

LLMBERJACK: Guided Trimming of Debate Trees for Multi-Party Conversation Creation

Leonardo Bottona^{1,2}, Nicolò Penzo^{1,2}, Bruno Lepri², Marco Guerini², Sara Tonelli²

¹University of Trento ²Fondazione Bruno Kessler

Correspondence: leonardo.bottona@studenti.unitn.it, npenzo@fbk.eu

Abstract

Online debates on platforms like Reddit or Kialo are structured as *reply trees*: each message can receive several replies, which in turn can each receive several replies, producing a branching structure rather than a linear conversation. Turning one path (or a mix of paths) through such a tree into a natural, turn-by-turn conversation among multiple speakers (a process we call *linearization*) requires deciding which messages to keep, in what order, and who each message is addressed to. To this aim, we present LLMBERJACK, a platform for creating multi-party conversations starting from existing debates, originally structured as reply trees. The system offers an interactive interface that visualizes discussion trees and enables users to construct coherent linearized dialogue sequences while preserving participant identity and discourse relations. It integrates optional large language model (LLM) assistance to support automatic editing of the messages and speakers' descriptions. We provide preliminary evidence about the platform's utility by showing how tree visualization facilitates the creation of coherent, meaningful conversation threads and how LLM support enhances output quality while reducing human effort. The tool is open-source and designed to promote transparent and reproducible workflows to create multi-party conversations, addressing a lack of resources of this type.

1 Introduction

Despite ongoing efforts in the NLP community to create large datasets and linguistic resources, there is traditionally a lack of high-quality datasets with multi-party conversations, shortened as MPCs (Penzo et al., 2024b). Platforms such as X, Reddit and Kialo provide a large amount of conversations in the form of *reply trees*: a root message can receive several replies, each of those replies can receive further replies, and so on, producing a branching structure rather than a single back-and-forth

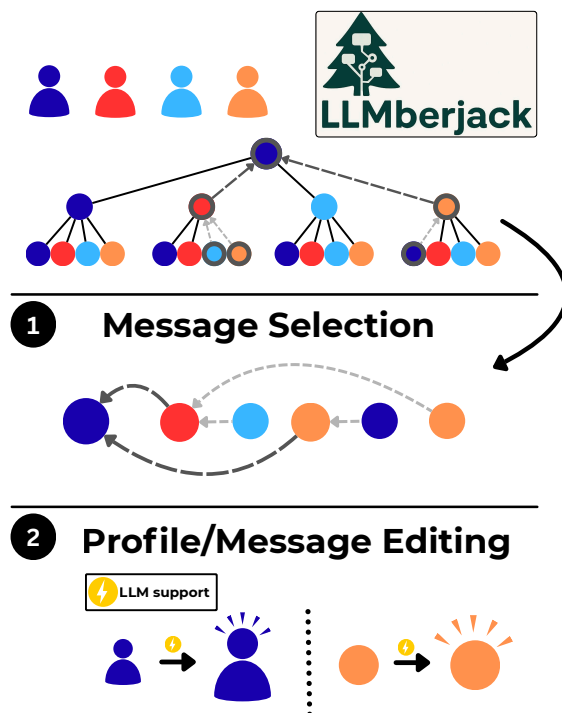


Figure 1: Overview of the LLMBERJACK platform. The interface integrates reply-tree visualization, message selection tools for building linearized multi-party conversations (1), and LLM-support for editing messages and speaker profiles (2).

exchange. For example, a root post “Should we introduce a four-day work week?” might receive two replies, one from a user arguing in favor and one arguing against, and each of these replies might in turn receive its own replies, so that the discussion develops along multiple, sometimes independent, branches. Each root-to-leaf path in such a tree can be read top to bottom as a multi-party conversation (Derczynski et al., 2017; Penzo et al., 2024a), since in this way it clearly shows who wrote each message and which earlier message it replies to; however, a single path rarely captures messages that address more than one participant at a time. We call the process of selecting and ordering mes-

sages from a reply tree into such a turn-by-turn conversation *linearization*.

