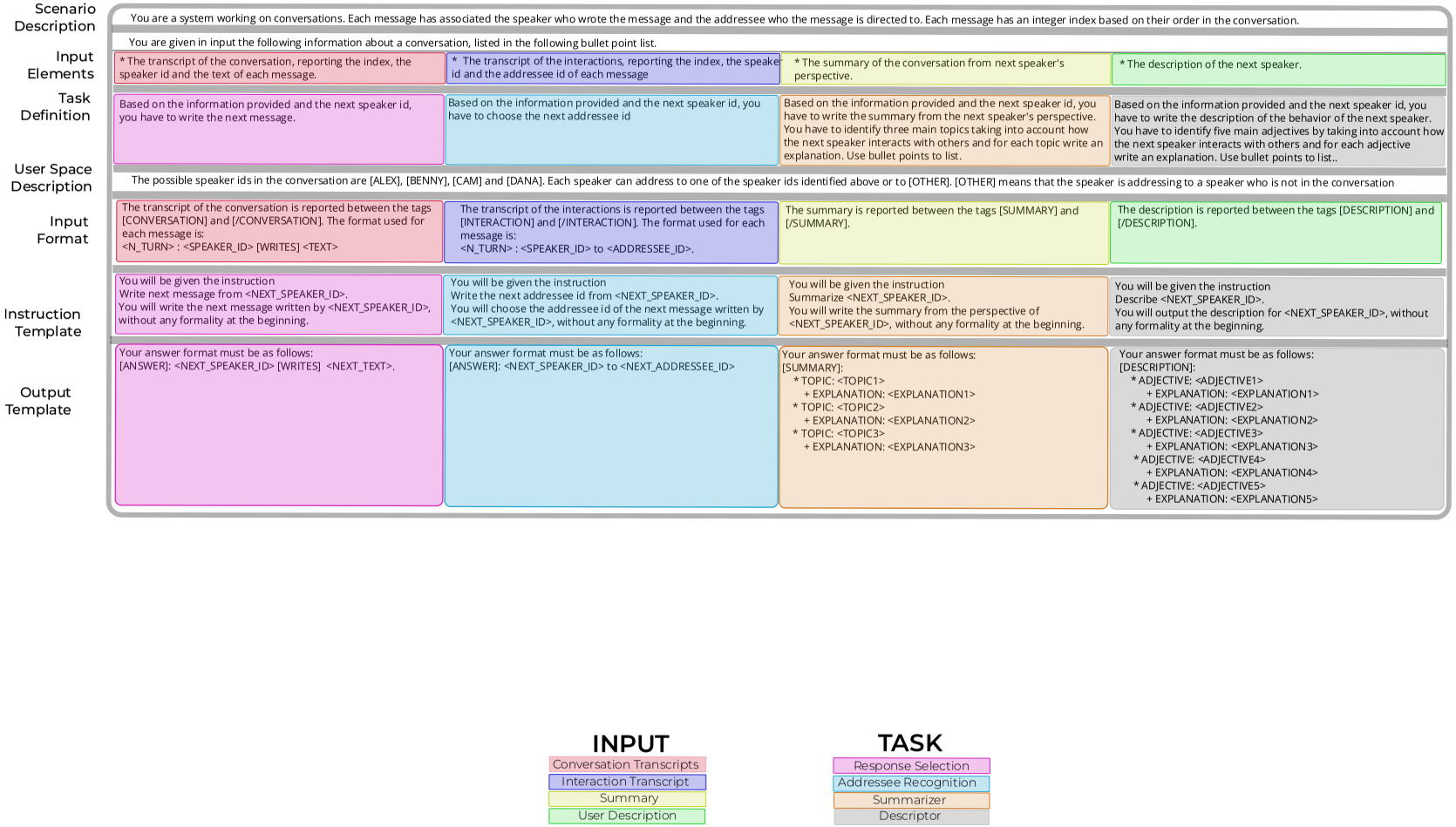

Figure 12: Third prompt scheme (concise version).

