ChatGPT vs. Crowdsourcing vs. Experts: Annotating Open-Domain Conversations with Speech Functions

Lidiia Ostyakova[1,2*]
ostyakova.ln@gmail.com

Veronika Smilga[1*]
smilgaveronika@gmail.com

Kseniia Petukhova [1*]
petukhova.ka@mipt.ru

Maria Molchanova[1]
molchanova@deeppavlov.ai

Daniel Kornev[1]
daniel@kornevs.org

[1]Moscow Institute of Physics and Technology, Russia
[2]HSE University, Russia

Abstract

This paper deals with the task of annotating open-domain conversations with speech functions. We propose a semi-automated method for annotating dialogs following the topic-oriented, multi-layered taxonomy of speech functions with the use of hierarchical guidelines using Large Language Models. These guidelines comprise simple questions about the topic and speaker change, sentence types, pragmatic aspects of the utterance, and examples that aid untrained annotators in understanding the taxonomy. We compare the results of dialog annotation performed by experts, crowdsourcing workers, and ChatGPT. To improve the performance of ChatGPT, several experiments utilising different prompt engineering techniques were conducted. We demonstrate that in some cases large language models can achieve human-like performance following a multi-step tree-like annotation pipeline on complex discourse annotation, which is usually challenging and costly in terms of time and money when performed by humans.

1 Introduction

Discourse analysis as a method of an abstract dialog representation is used in various NLP tasks: dialog management (Liang et al., 2020; Galitsky and Ilvovsky, 2017), dialog generation (Yang et al., 2022; Gu et al., 2021), dialog summarization (Chen et al., 2021), emotion recognition (Shou et al., 2022), etc. Mostly, discourse structure is considered to be an interconnected system of linguistic features such as a topic, pragmatics, and semantics. One of the main goals of discourse analysis is to describe pragmatics of actions performed by speakers within a communicative process, i.e., characterise the interlocutors' intentions at a certain moment of their interaction (Coulthard, 2014).

Despite the fact that there are numerous theoretical approaches to dialog discourse analysis, only a few of them consider the complexity of conversational nature and allow for annotation on multiple levels (Bunt et al., 2010, 2012; Cai et al., 2023). In this paper, we propose a multi-dimensional and hierarchical taxonomy of speech functions introduced by Eggins and Slade (2004) as an alternative for abstract dialog representation. In contrast to other annotation schemes, this taxonomy is topic-oriented and includes classes that are very similar in terms of pragmatics (see Appendix B). The taxonomy provides a comprehensive, systematic discourse model of dialogues (see Figure1).

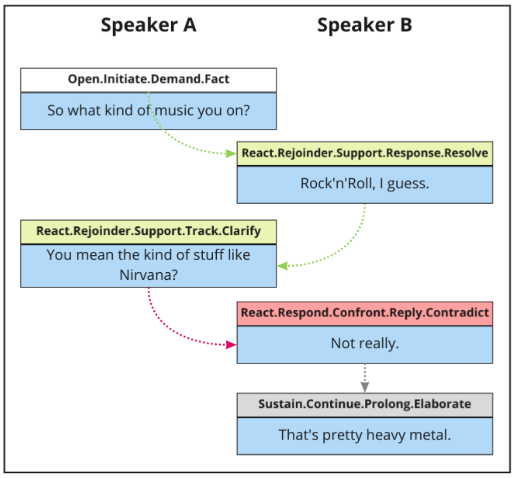

Discourse pattern
Speaker A
Speaker B
Open.Initiate.Demand.Fact
So what kind of music you on?
React.Rejoinder.Support.Response.Resolve
Rock'n'Roll, I guess.
React.Rejoinder.Support.Track.Clarify
You mean the kind of stuff like Nirvana?
React.Respond.Confront.Reply.Contradict
Not really.
Sustain.Continue.Prolong.Elaborate
That's pretty heavy metal.

Figure 1: Example of Dialog Annotation with Speech Functions

Traditionally, discourse annotation is performed manually by trained experts or crowdsourcing workers (Hoek and Scholman, 2017). Automating it partially or entirely is key to making this complicated process faster and cheaper. We argue that in complex discourse annotation tasks Large Language Models (LLMs) can be used to establish decent quality silver standards that would be later checked and improved by expert annotators.