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 MPCs, an annotation step would still be needed to make addressees (i.e., who a message is directed to) explicit and enable modelling their complex conversation structures. Furthermore, using discussions from online platforms to study MPCs raises significant privacy and profiling concerns (Kim et al., 2023).

LLMs could be potentially used to address the lack of MPC datasets by generating synthetic dialogues. However, as shown by Penzo et al. (2025), 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 parallel sub-dialogues starting from existing reply trees is to have human annotators traverse the tree following the reply-to links between messages, selecting at each step which message to include next and deciding who it is addressed to. Annotators can also edit message content or redefine addressee links along the way, 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.

In this paper, we introduce 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 traverse the tree step by step, 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 Kialo¹ or automatically generated may exhibit a style that is not fluent or natural enough. To enhance specific linguistic features or user traits, we implement an LLM-assisted protocol that supports two key tasks besides tree editing: (I.) *speaker profiling*, i.e., generating a short description of a speaker’s conversational style and

personality based on their messages (either from the emerging conversation or from the original reply tree), and merging it with any pre-existing description of that speaker; (II.) *message 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 evaluate the impact of both tree visualization and LLM-assisted message editing involving four annotators. Results suggest 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.

LLMBERJACK is available on a dedicated GitHub repository². The platform targets researchers from NLP and Social Sciences, helping them in the creation of high-quality MPCs with specific characteristics.

2 Related Work

Multi-party conversational corpora have been collected from a broad range of environments, including in-person meetings (Carletta et al., 2005; Janin et al., 2003) and online platforms (Ouchi and Tsuboi, 2016; Zhang et al., 2018; Chang and Danescu-Niculescu-Mizil, 2019). However, these diverse sources exhibit different characteristics that complicate cross-domain generalization. For instance, spoken multi-party dialogues are heavily shaped by non-verbal cues, physical setting, and overlapping turns, all of which are typically absent in written online interactions. Conversely, text-based conversations unfold asynchronously, without overlap, and often follow platform-specific conventions that further influence interaction patterns (Mahajan and Shaikh, 2021; Penzo et al., 2025). Heterogeneity in structure and annotation practices is shown also across datasets from similar sources.

The limited availability of 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 interactional plausibility. This has been tested in single-turn interactions (Fantoni

¹<https://www.kialo.com/>

²<https://github.com/LeonardoBottonaUniTn/llmberjack>

et al., 2021; Russo et al., 2023) and for multi-turn dialogues (Bonaldi et al., 2022; Occhipinti et al., 2024), but not yet for multi-party settings. Only Chen et al. (2023) and Penzo et al. (2025) have attempted to generate synthetic multi-party conversations, the former involving up to three users and the latter extending to interactions among four to six users.

Menini et al. (2025) introduced FIRSTAID, a platform for the synthetic creation of document-grounded dialogues among multiple participants, whose evaluation however has been limited to 1-to-1 interactions. Among the existing tools, CONVOKIT (Chang et al., 2020) is the most established one for multi-party settings, which focuses on the analysis of 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. LLMBERJACK is, to the best of our knowledge, the first open-source platform explicitly designed for this task, combining tree-based message selection with LLM-assisted speaker profiling and message refinement.

3 System Architecture

LLMBERJACK is designed to support the full process of 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.

3.1 Data Processing Backend

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.

3.2 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 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).

Labelling Support. The platform additionally provides a specific window for message or conversation-level labelling, supporting free-text labels and binary or Likert scale ratings, intended to support future evaluation workflows of this kind (details in Appendix A.4).

3.3 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 A.2). Upon request, the platform refines such profiles using an LLM. We use Llama 4 Maverick (Meta, 2025) for all the LLM-assisted tasks, exploiting the Groq cloud platform³. We select Llama 4 Maverick since it is one of the most widely-used instruction-following LLMs and could be easily run on the Groq cloud platform for free.⁴ However, our platform is model-agnostic by design, and users can configure their own API credentials and choose among multiple supported LLM providers and models (prompts, generation and implementation details in Appendix A.3).