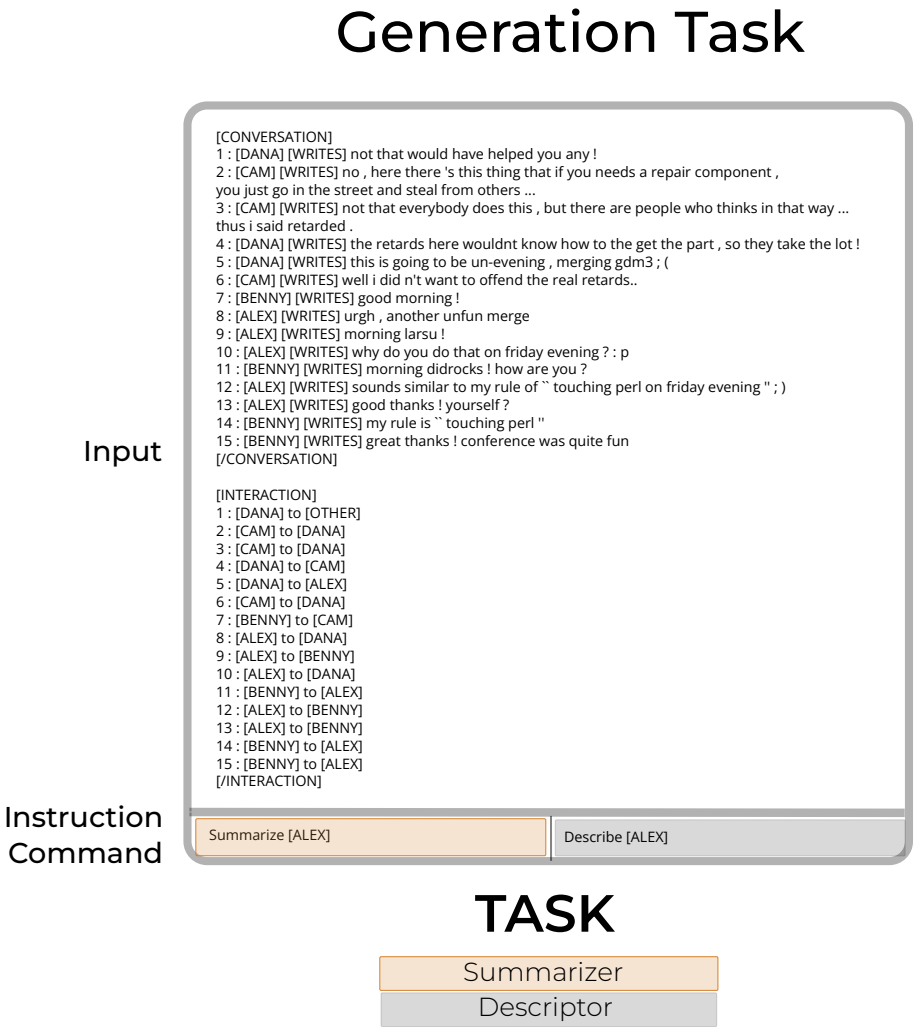

Figure 13: Input information and instruction command for generation.