In this paper, we annotate DailyDialog (Li et al., 2017), a multi-turn casual dialog dataset, using

[*] These authors contributed equally to this work

Proceedings of the 24th Meeting of the Special Interest Group on Discourse and Dialogue, pages 242–254
September 11–15, 2023. ©2023 Association for Computational Linguistics

the speech function taxonomy. The annotation is conducted in three ways: 1) by experts with at least B.A. in Linguistics; 2) by workers of Toloka [1], a crowdsourcing platform; 3) with the use of a large language model, specifically, ChatGPT (gpt−3.5−turbo). We then compare the performance of crowdsourcers and ChatGPT using expert annotation results as the gold standard and analyse the findings to prove that LLMs can achieve human-like performance on complex discourse annotation tasks. Finally, we release the repository with all the code we used to perform the annotation with ChatGPT [2].

2 Related Work

Theoretical Approaches to Discourse Analysis There are two basic theoretical approaches to the abstract dialog representation: Segmented Discourse Representation theory (SDRT) (Lascarides and Asher, 2007), which applies principles of Rhetorical Structures theory (RST) (Mann and Thompson, 1988) to the dialog, and theory of dialog acts (DA theory) (Core and Allen, 1997). According to the SDRT style, firstly, a relation between two elementary discourse units (EDUs) needs to be defined and then characterized with a discourse class (for instance, Question-Answer, Clarification, etc.). While SDRT represents a dialog structure as a graph (Asher et al., 2016; Li et al., 2020), most of DA theory interpretations such as DAMSL (Allen and Core, 1997), SWBD-DAMSL (Jurafsky, 1997), MIDAS (Yu and Yu, 2019) describe it sequentially giving pragmatic characteristics to each EDU. In addition, most classes used in DA taxonomies do not represent pragmatic purposes but rather focus on semantics or grammar form of utterances within a dialog, using tags such as 'yes/no question', 'statement', 'positive answer'.

To represent the discourse structure of dialogs in a more advanced way, Bunt et al. (2010, 2012) suggested Dialogue Annotation Markup Language (DiAML), a taxonomy including nine functional dimensions and 49 specific classes. Even though DiAML is claimed to be an ISO standard for DA annotation, it is challenging to apply it to real-world problems for several reasons. First, DiAML supports multi-label annotation, i.e., several classes can be assigned to one EDU, which complicates automatic classification. Moreover, there is not enough labelled data to experiment with the taxonomy. One more taxonomy designed to represent a conversational structure on several levels is Dependency Dialogue Acts (DDA) (Cai et al., 2023). A combination of dialog acts and rhetorical relations in the SDRT style showed a potential of applying multi-layered and multi-dimensional approaches for analyzing discourse structure within conversations. However, because there is no annotated data with this taxonomy, it is not clear whether it is applicable to automated tasks.

The taxonomy of speech functions is an alternative multidimensional scheme for discourse annotation introduced by Eggins and Slade (2004). It is multi-layer and hierarchical, which allows us to analyze dialog structure in a consistent manner. Unlike other multidimensional schemes, the taxonomy of speech functions supports single-label annotation. While inheriting the principle of assigning one label to a specific EDU from DA theory, speech functions taxonomy also considers relationships between utterances following the SDRT style. The tag of a current label is determined in connection with the previous one, so it is important to take into account the utterances' previous context when assigning the correct label. The potential of applying the taxonomy to manage a conversational flow within dialog systems is proven by several studies (Mattar and Wachsmuth, 2012; Kuznetsov et al., 2021; Baymurzina et al., 2021).

