Disentangling Approaches to Conversation Disentanglement: Fine-Tune or Learn from Scratch?

Debaditya Pal, Anton Leuski, Ron Artstein, David Traum, Kallirroi Georgila
Institute for Creative Technologies, University of Southern California
12015 Waterfront Drive, Los Angeles, CA 90094-2536, USA
debaditya.pal6@gmail.com, {leuski, artstein, traum, kgeorgila}@ict.usc.edu

Abstract

Conversation disentanglement is the process of segmenting a stream of messages or utterances into separate conversations or "threads" that can be more easily understood and processed. We compare the performance of GPT-4o and GPT-4o Mini with deep learning models built from scratch for this task. We show that, using the same amount of training data, out-of-the-box GPT-4o performs poorly, and fine-tuning GPT-4o Mini results in performance comparable to learning small-size models from scratch (based on standard hand-crafted features for this task), with performance reaching 74.4% F1-score for prediction of links between messages and 45.3% F1-score for prediction of perfectly matching conversations. However, the fine-tuned GPT-4o Mini model underperforms when compared to models that utilize complex structural information. We also provide a new method for detailed analysis of the successes and failures of our models, and a new visualization method.

Keywords: conversation disentanglement, multi-party dialogue, fine-tuned LLMs, small-size deep learning models, comparison of LLMs and small-size models, evaluation, visualization

1. Introduction

Compared with single speaker/author discourse or even dyadic conversation, multi-party chat can be quite complex to analyze and understand. Issues such as who is talking to whom and which parts of context are referred to by responses and referring expressions must take into account interleaved conversations, in which target and antecedent utterances are sometimes separated by material from a different conversation. Conversation disentanglement (Elsner and Charniak, 2008) is the process of segmenting a stream of messages or utterances into separate conversations or "threads" that can be more easily understood and processed.

A number of approaches to this problem have been proposed (see section 2), and, as with many NLP tasks, results often depend on the complexity of the data and amount of related training data available.

In this paper, we analyze several approaches with different computational costs and data requirements. Specifically, we compare using an out-of-the-box large language model (LLM) GPT-4o with GPT-4o Mini (a scaled down version of GPT-4o) fine-tuned using conversations from the Internet Relay Chat (IRC) Disentanglement Corpus manually annotated with reply-to relations between messages (Kummerfeld et al., 2019). We also build small-size deep learning models from scratch (using standard hand-crafted features for this task) based on two methods: an attention-based neural network and a feedforward neural network, both of which use BERT embeddings.

We show that, using the same amount of training data, out-of-the-box GPT-4o performs poorly, and fine-tuning GPT-4o Mini results in performance comparable to learning small-size models from scratch (based on standard hand-crafted features for this task). We also provide a detailed analysis of the successes and failures of our models.

The main contributions of our work are:

1. To our knowledge, this is the first experiment in the literature comparing fine-tuned LLMs (specifically GPT-4o Mini) with small-size models learned from scratch for conversation disentanglement. Recently, Li et al. (2024) performed preliminary experiments using GPT-3.5-turbo-03-01 and GPT-4-0125-preview without fine-tuning, both of which resulted in poor performance.
2. We provide a new method for detailed analysis and comparison of the fine-tuned GPT-4o Mini model, the attention-based model, and the feedforward model.
3. We describe a new method for visualizing the outputs of all models with respect to the gold-standard annotations.

The rest of the paper is structured as follows: In section 2 we present related work on conversation disentanglement. Section 3 presents the task and the dataset that we use for our experiments. In section 4 we describe our experiments and provide results. Section 5 provides a detailed analysis of the outputs of our models and a visualization method. Section 6 concludes and suggests directions for future work. Finally, in section 7 we discuss limitations of our work.

Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026), pages 2927–2941
11–16 May 2026.

2. Related Work

Conversation disentanglement addresses the challenge of separating interwoven dialogue threads in multi-party conversations. Early approaches predominantly relied on rule-based heuristics and clustering algorithms, given the limited availability of large annotated datasets. For instance, Elsner and Charniak (2008) introduced one of the earliest annotated datasets derived from Internet Relay Chat (IRC) and applied a graph-theoretic clustering method using hand-crafted features such as timing, participant identification, and lexical patterns. These initial techniques validated the feasibility of automated disentanglement, although their performance was inherently limited due to reliance on simplistic features and assumptions.

Early machine learning solutions, such as those described by Mehri and Carenini (2017), employed traditional classifiers (e.g., random forests) to determine reply links, supplemented by neural sequence models (recurrent neural networks) for thread formation, thereby surpassing heuristic methods.

Significant progress followed the creation of larger and more systematically annotated datasets. Kummerfeld et al. (2019) developed the now publicly available IRC Disentanglement Corpus, comprising 77,563 messages with explicitly annotated reply-to relationships. This corpus, substantially larger than previous datasets combined, solidified the shift towards data-driven, supervised learning methods. Their best model on various metrics (the same metrics that we use for our experiments) was based on combining GloVe embeddings with hand-crafted features.

With the rise of transformer-based architectures, pretrained language models, particularly BERT (Devlin et al., 2019), have substantially improved performance on disentanglement tasks. Thus, Li et al. (2021) proposed Dialogue BERT (DialBERT), a model integrating local and global semantics. This model was based on a combination of BERT (capturing the matching information in each sentence pair) and a bidirectional long short-term memory (BiLSTM) network (incorporating context information). It outperformed the models developed by Kummerfeld et al. (2019) for certain metrics.

Jiang et al. (2018) were the first to use convolutional neural networks (CNNs) for conversation disentanglement on a dataset they created from Reddit posts (see below), as well as on the dataset used by Elsner and Charniak (2008). In particular, they used a Siamese hierarchical CNN to capture both local and global semantics, and thus estimate the conversation-level similarity between closely posted messages. Tan et al. (2019) used an utterance-level LSTM network on a modified version of the Reddit dataset created by Jiang et al. (2018). Zhu et al. (2020) proposed a masked hierarchical transformer based on BERT but they evaluated their model only on the development set of the Ubuntu IRC corpus. Note that Ubuntu IRC is a subset of the IRC Disentanglement Corpus (Kummerfeld et al., 2019).