To construct or refine a profile, the model receives: (I.) the original speaker profile to be re-

³<https://groq.com/>

⁴The experiments have been performed before the deprecation of this model from Groq (<https://console.groq.com/docs/deprecations>)

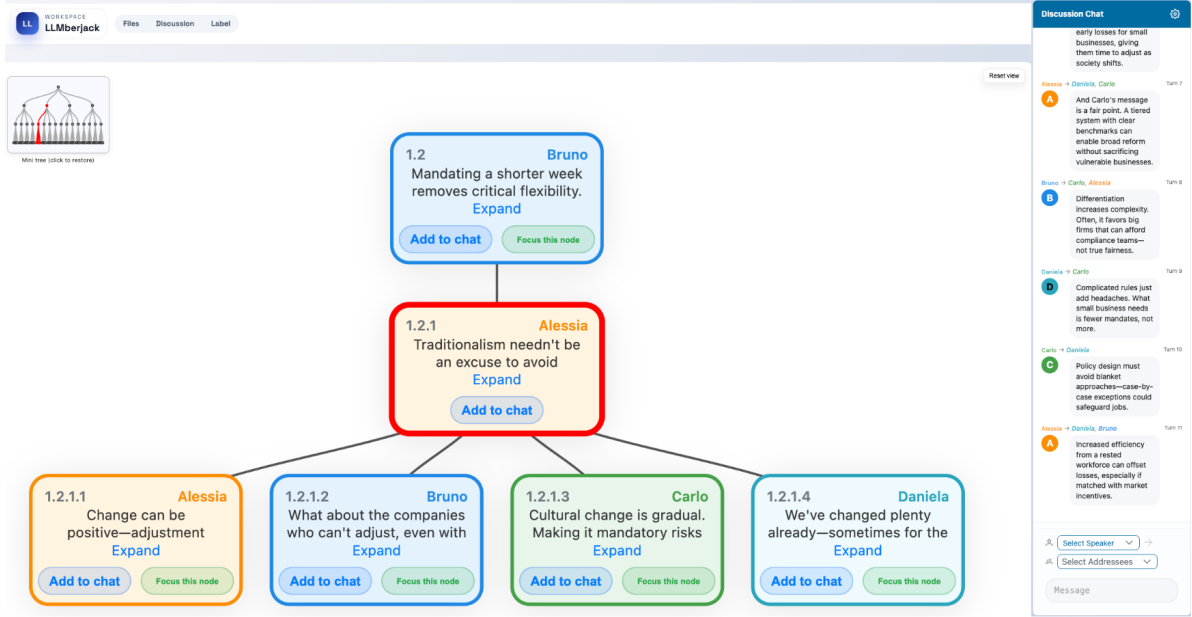


Figure 2: Screenshot of tree visualization for node 1.2.1 (left) and of the chat creation tab (right). Each node-box reports the speaker’s name on the top-right corner, and a preview of the message in the center (expandable).

Figure 3: Screenshot of the LLM-assisted message refinement page.

finer, and (II.) a set of selected messages from the speakers serving as contextual evidence (details in Appendix A.3). Based on this information, the LLM infers stylistic patterns and conversational temperament merging them into an updated profile.

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 3.

3.4 Data Export, Deployment and Availability

The system currently provides json export for the final conversations. The full platform is publicly available as open-source software via the dedicated GitHub repository mentioned in Section 1. 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.

4 Evaluation of LLMBERJACK Features

We evaluate through human assessment the impact of two core features of LLMBERJACK: the impact of tree visualization on the creation of MPCs and the LLM-assisted message editing. To isolate their effects, we split the analysis into two parts. Firstly, we perform the message selection task, starting from a reply tree, comparing conditions with and without tree visualization. Secondly, we edit a subset of messages, comparing scenarios with and without LLM support. The evaluation is conducted at a modest scale, under a counterbalanced design intended to reduce annotator- and topic-specific confounds (see Limitations).

4.1 Creation of Synthetic Reply-Trees

As a preliminary step for our evaluation, we first generate synthetic reply-trees.