Large Language Models for Discourse Annotation In the recent years, the paradigm of training and using NLP models has undergone significant changes. With the advance of Large Language Models (LLMs), the focus has shifted from the previously dominating "pre-train, fine-tune" procedure to "pre-train, prompt, and predict" (Liu et al., 2023), where an LLM is applied to downstream tasks directly. In this case, textual prompts are used to guide the models' behaviour and achieve the desired output without additional fine-tuning. Scaling up LLMs to billions of parameters leads to significantly improved results in terms of few-shot and zero-shot prompting (Brown et al., 2020; Wei et al., 2021, *i.a*). However, as the objective of training most LLMs is not following the instructions but simply predicting the next token, they may fail to perform the task. One solution is fine-tuning LLMs using Reinforcement Learning from Human Feedback (RLHF) (Christiano et al., 2017) to align

[1] https://toloka.ai/tolokers/

[2] https://github.com/deeppavlov/sf_corpus/

its behaviour in accordance with the trainers' values and needs (Ouyang et al., 2022). An example of such model is ChatGPT (OpenAI, 2022) that has shown state-of-the-art or comparable performance on a number of NLP tasks in few-shot or zero-shot setting and provoked a tide of research articles testing its capabilities in areas ranging from coding and bug-fixing (Tian et al., 2023; Kashefi and Mukerji, 2023; Sobania et al., 2023, *i.a*) to medical applications (Nori et al., 2023; Kung et al., 2023, *i.a*).

There have already been claims that ChatGPT (gpt–3.5–turbo) and (gpt–4) versions alike) outperforms crowdsourcing workers on a number of annotation tasks while being significantly cheaper and faster. The tasks in question included annotation of relevance, stance, topics, and frames detection (Gilardi et al., 2023); political affiliation classification of tweets (Törnberg, 2023); hate speech detection (Huang et al., 2023; Li et al., 2023; Zhu et al., 2023); sentiment analysis and bot detection (Zhu et al., 2023). In the above-listed works, the approach to obtaining the final label was straightforwardly simple. With one prompt containing textual instruction and a datapoint, the model either answered a single question about the datapoint, assigning a label to it, or scored the probability of the datapoint belonging to some class.

However, there still have been no attempts to apply LLMs to complex annotation tasks that deal with tens of labels and require multi-step reasoning. In this work, we test whether it is possible for LLMs to achieve human-like performance on such tasks. In particular, we use ChatGPT (gpt–3.5–turbo) to annotate a dialog corpus using complex multi-layer speech function taxonomy and experiment with various prompting techniques to find out which one yields the best results.

3 Taxonomy of Speech Functions

Although the original taxonomy of speech functions included 45 classes, we reduced it to 32 labels (see Appendix B). Created for analysing casual conversations, every speech function describes several functional dimensions performed on different segmentation levels. This approach allows for annotating all the speaker's intentions and communicative actions at each moment of the dialog.

3.1 Functional Dimensions

The tag set consists of speech functions representing five different functional dimensions (Eggins and Slade, 2004). The dimensions are embedded in speech functions but distributed unevenly between tags: from two to five dimensions can be featured in one speech functions (see Figure 2).

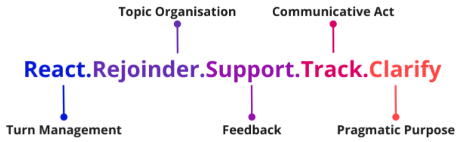


Figure 2: Example of Speech Function Structure

Turn Management denotes a speaker change at the current moment of conversation, which is represented in all speech functions except Opening moves defining a new topic. At this functional level, a *Sustain* label indicates that a speaker continues the conversation, whereas a *React* label implies that a speaker changes or the same speaker reacts to previous utterances of an interlocutor.

Topic Organisation level denotes the beginning of the dialog or a topic shift, as well as the development of a topic. *Open moves* are used to indicate the start of a dialog or a new topic. Sustain moves include a *Continue* label that shows a progression of the current topic. The *Respond* label is embedded in Reaction moves to define classes that are more likely to end the dialog and do not contribute to the topic's development. Such classes encounter more passive responses in the form of answers, back channelling, and continuation of previous narration. *Rejoinder* labels, on the other hand, define more active development of the conversation topic that has an impact on the dialog flow.

Feedback level is used to more accurately characterise moves of Reaction. *Confront* and *Support* labels indicate whether a speaker is challenging or supporting an interlocutor.

Communicative Acts are used to specify groups of pragmatic purposes that are very close in terms of interpretation and united by the same functionality within conversation. For instance, *Prolong* group includes those speech functions whose common functionality is to continue a narration supported by the same speaker (see Appendix B).