Yu and Joty (2020) developed a method based on Pointer Networks that does not rely only on hand-crafted features. Pointer Networks use additional information such as the fact that a speaker could mention another user's name in a message, which helps the model perform better. Again, using the Ubuntu IRC corpus, Zhu et al. (2021) found that a feedforward model with BERT embeddings and hand-crafted features achieves high performance. This model is very similar to one of the models that we have developed for our experiments (with small differences in implementation).

Ma et al. (2022) designed a model for conversation disentanglement taking into account structural information. Specifically, they used BERT and modelled two structural features, namely, speaker property (i.e., user identities of messages), and reference dependency (i.e., mention of users in messages). The assumptions are that some speakers may interact only with a limited number of other speakers, and that mentioning a user's name in a message is useful information that can help with disentanglement. Note that, as mentioned above, Pointer Networks (Yu and Joty, 2020) also encode mentions of user names in messages.

Lam and Yang (2025) investigated the effectiveness of BERT, XLNet, ELECTRA, RoBERTa, DeBERTa, and ModernBERT in conversation disentanglement. Similarly to Zhu et al. (2021), they used a feedforward neural network and hand-crafted features. DeBERTa outperformed all other models.

All of the above approaches are designed to optimize local decisions only (reply-to link predictions), except for the models of Li et al. (2021) and Jiang et al. (2018) which capture both local and global semantics. Bhukar et al. (2023) employed Reinforcement Learning (RL) on top of the "Structural BERT" approach mentioned above (Ma et al., 2022). They used a thread-level reward function that directly optimizes global metrics (such as variation of information and adjusted rank index) without ignoring local decisions. Their model resulted in state of the art performance on the Ubuntu IRC dataset.

Several researchers have created synthetic datasets for testing disentanglement. These include Jiang et al. (2018) who created a dataset from Reddit posts, and Liu et al. (2020) who created a dataset by intermingling different "sessions" from movie scripts. While these datasets cover different topics and communication media than IRC, they unfortunately do not really address the real disentanglement problem. They both make assumptions

that all utterances from one source (comments under a Reddit post or a movie session) are part of the same thread, while utterances from different sources represent different threads. While this may often be the case, there are many examples where more than one thread (when viewed in terms of topic, content, and relation to previous messages) are present in a session or Reddit comments. Moreover, the utterances themselves are taken out of their original context. The speakers/composers may have phrased them differently if in a merged context where it may be less clear which previous utterance they are referring to. Likewise, addressees and other participants may have tried to clarify if they saw confusing messages in this content.

Below, we present other recent work related to the task of conversation disentanglement. Kawano et al. (2023) built a neural dialogue structure parser with an attention mechanism and multi-task learning to automatically identify the dialogue structure of multi-floor dialogues in a collaborative robot navigation domain. Dialogue structure parsing included predicting the correct antecedent of an utterance as well as their relation type. Kawano et al. (2023) used the publicly available SCOUT (Situated Corpus Of Understanding Transactions) corpus of human-robot dialogues in this domain (Lukin et al., 2024). This corpus includes dialogue structure annotations such as transactional units and information about how utterances are related to each other (Traum et al., 2018).

Georgila et al. (2024, 2025) presented annotation schemes for team dialogue processing in military domains. These annotations include transactional units as well as how information is passed up or down the chain of command. In addition, there are explicit links between utterances marking the initiation and resolution points for events (commands, suggestions, and requests).

Recently, LLMs such as GPT-3.5 and GPT-4 have attracted attention in the task of conversational disentanglement. Li et al. (2024) conducted initial experiments (only using 100 utterances for testing) employing GPT variants without fine-tuning (specifically GPT-3.5-turbo-03-01 and GPT-4-0125-preview), revealing poor performance. This outcome highlighted that, despite substantial language understanding capabilities, general-purpose LLMs require task-specific fine-tuning to handle the complexities inherent in disentangling conversations.

To our knowledge, no prior work on conversation disentanglement has systematically compared the effectiveness of fine-tuned LLMs with bespoke neural models trained from scratch under comparable conditions. Addressing this gap, our study compares fine-tuned and non-fine-tuned variants of GPT models (GPT-4o Mini and GPT-4o respectively) against neural architectures explicitly designed and trained for conversation disentanglement, investigating the trade-offs between leveraging pretrained knowledge and learning task-specific knowledge anew.

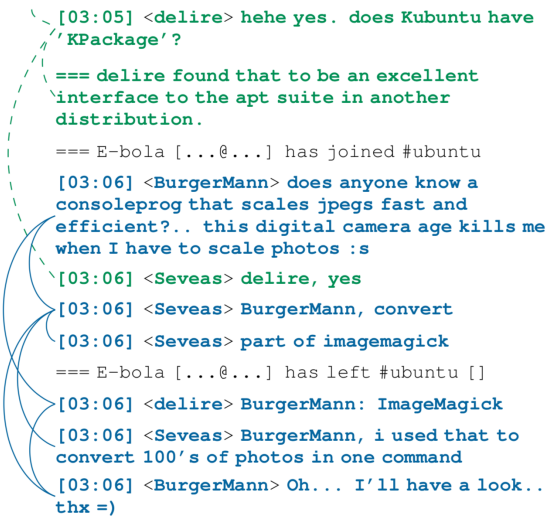

Figure 1: Example annotation in the IRC Disentanglement Corpus taken from Kummerfeld et al. (2019). Curved lines show the reply structure. Blue and green colors denote different conversation threads.

3. Task and Dataset

We use the IRC Disentanglement Corpus (Kummerfeld et al., 2019), a large-scale dataset consisting of multi-participant chat conversations annotated with reply links, for training our neural models (attention and feedforward), fine-tuning GPT-4o Mini, and testing all our models. The task is framed as antecedent selection: given an utterance and a list of prior utterances in the same channel, the model must identify the utterance it is replying to (or mark it as starting a new thread).

Figure 1 shows an example annotation in the IRC Disentanglement Corpus taken from Kummerfeld et al. (2019). Here we can see two entangled conversation threads (in blue and green respectively) and their graph structure. A message can respond to multiple messages and receive multiple responses. Also, users may participate in more than one conversation thread. Some messages are "system" messages indicating actions (e.g., entering or leaving the channel) such as the third message in the example. Of the "user" messages some are directed to a specific named user such as the fifth message in the example.