Specifically, for each synthetic reply-tree we first ask the LLM to define a set of 4 users and then generate iteratively the full 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 (4 in our case), resulting in a total of 85 messages. LLMs are generally proficient at producing coherent one-to-one replies that respect speaker profiles or conversational roles. From these generated reply trees, we make the annotators build linearized multi-party conversations.

We generate 4 synthetic reply trees using GPT-4.1⁵. Each reply tree is about a different topic. For each topic, two speakers were assigned a pro stance and the other two speakers a counter stance with respect to the topic. We report the selected topics and further details in Appendix B.1.

The evaluation process consists of two main steps: (I.) selection of messages to build a MPC starting from the synthetic reply tree, with and without tree visualization (Section 4.2); (II.) editing of the resulting MPC messages, with and without LLM support (Section 4.3).

4.2 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.

R₁: The opening message must be a general statement on the given topic addressed to everyone.

R₂: Each conversation should contain between 10 and 15 messages and should resemble the style of a typical Telegram chat.

R₃: 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.

R₄: Annotators should perform only minimal, necessary edits 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 4.3).

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. We report further details of the annotation process in Appendix B.2.

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, speaker-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 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 evident for the Naturalness dimension. Furthermore, the average speed in-

⁵platform.openai.com/docs/models/gpt-4.1

	Nat.	Var.	Eng.	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	/
κ_w	0.44	0.40	0.25	/

Table 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).

creases 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 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.

Structural Metrics. Table 2 summarizes structural properties of the created MPCs, comparing the *w Tree* and *w/o Tree* conditions. The number of messages per MPC and the number of addressees per turn are comparable across the two conditions, indicating that tree visualization does not change how much a conversation grows or how broadly turns are addressed. In contrast, the interleaving score, which measures how often consecutive turns switch between different branches of the original

Metric	w Tree	w/o Tree
Avg Number of Turns	$12.17^{\pm 1.43}$	$12.42^{\pm 1.47}$
Avg Addressees per Turn	$1.53^{\pm 0.72}$	$1.57^{\pm 0.73}$
Avg Interleaving score	$0.22^{\pm 0.21}$	$0.37^{\pm 0.26}$

Table 2: Structural metrics of the MPCs created with and without tree visualization (mean $\pm$ standard deviation). Number of turns per MPC and addressees per turn describe conversation size and engagement; the interleaving score describes how often turn order switches between first-level branches of the original reply tree, with higher values indicating more frequent switching.

reply tree⁶, is higher *w/o Tree* (0.37 vs. 0.22), showing that annotators switch between first-reply branches more often when the tree structure is not visible, while *w Tree*, MPCs tend to stay within a single branch for longer before switching.

4.3 Evaluating the Impact of LLM Support

In the second evaluation step, we aim 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 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

⁶We measured the interleaving score as the fraction of consecutive turn-pairs, taken in chronological order, whose belong to different first-level branches.

	Gen.	Len.	Style	Temp.	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	/
κ_w	0.36	0.58	0.43	0.44	/

Table 3: 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).

whether the *w LLM* support or *w/o LLM* editing was of better quality, or whether the two versions were considered equivalent. Details about task and evaluation are reported in Appendix B.3.

Quantitative Evaluation. In Table 3, we report the evaluation results together with the average refinement velocity in terms of tokens⁷/second (v_{tokens}). Overall, the results show an 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).

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.

5 Conclusion

In this paper, 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 is 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. Within this setting, our evaluations provide preliminary evidence that the platform can accelerate the overall creation workflow, both in message selection and in refinement, while also leading to conversations of higher or equal quality.

Limitations

Our evaluation is conducted at a modest but tightly controlled scale: four synthetic reply trees, four annotators, and two evaluators, yielding 16 pairwise comparisons for the tree-visualization condition (Section 4.2) and 32 for the LLM-support condition (Section 4.3). The counterbalanced design reduces annotator- and topic-specific confounds, but a sample of this size is not intended to support strong statistical claims: we do not report significance tests or confidence intervals, and the percentages in Tables 1 and 3 should be read as descriptive summaries of this specific sample. We view these results as promising, preliminary evidence, with replication across more reply trees, topics, and annotators as a natural next step to establish which differences are most robust.