Pragmatic Purposes level is the last one in hierarchical taxonomy of speech functions specifying speakers' intentions. This layer of annotation is considered to be the most challenging for annotation as those are very pragmatically similar classes. Although speech functions from the *Track* group share the same functionality, they're performed with different pragmatic purposes in the dialog: *Check*, *Confirm*, *Clarify*, or *Probe* (see Appendix B).

It is important to note that speech function taxonomy is flexible enough as there is a potential of enriching the scheme with additional annotation layers indicating different features of utterances.

3.2 Levels of Segmentation

Bunt et al. (2012) defined EDUs as 'functional segments' and claimed that a speaker can perform several functions within one utterance. So, the boundaries of elementary discourse units are determined by communicative actions' functions depending on a chosen taxonomy. As a taxonomy of speech functions is topic-oriented, the first level of segmentation is determined by a topic shift in the dialog. Utterances united by a specific topic compound a **discourse pattern** (see Figure 1). Every discourse pattern is segmented into **turns** defined by a speaker change that can include one or several **utterances**. In most cases, utterance boundaries coincide with sentence boundaries, but some speech functions demand a finer division or a combination of several sentences. Every utterance is actually a functional segment characterized by a particular speech function.

4 Human Annotation using Speech Function Taxonomy

The annotation of discourse structures or dialog acts is not a simple task as it requires either linguistic knowledge or trained workers (Yung et al., 2019). Additionally, understanding the speaker' intentions in utterances can vary among individuals, further complicating the task. In this section, we compare the results of speech function annotation completed by experts with professional backgrounds in linguistics and crowdsourcing assessors. To evaluate the agreement between the experts and between the assessors, we use Fleiss' kappa that is an extension of Scott's pi (π) for two coders. Fleiss' kappa can deal with any number of annotators, where every item is not necessarily annotated by each annotator. It is the most commonly used method to evaluate taxonomy reliability in tasks related to discourse analysis. However, this method has the limitation of not considering the common mistakes of annotators. Therefore, we measured not only inter-annotator agreement but also three most common metrics for multi-class classification tasks with imbalanced data — Macro F1, Weighted Precision and Weighted Recall, by comparing the workers' annotations to the results of experts.

4.1 Tree-like Design of Annotation Instruction

To facilitate annotation, we designed a tree-like scheme comprised of a series of questions and their corresponding answer options that reproduces logic of a hierarchy of speech functions taxonomy. Due to multidimensional structure of speech functions, the path to each final label can be represented as a series of straightforward questions in form of instructions. This tree-like structure was used by both experts and annotators during annotation process.

4.2 Crowdsourcing Process

For crowdsourcing, we used Toloka platform for data annotation enabling project management and review cycles. When carrying out complex discourse annotation, the following two main problems are encountered:

- pragmatic classes are difficult to differentiate for annotators without a strong linguistic background;
- an issue of unreliable annotators who prioritize speed over accuracy.

To address the first issue, we used a tree-like design of guidelines rather than asking to choose one of 32 different speech functions directly. At each stage of annotation, a crowdsourcing worker answers a simple question with 2-4 possible options. An instruction with explanations and examples is attached to each question. Having answered all the questions in the chain, the annotator reaches the final label.

As for the second problem, we developed several mechanisms for tracking the quality of answers, including (1) detecting the fast answers that are selected without reading instructions, (2) checking answer consistency across related questions, and (3) using trained classifiers to detect answers that do not match the expected annotation.

Furthermore, we developed multi-level qualification tasks to enhance the quality of dialog annotations. The first stage involves both training and the exam process on a single dialog, with hints shown to crowdsourcing workers if they answer incorrectly. Workers who fail to achieve the appropriate quality can retry one more time. Those who pass the examination are selected for the main annotation pool. Each dialog is evaluated based on custom validation rules and control questions. If the dialog fails validation, annotators cannot continue the annotation.