The IRC Disentanglement Corpus contains 77,563 messages: 74,963 messages from the Ubuntu IRC channel and 2,600 messages from the Linux IRC channel. For our experiments we

used the messages from the Ubuntu IRC channel (67,463 in the training set, 2,500 in the development set, and 5,000 in the test set).

4. Experiments and Results

We perform experiments with new models and compare to previous results from the literature on the Ubuntu IRC dataset, which is a subset of the IRC Disentanglement Corpus described in the previous section.

4.1. Evaluation Metrics

We use the official evaluation script provided by (Kummerfeld et al., 2019) to compute several metrics measuring performance in determining how messages are connected (links-related metrics), and in clustering messages into conversations (conversation-related metrics). This script ensures a standardized evaluation procedure, consistent with prior work on this dataset.

4.1.1. Links-related Metrics

The script evaluates system output against gold-standard reply links by calculating precision, recall, and F1-score over directed utterance pairs. Given a set of system automatically predicted links and a set of gold links, the script matches utterance pairs across files. Each pair is represented as a directed link from a source utterance to its antecedent (or to itself, in case of thread initialization).

Below we define precision, recall, and F1-score. Note that G is the total number of gold links, A is the number of automatically predicted links, and M is the number of correctly matched links.

- **Precision:** The percentage of predicted links that are present in the gold standard, computed as Precision $= \frac{M}{A}$.
- **Recall:** The percentage of gold-standard links that are correctly predicted, computed as Recall $= \frac{M}{G}$.
- **F1-score:** The harmonic mean of precision and recall, computed as F1 $= \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$.

So for example, suppose the gold file contains the links `(5,2)`, `(6,3)`, `(7,2)`, and `(7,5)`, while the system output contains `(5,2)`, `(6,4)`, and `(7,5)`. The matched links are `(5,2)` and `(7,5)`, so $G = 4$, $A = 3$, and $M = 2$. Thus, the resulting scores are: Precision = 66.7%, Recall = 50%, and F1-score = 57.2%.

4.1.2. Conversation-related Metrics

The script also calculates metrics on conversation level rather than just link level. Based on the gold links, it generates clusters of messages ("gold clusters"), where each cluster includes messages belonging to the same conversation. Likewise, based on the automatically predicted links, the script generates "automatically predicted clusters" of messages. We use the following metrics:

- **Modified Variation of Information (VI) (Meilă, 2007):** This measures the information gained or lost when going from one clustering to another. It is the sum of conditional entropies $H(Y|X) + H(X|Y)$, where X and Y are clusterings of the same set of items. We used the bound for n items that $VI(X;Y) \leq log(n)$, and present $1 - VI$ so that larger values are better.
- **One-to-One Overlap (1-to-1) (Elsner and Charniak, 2008):** This measure is computed by pairing up conversations (clusters) from two annotations to maximize total overlap, and then reporting the percentage of overlap found.
- **Exact Match Precision, Recall, and F1-score:** This is computed using the number of perfectly matching conversations (clusters), excluding conversations with only one message (mostly system messages). Two conversations are considered as matching when they include the same messages without necessarily having the same link structure. This is a very challenging but easy to understand metric which directly measures performance in terms of perfectly extracted conversations.

4.2. Model Architectures

We experiment with GPT-4o, GPT-4o Mini, and two neural models that utilize BERT-based utterance representations along with a set of auxiliary hand-crafted features derived from the earlier implementation by Kummerfeld et al. (2019). These features capture structural and temporal characteristics of IRC conversations, providing complementary information to the semantic representations from BERT. For example, such features encode whether a message is a system message or a normal message, whether the message is targeted or not, whether there is a previous message from the user of the current message or not, how long ago in minutes was the previous message from the user of the current message, etc. For details, see Kummerfeld et al. (2019). Henceforth, these features are referred to as "standard hand-crafted features".

4.2.1. GPT-4o and GPT-4o Mini

We conducted two different experiments with GPT. In the first experiment, we crafted a prompt for the system that contained the description of the problem, the description of the edge cases (see the documentation in the GitHub repository[1]), and a short snippet from an IRC log file (20 messages) together with the corresponding set of links. We used this system prompt together with each individual test file as an input to the GPT-4o model. The results were significantly below previous state of the art, with roughly 3/4 of the messages misclassified (see Table 1).

For our second GPT experiment, we fine-tuned the model. Fine-tuning a GPT model requires providing it with a set of dialogue examples. Each example consists of a system prompt, a sequence of user and assistant messages for context, and the desired assistant output. We created a dialogue example from each of the training IRC chat logs. We used the same prompt as in the first experiment (minus the 20 messages) as the system prompt. The full text of a chat log was the user message, and the list of links was the desired output. In this way we had 153 examples (chat logs) for tuning (the full training set of Ubuntu IRC). At the time of our experiments GPT-4o was not available for fine-tuning, so we fine-tuned the GPT-4o Mini model. We evaluated the fine-tuned model on the test set of Ubuntu IRC. The results of the evaluation are shown in Table 1. The fine-tuned model performs noticeably better.

Note that in both experiments, GPT annotates messages "in batches" – it takes a log file of several thousand messages and produces a list of links between messages in one call. An alternative setup would be to feed some portion of a log to GPT and ask it to link the last message to some of the previous ones. We believe this might be an easier task for the model to handle, however, it requires running the model separately for each individual chat message, which we deemed to be too expensive.

4.2.2. Neural Models Learned from Scratch

We implemented and tested two neural models that use BERT embeddings along with the auxiliary standard hand-crafted features (same as the ones used by Kummerfeld et al. (2019), see above). These BERT embeddings represent messages and antecedents as vectors that can be used as input to the neural networks. The neural models include a novel attention-based model (BERT+ATT) as well as a refinement of the models from Kummerfeld et al. (2019) to use BERT rather than GloVe (BERT+FF).