Moreover, our evaluation setup involves some specific choices whose generalizability could be further tested. The reply trees were synthetically generated with a single LLM (GPT-4.1) on topics salient to an Italian audience, and annotators and evaluators were Italian speakers based at the same institutions, which allowed for a controlled, well-understood setting but leaves open how results transfer to other languages, topics, or annotator pools. Similarly, the LLM-assisted platform feature for message refinement was evaluated using a single model (Llama 4 Maverick); the platform itself is model-agnostic by design, and testing additional providers and model sizes is a valuable direction for future work.

Finally, the current evaluation focuses specifically on isolating the effect of tree visualization and of LLM support within LLMBERJACK, rather than on benchmarking the platform as a whole. As such, it does not yet include reply trees from real online debates, which could introduce additional challenges such as uneven reply depth or off-topic

⁷Number of tokens of the final refined sentence

branches, nor a comparison against external baselines or related tools such as CONVOKIT (Chang et al., 2020) or FIRSTAID (Menini et al., 2025). Evaluating the platform on real debate data and against alternative workflows is a promising direction for follow-up work.

Ethics Statement

All annotators and evaluators involved in the data collection/evaluation were hired as PhD students or postdoctoral researchers at one of the institutions involved. The synthetic reply trees given to the annotators were carefully analyzed at the beginning to check the possible presence of offensive language or toxic content. No personal data were used to conduct this study and the speaker profiles were fully synthetic. Still, we are aware of the potential data leakage from LLM training data. This platform can help to paraphrase also real conversations for pseudo-anonymization purposes.

Acknowledgements

We thank Genoveffa Martone, Daniela Occhipinti, Daniel Russo and Sebastiano Vecellio Salto for their contribution to the evaluation activities. We thank Helena Bonaldi, Stefano Menini and Daniel Russo for the suggestions. The work of Bruno Lepri was partially supported by the NextGenerationEU Horizon Europe Programme, grant No. 101120237 – ELIAS and grant number 101120763 – TANGO. The work of Marco Guerini was partially supported by the European Union’s CERV fund under grant agreement No. 101143249 – HATEDEMICS. Nicolò Penzo’s activities have been part of the network of excellence of the European Laboratory for Learning and Intelligent Systems (ELLIS). Leonardo Bottona’s work was developed during his internship in Fondazione Bruno Kessler (Italy). The authors also thank AitomotiveLab for supporting Leonardo Bottona’s work.

References

- 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.
- 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.
- 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.
- 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.
- 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.
- 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.
- A. Janin, D. Baron, J. Edwards, D. Ellis, D. Gelbart, N. Morgan, and 1 others. 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.
- 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.
- 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.
- Stefano Menini, Daniel Russo, Alessio Palmero Aprosio, and Marco Guerini. 2025. [First-AID: 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.
- Daniela Occhipinti, Michele Marchi, Irene Mondella, Huiyuan Lai, Felice Dell’Orletta, Malvina Nissim, and Marco Guerini. 2024. [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.
- 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. 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. 2025. [Don’t stop the multi-party! on generating synthetic multi-party conversations with constraints](#). *Preprint*, arXiv:2502.13592.
- Daniel Russo, Shane Kaszefski-Yaschuk, Jacopo Staliano, 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.
- Justine Zhang, Jonathan Chang, Cristian Danescu-Niculescu-Mizil, Lucas Dixon, Yiqing Hua, Dario Taraborelli, and Nithum Thain. 2018. [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.

A Technical details

A.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.

A.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”.

A.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 (Llama 4 Maverick) and a fixed seed (42). Complete tem-

plates and parameter settings are available in the project repository (see the `llm_calls.py` module). The platform allows users to register API keys for multiple providers (currently Groq and OpenAI) and to switch among models; keys are stored locally and used only for the user’s own annotation session.