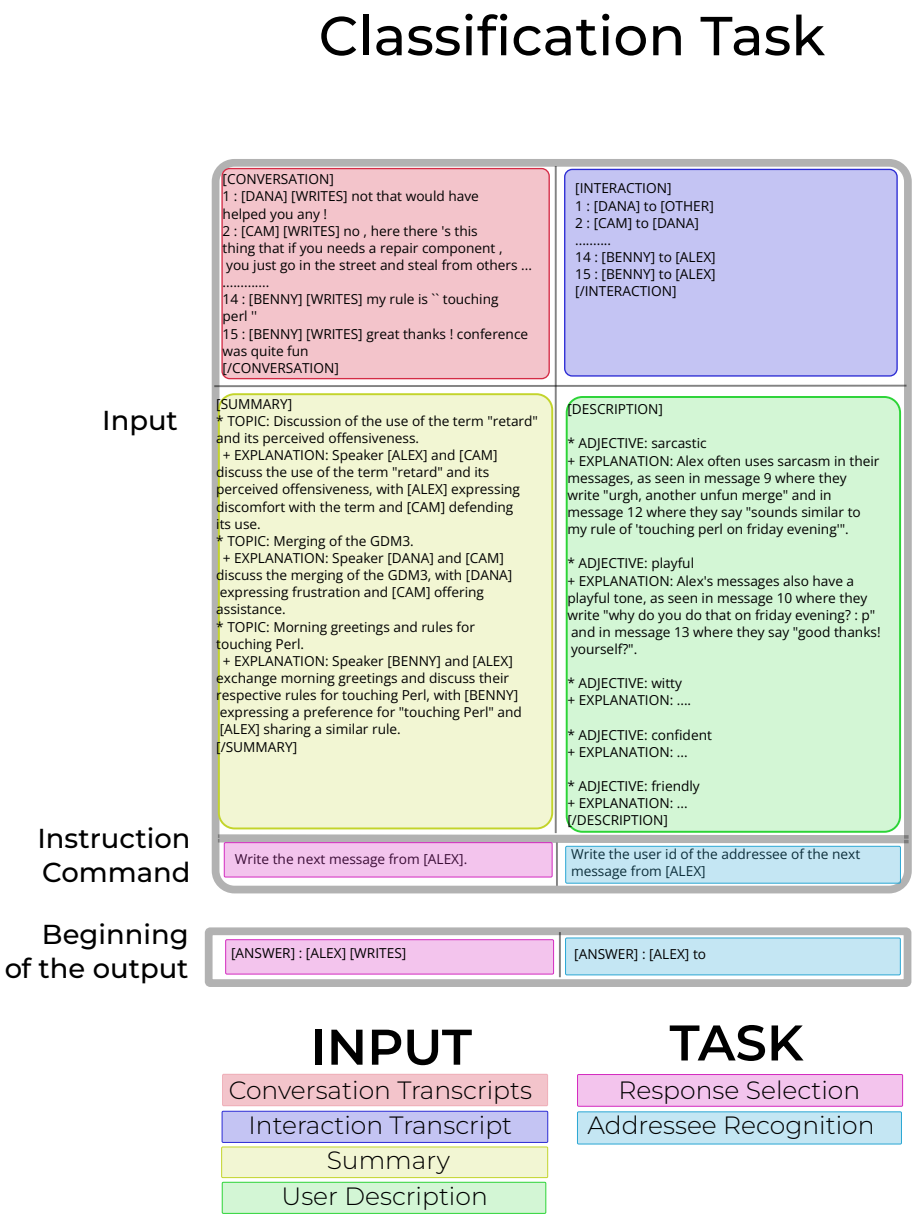

Figure 14: Input information and instruction command for classification. Here we also add a "beginning of input"section.