[1] https://github.com/jkkummerfeld/irc-disentanglement

Attention-based Model (BERT+ATT). This model applies multiple layers of multi-head self-attention and position-wise feedforward networks to contextualize each candidate antecedent with respect to the query (current message of interest). The input to the model consists of the BERT embedding of the query, concatenated with the standard hand-crafted features. Each attention layer computes attention from candidate utterances to the query, and the resulting representations are passed through residual connections and layer normalization. The final link prediction is obtained via Softmax over similarity scores computed from the contextualized representations.

Feedforward Model (BERT+FF). Here, we concatenate the BERT embeddings of the query and candidate antecedent, along with standard hand-crafted features, and feed the result into a deep feedforward network comprising three fully connected layers with ReLU and Softsign activations. The final scalar output is treated as a relevance score, and the most relevant antecedent is selected via Softmax.

Each of the two small-size neural models is trained using a cross-entropy-style loss over candidate antecedents. We run five independent training instances per model using the random seeds {0, 10, 42, 86, 523}. Optimization is performed using the Adam optimizer. Note that in both neural models, sentence embeddings were calculated by averaging over BERT word embeddings from the final layer of BERT. Results for Single Models shown in Table 1 are based on the best instance of BERT+ATT and the best instance of BERT+FF, both of which used seed 42.

4.2.3. Ensembling Strategies

To improve robustness and reduce variance, we ensemble model outputs using the following strategies:

- **Union:** A link is predicted if any of the ensemble members predict it.
- **Vote:** Majority voting is used over the predicted links from individual models.

We evaluate six ensemble configurations: four that ensemble five models of a single architecture (either BERT+ATT or BERT+FF), and two that combine all ten models (five of each type). For each setting, we compare the Union and Vote strategies described above.

Model	Links			Conversations				
	P	R	F1	VI	1-to-1	P	R	F1
Single Models								
GPT-4o	26.2	18.9	22	59.4	20.9	0	0	0
GPT-4o Mini fine-tuned	**75.3**	**73.5**	**74.4**	**92.5**	**78**	**42.6**	**48.5**	**45.3**
GloVe+FF (Kummerfeld et al.)	73.7	71	72.3	91.3	75.6	34.6	38	36.2
Glove+FF+self-links (Yu and Joty)	74.2	71.5	72.8	92	70.4	41.9	40.1	41
BERT+FF prev (Zhu et al.)	73.9	71.3	72.6	92	77	-	-	40.9
DeBERTa+FF (Lam and Yang)	73.6	71	72.3	91.8	-	36.6	41.8	39
Structural BERT (Ma et al.)	-	-	-	**94.6**	**84.2**	**51.8**	**51.7**	**51.7**
Structural BERT (Bhukar et al.)	**75.3**	**75.5**	**75.4**	93	-	46.7	47.6	47.1
Structural BERT + RL (Bhukar et al.)	**83.3**	**83.3**	**83.3**	**96.2**	-	51.5	**52.3**	**51.9**
Pointer Networks (Yu and Joty)	74.5	71.7	73.1	94.2	80.1	44.9	44.2	44.5
DialBERT (Li et al.)	-	-	-	93.2	79.7	42.1	47.9	44.8
BERT+FF	73.8	71.1	72.4	90.2	72.5	30.5	33.2	31.8
BERT+ATT	69.6	67.1	68.3	89.6	71.5	26.8	33	29.6
Ensembles								
GloVe+FF U x10 (Kummerfeld et al.)	64.3	79.7	71.2	86.2	62.5	**40.4**	28.5	33.4
GloVe+FF V x10 (Kummerfeld et al.)	**74.9**	72.2	**73.5**	**91.5**	**76**	36.3	**39.7**	**38**
BERT+FF U x5	63.6	78.9	70.4	86.7	60.8	39	27.9	32.5
BERT+FF V x5	74.3	71.6	73	91	74.6	32.4	36.9	34.5
BERT+ATT U x5	60.7	74.1	66.8	86.2	61.9	33	23.9	27.7
BERT+ATT V x5	70.2	67.7	68.9	89.9	71.8	27.3	31.8	29.4
BERT+ATT & BERT+FF U x10	56.9	**80.9**	66.8	84.1	55.8	38.4	22.3	28.2
BERT+ATT & BERT+FF V x10	73.2	70.5	71.8	90.5	73.2	29.4	34.7	31.8

Table 1: Results for single models and ensemble configurations. The best scores for each metric are highlighted in bold black. For single models that use only local information, the best scores are shown in bold blue. For single models, bold violet indicates when fine-tuned GPT-4o Mini outperforms models that use only local information and standard hand-crafted features for this task. For ensemble models, the best scores are shown in bold red. U: Union, V: Vote. For Structural BERT, the results generated by Bhukar et al. (2023) are based on code released by Ma et al. (2022).

4.3. Results

Evaluation results for all model configurations are presented in Table 1. We also include results from Kummerfeld et al. (2019) (GloVe+FF), Zhu et al. (2021) (BERT+FF prev), Yu and Joty (2020) (GloVe+FF+self-links and Pointer Networks), DialBERT (Li et al., 2021), DeBERTa+FF (Lam and Yang, 2025), and Structural BERT with and without RL (Ma et al., 2022; Bhukar et al., 2023).

We can see that of the single models, the fine-tuned GPT-4o Mini outperforms for all metrics all models based on the standard hand-crafted features for this task (local information). Differences in performance among GloVe+FF, BERT+FF prev, and BERT+FF are negligible in terms of links-related metrics but there are variations in terms of conversation-related metrics. Differences between BERT+FF prev (Zhu et al., 2021) and BERT+FF (our model) are most likely due to variations in implementation; for example, to calculate utterance embeddings we use averages over BERT word embeddings whereas Zhu et al. (2021) use the BERT [CLS] token. Note also that precision, recall, and F1-score (conversations) are very sensitive metrics and minor differences in clusters may result in large differences in these measures. For example, two conversations that differ in only one message will be considered as not matching (see 4.1.2). GloVe+FF+self-links outperforms GloVe+FF for almost all metrics. This is because GloVe+FF+self-links uses information from self-links (predicting when an utterance points to itself), which plays an important role especially in clustering performance.