A.4 Labelling interface

In addition to tree visualization and LLM-assisted refinement, LLLMBERJACK provides a labelling window for evaluating linearized conversations, operating at either the message or the conversation level. Annotators can assign free-text string labels, binary (0/1) or Likert-scale ratings. To meet the needs of different labelling tasks, we give the possibility to freely configure the labelling templates. For the future, we plan to include pairwise comparisons between conversations, selecting a preferred candidate or indicating a tie. This is intended to support a complete pipeline of MPC synthetic generation through LLLMBERJACK. A screenshot of the labelling interface is reported in Figure 4.

B Evaluation details

B.1 Synthetic Reply Trees

The selected topics for the synthetic reply trees are:

- T₁:** Legalization of marijuana in Italy
- T₂:** Legalization of euthanasia in Italy
- T₃:** Introduction of a four-day work week
- T₄:** 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 MPCs derived from it, was used as tutorial material to familiarize annotators with the platform and the tasks, thereby minimizing platform-related issues during the actual annotation process.

B.2 Step 1 details

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

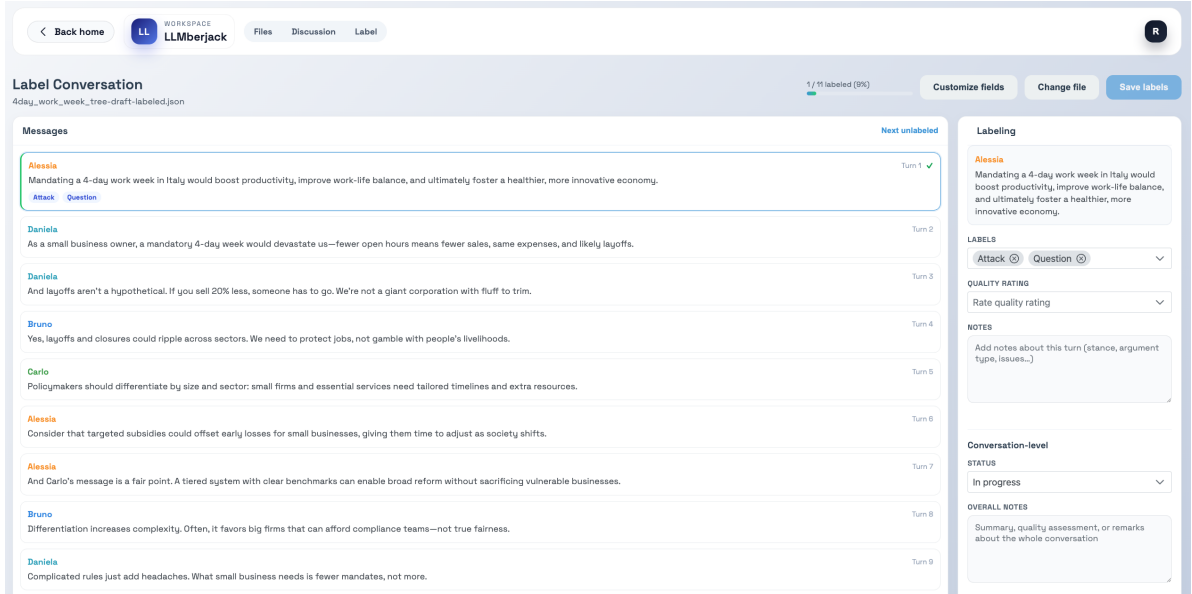


Figure 4: Screenshot of the labelling interface. The left panel shows the conversation, where users can select a message and assign it the labels from the supported label types. The right panel reports the labels assigned at both the message level (for the selected message) and the conversation level.

more challenging, while also minimizing annotator-specific variance in the quality assessment.

B.3 Step 2 guidelines

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, so the same speaker for both MPCs. Respectively we asked to make messages more: (I.) *aggressive, informal* and *much longer* for T₁; (II.) *exuberant, expressive* and *same length* for T₂; (III.) *cynical, concise* and *slightly shorter* for T₃; (IV.) *detached, formal* and *slightly longer* for T₄. The combination *sarcastic, neutral* and *much shorter* was used as “tutorial” setting.