4.3 Crowdsourcing vs. Experts

As the source of dialog data, we used DailyDialog (Li et al., 2017), a hand-crafted dataset of multi-turn casual human conversations about daily life. First, we splitted the utterances into EDUs. Second, three non-native experts with at least B.A. in Linguistics annotated 64 dialogs (1030 utterances). In cases where there was a lack of consensus among the expert annotators, and a majority vote could not be established, we considered all expert responses as correct and included them in the final gold standard. This decision was made due to the understanding that people may perceive the intentions of the speaker differently. Third, the same data was annotated via crowdsourcing with three non-professional workers annotating each dialog. The key criterion for recruitment was the successful completion of the test task assessing the annotators' labeling quality. This test automatically evaluated the annotator's ability to perform the required dialogue annotation tasks. Additionally, we emphasized implementing validation systems to filter out low-quality responses. Access to the test task was granted to those who previosuly passed the English language proficiency test on the Toloka platform. Statistical data shows that while crowdsourcers from many countries participated in the annotation process, the largest number of annotators originated from Brazil and Egypt. The minimum age of crowdsourcers was 19 years, with an average age of 27.

We evaluated the results for 16 high-level cut labels and the complete taxonomy to identify the weak points of the established hierarchical guidelines (see Appendix B for an overview of taxonomy). We also examined cases of voting, in which the majority of annotators agreed on a tag. The cut labels were labeled with high accuracy by crowdsourcing workers, while the annotation of full tags was more challenging for non-experts, as proven by all metrics. Macro F1 value indicates that improving the quality of annotating low-level classes is necessary (see Table 3a). Fleiss' Kappa revealed that differentiating tags with similar pragmatics is difficult not only for untrained workers but also for experts. Nonetheless, the chosen taxonomy is quite reliable, as Fleiss' Kappa for experts' annotation is more than 0.6, standing for substantial agreement (see Figure 3).

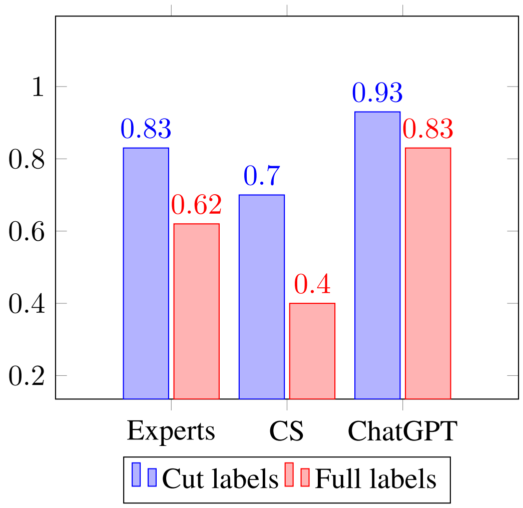
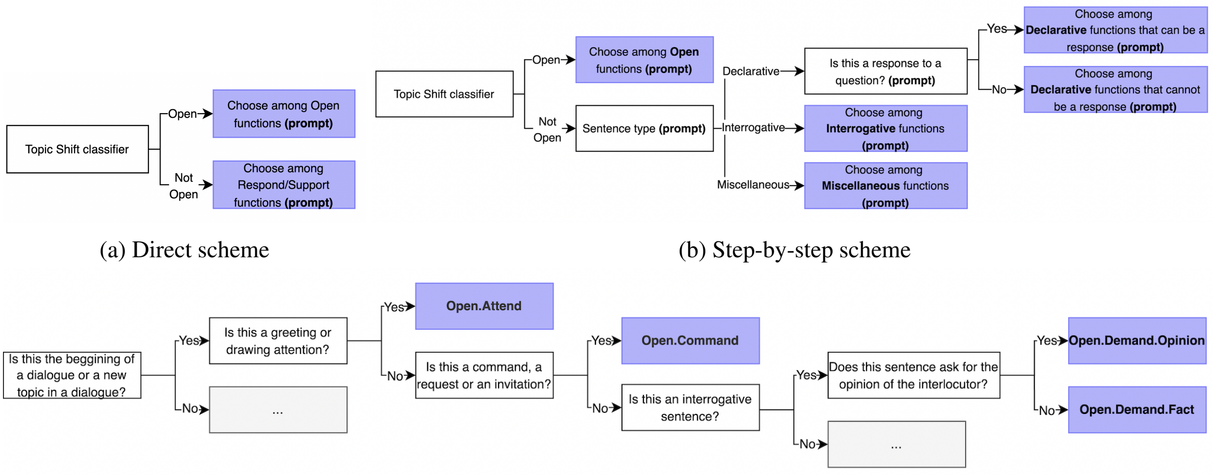


Figure 3: Inter-annotator Agreement (CS - crowdsourcing)

The use of speech function taxonomy implies a noticeable class imbalance, with certain speech functions occurring more frequently than others (see confusion matrix 6a in Appendix A). Classes that have a limited number of examples are Rebound, Re-challenge, Refute, etc. Certain classes are well-defined and easily distinguishable, including Open.Attend, Register, Resolve, Clarify, and Open.Demand.Fact. However, the classes of Extend, Enhance, and Elaborate are challenging to distinguish accurately because they are very close in terms of pragmatics.