Among the models that use additional information, Structural BERT combined with RL performs the best (Bhukar et al., 2023), which is not surprising given that it is optimized on both global and local metrics. Structural BERT without RL also performs well in both of its versions (Ma et al., 2022; Bhukar et al., 2023). Note that Pointer Networks (Yu and Joty, 2020), which rely on some of the information used by Structural BERT but not on hand-crafted features, overall perform worse than fine-tuned GPT-4o Mini. Pointer Networks outperform fine-tuned GPT-4o Mini only in terms of VI, 1-to-1, and precision (conversation-related).

Of the ensembles, GloVe+FF (x10 Vote) performs the best in terms of precision and F1-score (links-related), and VI and 1-to-1, but similarly to BERT+FF (x5 Vote). However, GloVe+FF (x10 Vote) outperforms BERT+FF (x5 Vote) with regard to precision, recall, and F1-score (conversation-related); see discussion above on the sensitivity of conversation-related precision, recall, and F1-score metrics. Note that "x5 Vote" means that there were five independent training instances per model (using five different seeds) and "x10 Vote" means that there were ten independent training instances per model (using ten different seeds). Likewise for the Union models. The Combined BERT+ATT and BERT+FF (x10 Union) model performs the best with regard to recall at the expense of precision (links-related).

If we compare both the single models (with the standard hand-crafted features for this task) and the ensembles, for the links-related metrics, fine-tuned GPT-4o Mini achieves the highest precision and F1-score, and Combined (x10 Union) achieves the highest recall.

For practical purposes the performance of fine-tuned GPT-4o Mini is comparable to the performance of the feedforward models (with GloVe, BERT, or DeBERTa embeddings), Pointer Networks, and DialBERT. However, given that it is much easier to fine-tune GPT than build a model from scratch, this shows how powerful LLMs have become. On the other hand, for applications that require privacy or need to run on small devices with limited memory, building a small-size model from scratch seems like a good alternative. In addition, the models that outperform fine-tuned GPT-4o Mini (Structural BERT and Structural BERT combined with RL) require more sophisticated training procedures.

Another consideration is that most of the small-size models (excluding Pointer Networks) require hand-crafted features (in addition to the embeddings) to perform well. Zhu et al. (2021) showed that performance drops significantly when only BERT embeddings are used without any hand-crafted features. Interestingly, Georgila (2024) also found that combining hand-crafted features with embeddings outperforms only using embeddings for the task of predicting user ratings after their interaction with dialogue systems, although in that case relying only on embeddings did not result in significant drop in performance.

5. Analysis

5.1. Categorization of Links

The task of disentanglement is to split a list of messages into conversations, where a conversation is defined as a maximal set of messages connected through antecedent links. In practice, systems only annotate the antecedent links, and conversations are inferred from these links. As discussed above, Kummerfeld et al. (2019) use separate measures for evaluating links and conversations: the link measures do not look at the overall conversations, while the conversation measures do not consider the internal link structure within each conversation. The analysis below is an attempt at understanding a system's performance using both links and conversations: it primarily looks at the individual antecedent links, since these are what the systems produce, but also at how the conversations match for each annotated link. Each antecedent link is categorized as one of the following:

Full Match: The system identifies the same link as the gold annotation, and the two conversations are identical (exactly the same messages, although links within each conversation may differ). This category also includes messages which both system and gold identify as singular, that is, not linked to any other message.

Same Link: The system identifies the same link as the gold annotation, and the two conversations are not identical (although obviously overlapping).

Same Conversation: The system identifies a different link (or links) than the gold annotation, and the two conversations are identical (exactly the same messages, although obviously some different links).

Conversation Overlap: The system identifies a different link (or links) than the gold annotation, the two conversations are not identical, and they overlap with more than one message in common (that is, more than just the current message).

No Match: The system identifies a different link (or links) than the gold annotation, and there is no overlap in the identified conversations.

For example, let us assume that the gold annotation includes links $2 \rightarrow 1$ and $3 \rightarrow 2$ and the system annotation includes links $2 \rightarrow 1$ and $3 \rightarrow 1$. For utterance 2, the links are the same for gold and system, and the conversation (set of messages) is identical: $\{1, 2, 3\}$, so message 2 counts as Full Match. But there are links in the conversation that are different between the gold and the system, and message 3 counts as Same Conversation.

The "No Match" category is further categorized as follows:

Gold Single: The gold annotation identifies this as a singular message (not linked to anything

Test Set	Full Match	Same Link	Same Conv	Conv Overlap	No Match				Total
					G-single	S-single	None	N/A	
GPT 10 Test Logs	1482	2140	202	880	26	240	27	3	5000
GPT 9 Test Logs	1323	2057	176	833	26	56	26	3	4500
ATT Union	804	2131	61	1859	28	81	36	0	5000
FF Union	901	2152	105	1729	24	70	19	0	5000
Combined Union	778	1956	80	2069	25	70	22	0	5000
ATT Vote	1027	2330	121	1325	28	91	78	0	5000
FF Vote	1117	2443	111	1173	24	84	48	0	5000
Combined Vote	1121	2378	115	1205	25	92	64	0	5000

Table 2: Categorizations of fine-tuned GPT-4o Mini and BERT (ATT or FF) antecedent links.

else), while the system identifies either a link to a previous message or a link from a subsequent message.

System Single: The system identifies this as a singular message (not linked to anything else), while the gold annotation identifies either a link to a previous message or a link from a subsequent message.

None: Both the system and the gold annotation identify at least one link (either to a previous message or from a subsequent message), but there is no overlap between the conversations.

Not Annotated: The system did not provide an annotation.

The categorizations of links from fine-tuned GPT-4o Mini and BERT (ATT or FF) appear in Table 2.

For fine-tuned GPT-4o Mini we distinguish between results with 10 test logs and 9 test logs (excluding 2010-08-17_18), because of an anomaly with one of the test logs (2010-08-17_18), where fine-tuned GPT-4o Mini identifies 184 messages (37%) as singular while the gold annotation identifies them as part of a larger conversation; these are mostly concentrated at the beginning of the file (171 of the first 210 messages). Table 3 in the Appendix includes results per test log for the fine-tuned GPT-4o Mini model.

The discrepancies between fine-tuned GPT-4o Mini and the gold annotation are fairly limited. Fine-tuned GPT-4o Mini agrees with the gold annotation on 72% of the links (75% excluding 2010-08-17_18), and identifies a link in an overlapping conversation with the gold annotation in a further 22% of messages (22% excluding 2010-08-17_18).