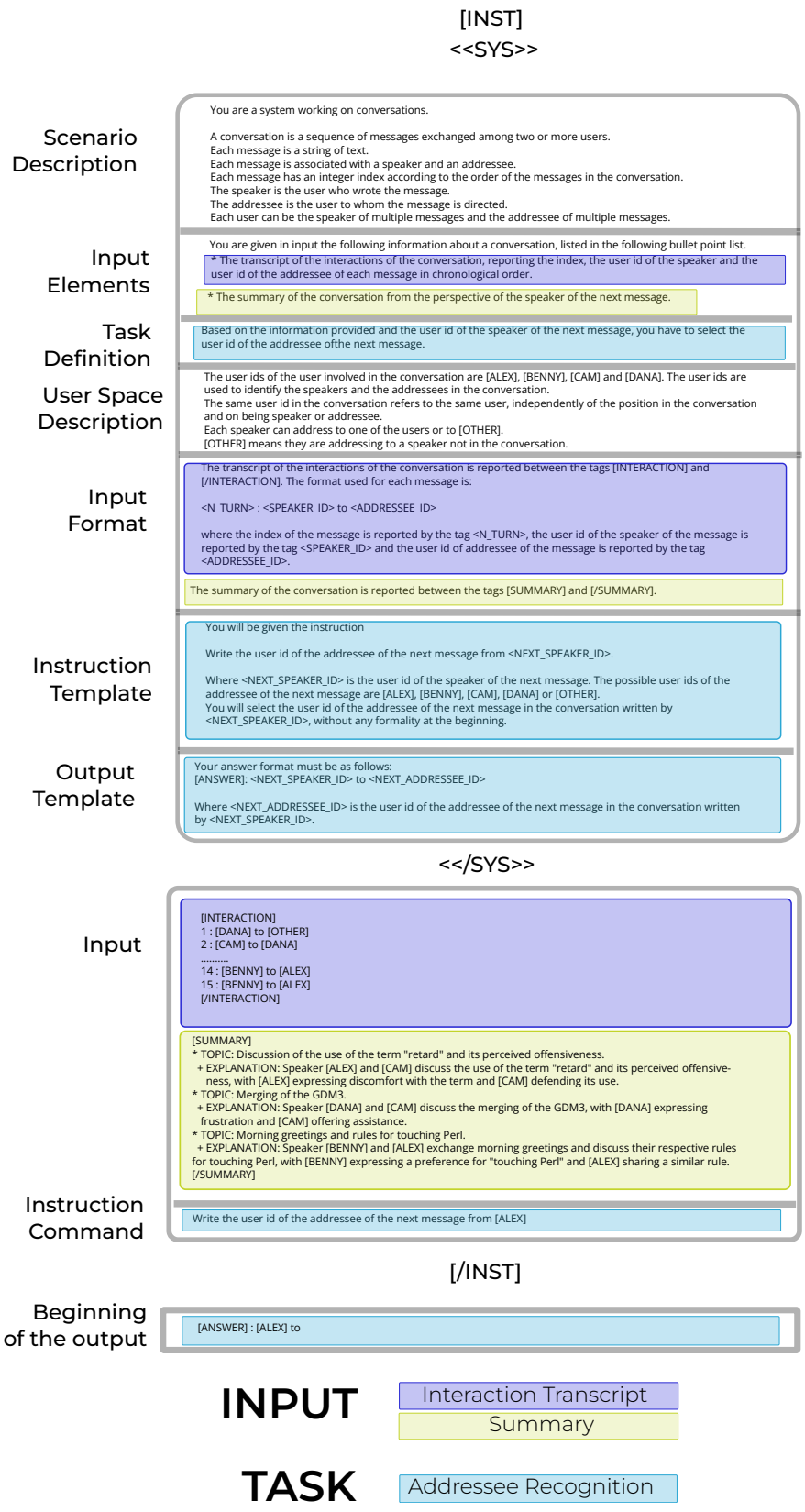

Figure 15: Example of final prompt for the STRUCT+SUMM combination in the AR task, following the first prompt scheme.

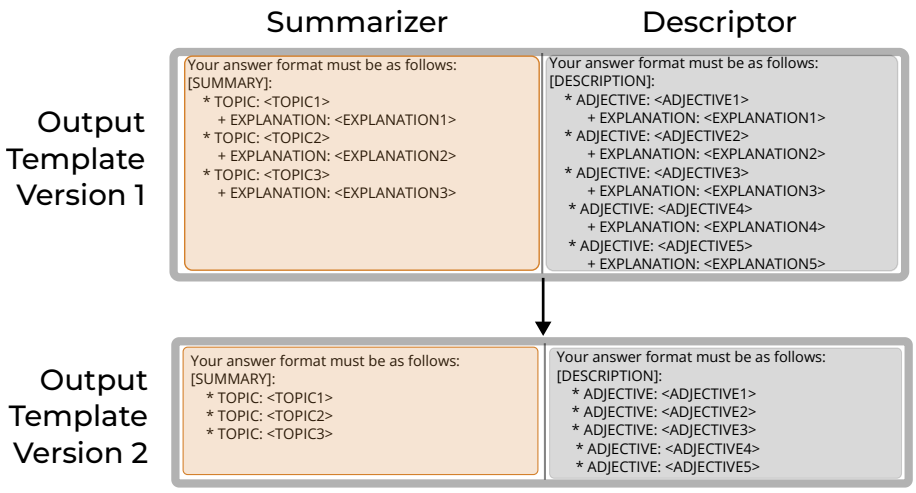

Figure 16: Versions of output template tested for generating summary/user description.