5 Methods

The annotation task in question required careful instruction preparation even for human annotators as opposed to simpler tasks such as sentiment classification, bot detection, etc. Thus, the process of creating the best prompt for an LLM is also a challenging and multi-step process. We conduct a number of experiments in order to find the best way to use ChatGPT for complex discourse annotation tasks. In all cases, the `system_message` we used while querying ChatGPT API was "You are a professional linguist annotator who has to perform a


Topic Shift classifier
Open
Choose among Open functions (prompt)
Not Open
Choose among Respond/Support functions (prompt)
Topic Shift classifier
Open
Choose among Open functions (prompt)
Not Open
Sentence type (prompt)
Declarative
Is this a response to a question? (prompt)
Yes
Choose among Declarative functions that can be a response (prompt)
No
Choose among Declarative functions that cannot be a response (prompt)
Interrogative
Choose among Interrogative functions (prompt)
Miscellaneous
Choose among Miscellaneous functions (prompt)
Is this the beggining of a dialogue or a new topic in a dialogue?
Yes
Is this a greeting or drawing attention?
Yes
Open.Attend
No
Is this a command, a request or an invitation?
Yes
Open.Command
No
Is this an interrogative sentence?
Yes
Does this sentence ask for the opinion of the interlocutor?
Yes
Open.Demand.Opinion
No
Open.Demand.Fact
No
...
No
...

(a) Direct scheme

(b) Step-by-step scheme

(c) Tree-like scheme

Figure 4: Experiment pipelines

discourse annotation task". The user_message varied for different experiments. See Figure 5 for an example of user_message.

```
TASK: This is part of the dialog is between 2
speakers. Answer QUESTION about CURRENT UTTERANCE.
You must analyze relations between CURRENT UTTERANCE
and PREVIOUS CONTEXT.

PREVIOUS CONTEXT:
speaker_1: Hey!
speaker_1: I heard you'd annotated a corpus of 1000
utterances in just an hour!
speaker_1: Is that true?
CURRENT UTTERANCE:
speaker_2: Well, technically, I made ChatGPT do that.

QUESTION: Can this utterance be an answer to the
previous speaker's question?
POSSIBLE ANSWERS: Yes, No
You must always select an option. Provide only one
answer without explanation.
ANSWER (Yes or No):
```

Figure 5: An example of user_message.

To reduce the number of API calls and thus the time and the cost of the annotation, we also used automatic methods other than ChatGPT on some steps of the annotation. For example, in all our experiments we used Topic Shift Classifier to detect the beginning of a new topic in a dialog. It is worth noting that ChatGPT did not perform well in this particular task. The Topic Shift Classifier was trained using the DeepPavlov (Burtsev et al., 2018) library utilizing a double sequence binary classifier model based on roberta-large-mnli, with two sequential utterances as input. The true labels indicate topic change in the utterances. The following hyper-parameters were used to train the model: learning rate = 2e-5, optimizer = AdamW, input max length = 128. To successfully train the model, we used the early-stopping technique. The classifier was able to transfer the knowledge acquired during pre-training on mnli to the related problem of shift identification by using a pre-trained model (Konovalov et al., 2020; Gulyaev et al., 2020).

5.1 Choosing the best annotation scheme

First, we compare three approaches to automatic discourse annotation using ChatGPT:

- Direct annotation – providing an full list of labels to choose from;
- Step-by-step scheme with intermediate labels;
- Complex tree-like scheme with intermediate labels and yes-no questions prevailing on each step.