For the BERT+ATT and BERT+FF models, it is expected that the voting ensembles would underproduce links and that the union ensembles would overproduce links. The results show that the voting ensembles better match the exact links provided by the gold annotations, presumably because multiple linking is not so common in the gold annotations. The feedforward voting ensemble is the BERT system with the highest number of identical links (71%), that is, messages for which the full set of links is identical to that of the gold standard. This is still lower than the corresponding value for fine-tuned GPT-4o Mini when excluding the anomalous test logs. Part of the reason for this lower performance may be due to a structural limitation of the voting ensemble: the gold standard has multiple antecedent links for 183 messages (3.7%), and these can never receive an exact match from a voting ensemble that produces only a single antecedent for each message. Fine-tuned GPT-4o Mini, on the other hand, is able to exactly match messages with multiple antecedents, while not overproducing multiple links like the union ensembles.

Due to the problem with 2010-08-17_18 test log, fine-tuned GPT-4o Mini (10 test logs) generated the highest number of “no match” results (6% of the test messages) followed by the attention voting ensemble (4% of the test messages). For fine-tuned GPT-4o Mini (9 test logs) only 2% of the test messages were categorized as “no match”.

Thus, fine-tuned GPT-4o Mini (9 test logs) is the best model, balancing having the highest number of identical links and the lowest number of “no match” outputs. The Union models also had a relatively small number of “no match” outputs but their identical links were low compared to the rest of the models.

5.2. Visualization

In addition to performance analysis, we wanted to get a sense of how human (gold) and automatic (system) annotations compare. Thus, we developed visualization software that shows the IRC chat log messages together with the links between the messages and individual conversations. In this context, a conversation is a cluster of connected messages – each message in a conversation is connected to at least one other message from the conversation and no message outside of the conversation connects to a message in the conversation.

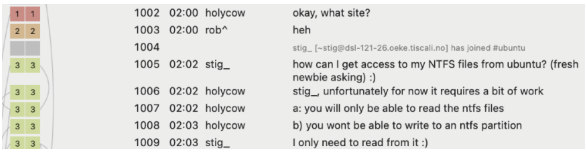

Figure 2: Example visualization of fine-tuned GPT-4o Mini output.

An example of the visualization is shown in Figure 2, with longer examples from different systems in Figures 3, 4, and 5 in the Appendix. The message id is followed by the message time, sender id, and the message content. Most of the IRC messages are human generated and are shown in normal font. Some messages are system notifications, and are shown in smaller lighter font.

The annotations are represented by the color squares on the left side of the window. There are two columns, left and right, corresponding to the gold and system annotations, respectively. A colored box with the number inside corresponds to a cluster with multiple messages. A dark gray box is a message that is a singleton cluster (not linked to any other messages).

We also draw links between messages as gray arcs connecting the boxes. As there are too many links, the screenshot only shows the links that are different between two annotations. For example, Figure 2 shows that gold annotation connects message 1007 to message 1006, while the system annotation connects 1007 to 1005. While both systems connect message 1008 to the same message, in this mode (where we only see links that differ between the gold annotation and the system annotation) we cannot tell which one. A user can change the mode and show all the links by changing her selection in window toolbar menu.

We assign the numbers to clusters sequentially, starting from the conversation with the earliest message. We do it first for the gold annotation. For the system annotation, we start with the longest cluster and assign it both the number and the color of the gold's cluster that has the largest overlap with it.

Overall, we see some notable differences in links between annotations, but conversations are rather similar between gold and system annotations.

6. Conclusion and Future Work

We presented an experiment on conversation disentanglement comparing GPT-4o and fine-tuned GPT-4o Mini with small-size models learned from scratch. We showed that, using the same amount of training data, out-of-the-box GPT-4o performs poorly, and fine-tuning GPT-4o Mini results in performance comparable to learning small-size models from scratch (based on standard hand-crafted features for this task). However, the fine-tuned GPT-4o Mini model underperforms when compared to models that utilize complex structural information. Of course, it is much easier to fine-tune an LLM rather than learn a model from scratch but in some cases there can be advantages in opting for the more complex approach of learning a model from scratch (e.g., when there are privacy considerations or device limitations).

We provided a new method for detailed analysis of the successes and failures of our models, which is an attempt at understanding a model's performance using both links and conversations, unlike standard metrics which focus either on links or conversations but not both. We also presented a new method for visualizing the outputs of all models with respect to the gold-standard annotations. To our knowledge, this is the first experiment in the literature comparing fine-tuned LLMs (specifically GPT-4o Mini) with small-size models learned from scratch for conversation disentanglement.

There are several possible directions for future work. First, it is important to test other LLMs in this task, not only GPT-4o. Second, even though our results show that fine-tuning an LLM results in high performance, it is worthwhile to investigate other methods for developing small-size models from scratch, especially considering that generally performance of machine learning models is largely dependent on the size of the training set. In our experiments we used the full training set of the Ubuntu IRC corpus and we intend to repeat our experiments varying the training set size. It is an open research question if and how much performance will be affected by using less training data. Third, as mentioned in subsection 4.2.2, for our experiments we used BERT embeddings where a vector representation of an utterance is based on averaging over word embeddings of the final layer of BERT. It is worth investigating the effect of other types of embeddings on performance. As we saw in section 2, Lam and Yang (2025) investigated the effectiveness of BERT, XLNet, ELECTRA, RoBERTa, DeBERTa, and ModernBERT in conversation disentanglement, and found that the type of embedding model made a difference. Such comparisons have also been done for other tasks, for example, for predicting user satisfaction ratings, Georgila (2024) compared various types of embeddings, and again showed that performance can vary depending on the embedding model. Thus, it is possible that newer embedding models will result in performance improvements.

We also want to examine other corpora involving multiple participants and threads, from other media than IRC, including face to face and radio communications as well as other forms of computer messaging.

7. Limitations

As discussed in section 6, our comparison between LLMs (fine-tuned or not) and models learned from scratch is only based on a handful of models. Ideally there should be a comparison of different types of LLMs (not only GPT-4o) and a variety of small-size models learned from scratch. Moreover, the effect of the training set size should be measured.

Last but not least, we use the standard Ubuntu IRC dataset which has been used by other researchers in the field, but we acknowledge the need for testing on other appropriate corpora. For reasons explained in section 2, we do not think synthetic datasets (used by other researchers for this task) are relevant, given the unnatural assumptions in their creation and resulting unreliability of their "ground truth" labelling. Thus, as a community, we need to develop other realistic benchmark datasets for this task.

8. Acknowledgements

This work was supported by the U.S. Army Research Office under Cooperative Agreement Numbers W911NF-20-2- 0053 and W911NF-25-2-0040.

9. Bibliographical References

Karan Bhukar, Harshit Kumar, Dinesh Raghu, and Ajay Gupta. 2023. End-to-end deep reinforcement learning for conversation disentanglement. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023*, pages 12571–12579, Washington DC, USA. AAAI Press.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Micha Elsner and Eugene Charniak. 2008. You talking to me? A corpus and algorithm for conversation disentanglement. In *Proceedings of ACL-08: HLT*, pages 834–842, Columbus, Ohio, USA. Association for Computational Linguistics.

Kallirroi Georgila. 2024. Comparing pre-trained embeddings and domain-independent features for regression-based evaluation of task-oriented dialogue systems. In *Proceedings of the 25th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 610–623, Kyoto, Japan. Association for Computational Linguistics.

Kallirroi Georgila, Carla Gordon, Anton Leuski, Ron Artstein, and David Traum. 2024. Studying team effectiveness via dialogue analysis. In *Proceedings of the Interservice/Industry Training, Simulation and Education Conference (I/ITSEC)*, Orlando, Florida, USA.

Kallirroi Georgila, Carla Gordon, Anton Leuski, Ron Artstein, and David Traum. 2025. Can dialogue features help predict team performance? In *Proceedings of the Interservice/Industry Training, Simulation and Education Conference (I/ITSEC)*, Orlando, Florida, USA.

Jyun-Yu Jiang, Francine Chen, Yan-Ying Chen, and Wei Wang. 2018. Learning to disentangle interleaved conversational threads with a Siamese hierarchical network and similarity ranking. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1812–1822, New Orleans, Louisiana, USA. Association for Computational Linguistics.

Seiya Kawano, Koichiro Yoshino, David Traum, and Satoshi Nakamura. 2023. End-to-end dialogue structure parsing on multi-floor dialogue based on multi-task learning. *Frontiers in Robotics and AI*, 10.

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

Tung-Thien Lam and Cheng-Zen Yang. 2025. Revisiting pre-trained language models for conversation disentanglement. In *Proceedings of the 37th Conference on Computational Linguistics and Speech Processing (ROCLING 2025)*, pages 296–302, National Taiwan University, Taipei City, Taiwan. Association for Computational Linguistics.

Bobo Li, Hao Fei, Fei Li, Shengqiong Wu, Lizi Liao, Yinwei Wei, Tat-Seng Chua, and Donghong Ji. 2024. Revisiting conversation discourse for dialogue disentanglement. *ACM Transactions on Information Systems*, 43(1).

Tianda Li, Jia-Chen Gu, Xiaodan Zhu, Quan Liu, Zhen-Hua Ling, Zhiming Su, and Si Wei. 2021. DialBERT: A hierarchical pre-trained model for conversation disentanglement.

Hui Liu, Zhan Shi, Jia-Chen Gu, Quan Liu, Si Wei, and Xiaodan Zhu. 2020. End-to-end transition-based online dialogue disentanglement. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, (IJCAI-20)*, pages 3868–3874, Yokohama, Japan.

Stephanie M. Lukin, Claire Bonial, Matthew Marge, Taylor A. Hudson, Cory J. Hayes, Kimberly Pollard, Anthony Baker, Ashley N. Foots, Ron Artstein, Felix Gervits, Mitchell Abrams, Cassidy Henry, Lucia Donatelli, Anton Leuski, Susan G. Hill, David Traum, and Clare Voss. 2024. SCOUT: A situated and multi-modal human-robot dialogue corpus. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 14445–14458, Torino, Italia. ELRA and ICCL.

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

Shikib Mehri and Giuseppe Carenini. 2017. Chat disentanglement: Identifying semantic reply relationships with random forests and recurrent neural networks. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 615–623, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Marina Meilă. 2007. Comparing clusterings—an information based distance. *Journal of Multivariate Analysis*, 98(5):873–895.

Ming Tan, Dakuo Wang, Yupeng Gao, Haoyu Wang, Saloni Potdar, Xiaoxiao Guo, Shiyu Chang, and Mo Yu. 2019. Context-aware conversation thread detection in multi-party chat. 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 6456–6461, Hong Kong, China. Association for Computational Linguistics.

David Traum, Cassidy Henry, Stephanie Lukin, Ron Artstein, Felix Gervits, Kimberly Pollard, Claire Bonial, Su Lei, Clare Voss, Matthew Marge, Cory Hayes, and Susan Hill. 2018. Dialogue structure annotation for multi-floor interaction. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Tao Yu and Shafiq Joty. 2020. Online conversation disentanglement with pointer networks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6321–6330, Online. Association for Computational Linguistics.

Henghui Zhu, Feng Nan, Zhiguo Wang, Ramesh Nallapati, and Bing Xiang. 2020. Who did they respond to? Conversation structure modeling using masked hierarchical transformer. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020*, pages 9741–9748, New York, New York, USA. AAAI Press.

Rongxin Zhu, Jey Han Lau, and Jianzhong Qi. 2021. Findings on conversation disentanglement. In *Proceedings of the 19th Annual Workshop of the Australasian Language Technology Association*, pages 1–11, Online. Australasian Language Technology Association.

10. Language Resource References

Jonathan K. Kummerfeld and Sai R. Gouravajhala and Joseph J. Peper and Vignesh Athreya and Chulaka Gunasekara and Jatin Ganhotra and Siva Sankalp Patel and Lazaros C. Polymenakos and Walter Lasecki. 2019. *IRC Disentanglement Corpus*.

11. Optional Supplementary Materials: Appendices, Software, and Data

11.1. Appendices

11.1.1. Detailed Fine-Tuned GPT-4o Mini Results

Table 3 shows the breakdown of link types per test log, including the anomaly for the 2010-08-17_18 test log, where many of the early messages are annotated as singular.

11.1.2. Visualizations

Here we provide more detailed descriptions and examples of the visualizations described in section 5.2. A light gray box corresponds to a message that has not been annotated – in the original

Test Set	Full Match	Same Link	Same Conv	Conv Overlap	No Match				Total
					G-single	S-single	None	N/A	
2005-07-06_14	202	204	16	65	1	5	7	0	500
2007-01-11_12	184	205	2	99	4	3	3	0	500
2007-12-01_03	57	314	4	109	1	9	6	0	500
2008-07-14_18	95	268	7	119	0	5	3	3	500
2010-08-17_18	159	83	26	47	0	184	1	0	500
2013-09-01_02	95	252	15	123	1	14	0	0	500
2014-06-18_13	195	194	39	66	2	1	3	0	500
2015-03-18_05	123	240	15	99	13	8	2	0	500
2016-02-22_17	181	184	49	82	1	3	0	0	500
2016-06-08_07	191	196	29	71	3	8	2	0	500
Total	1482	2140	202	880	26	240	27	3	5000
Percent	29.64	42.80	4.04	17.60	0.52	4.80	0.54	0.06	100.00
9 Test Logs	1323	2057	176	833	26	56	26	3	4500
Percent	29.40	45.71	3.91	18.51	0.58	1.24	0.58	0.07	100.00

Table 3: Categorization of fine-tuned GPT-4o Mini antecedent links including results for each of the 10 test logs.

experimental design only messages with id 1000 and above were considered, the rest were provided for context (Kummerfeld et al., 2019). A message with id 1000 and above can still be connected to a message with id below 1000.

An example of conversation overlap can be seen in Figure 5, which shows that messages 1013, 1014, and 1015 are assigned to cluster 1 by both annotations. But in contrast to the gold annotation, the system does not connect them to messages 993 or 1002, so those two messages form a separate cluster 44 in the system annotation.

Figure 3: Example visualization of fine-tuned GPT-4o Mini output.

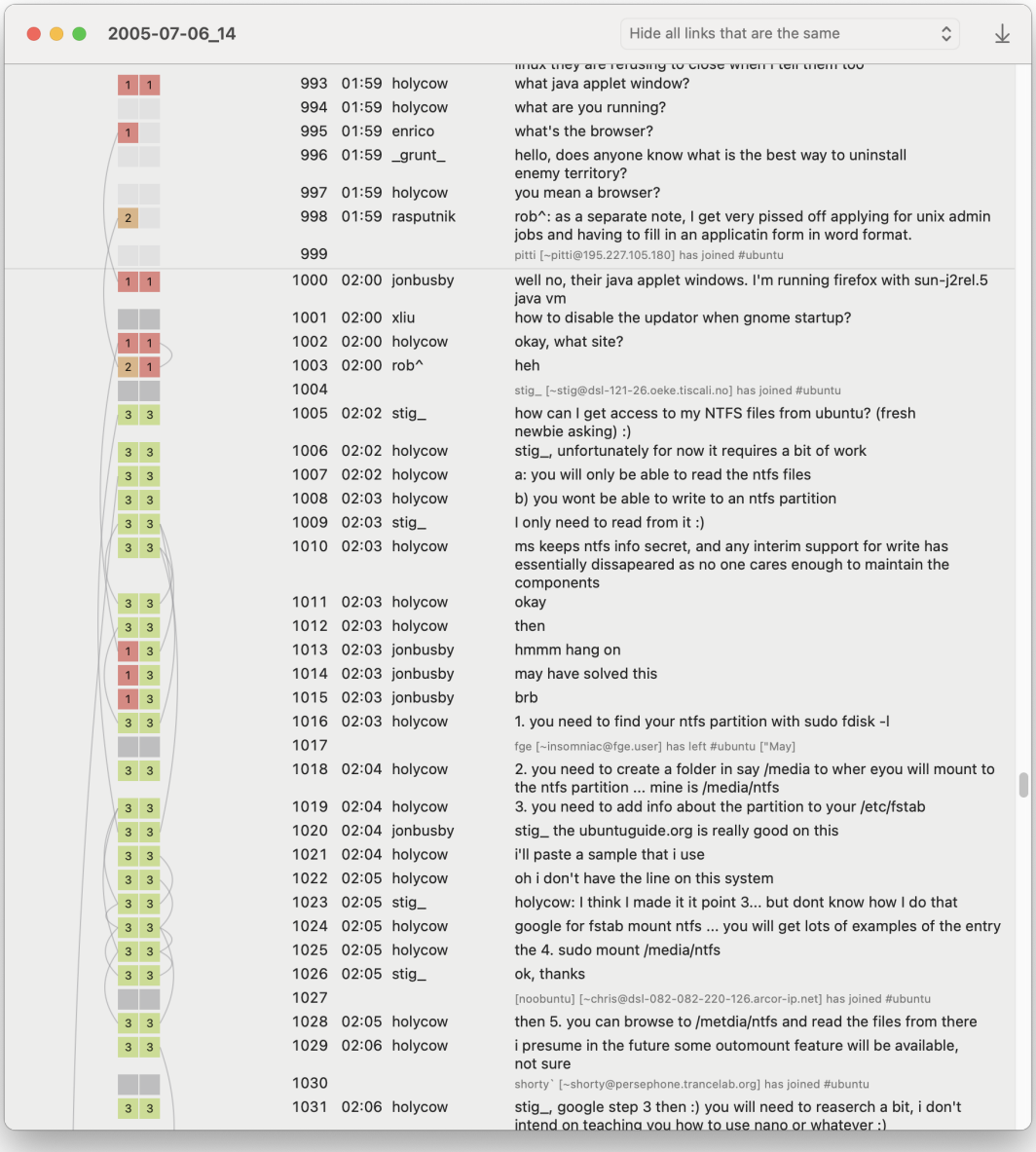

Figure 4: Example visualization of BERT feedforward output (model instance using seed 42).

Figure 5: Example visualization of BERT with attention output (model instance using seed 42).