5.1.1 Direct annotation scheme

The most straightforward approach is providing the final labels, their description and 2 examples for each to the model as they are. However, even at this step we chose to distinguish between 6 *Open* speech functions – the ones that begin the dialog or a new topic in the dialog – and 27 *React/Sustain* speech functions via a preliminary classification step. Here, the pipeline consists of two steps. See Figure 4a for an overview.

5.1.2 Step-by-step annotation scheme

Here, the annotation process was broken down into smaller steps. The pipeline consisted of 2-5 steps depending on the outcome of each step. In the end, the model once again had to choose between several final labels (from 4 to 12). See Figure 4b for an overview.

5.1.3 Tree-like annotation scheme

In this experiment, we used complex tree-like annotation pipeline that was primarily designed to facilitate human crowdsourcing annotation process. As breaking the task of selecting one of many labels into smaller sub-tasks of a tree-like structure with simpler questions on each step is used to improve performance of humans on complex discourse annotation tasks (Scholman et al., 2016), we speculate that the same holds true for annotation via ChatGPT. Additionally, novel research suggests that making the model follow a number of tree-like structured prompts may greatly improve its performance (as applied to sudoku puzzles in Yao et al. (2023)). The major difference from the Step-by-step annotation scheme is that the Tree-like annotation scheme favours prompts containing yes-no questions over prompts asking to select one option out of many. As a results, the scheme is much more complex than the ones described before, with 2-12 steps to be completed before reaching the final label. However, the majority of questions are extremely simplified, guiding the model to the final label via a series of yes-no questions. For an example of how some final labels can be reached, see Figure 4c.

5.2 Hyperparameter tuning

While examining the results of the annotation in Subsection 5.1, we observed some cases where the model's selections appeared confused by the class names it had to choose from in the final labeling step. For example, when asked to choose from labels Check, Confirm, Clarify, and Probe, the model tended to ignore the instruction that Check is only used to get the previous speaker to repeat something, and overuse this label (see Appendix B for detailed definitions of each label). When asked to provide an explanation of its choice, the model would produce explanations based on the semantics of the word Check, e.g. "The speaker wanted to check what the previous speaker thinks". Thus, we decided to check if the performance improves if the final labels are masked, replacing the speech function name with a number and leaving the definitions and instructions intact.

We also experimented with model temperature (0.0, 0.5, 0.9), a hyperparameter that controls the randomness of the generated content.

Another feature that we tested was a modification of zero-shot Chain-of-Thought prompting as described in Kojima et al. (2022). Here, the model was asked to provide an answer in the following format: `"Reasoning: (your reasoning). The final answer: (your final answer)"`. However, in our case, generating reasoning and grounding the final answer in it did not improve the quality.

Finally, we experimented with the size of the context window (1, 3, 5), i.e., the number of previous utterances provided to the model.

6 Experiments & Results

6.1 Evaluation of annotation schemes

Due to the limitations in funding and a large number of experiments, to evaluate the different annotation schemes, we ran experiments on a subset of 12 dialogs containing 189 utterances (approximately 1/5 of the final corpus). For each scheme, we prompted ChatGPT to annotate the subset of dialogs and compared the predicted labels to the ground truth expert annotations.

Naturally, with more detailed schemes and simpler questions on each step, the model achieved better results. As Table 1 demonstrates, Macro F1 is significantly lower than Weighted Recall and Weighted Precision for complex schemes, Step-by-step and Tree-like annotation. The Speech Function annotation scheme is deemed to produce imbalanced data classes due to its nature – some classes are by definition more common and some are rare. Thus, the difference between higher Weighted Recall and Precision demonstrate that we were able to classify more common categories well as those categories have a greater influence on weighted metrics. On the opposite, as Macro F1 treats all classes equally regardless of their size, lower Macro F1 in all schemes shows that the model's performance consistently deteriorates on smaller classes.

Even though Weighted Precision is higher for less complex Step-by-step scheme, we can say that with Tree-like scheme the model performed the task better as higher Macro F1 demonstrates that it was better at distinguishing between smaller classes.

6.2 Hyperparameter evaluation

We evaluated different hyperparametrs including temperature, masking, context size, and reasoning on the Tree-like scheme. Higher temperature,

	Weighted Recall	Weighted Precision	Macro F1
Direct annotation	0.23	0.33	0.28
Step-by-step scheme	0.57	**0.75**	0.31
Tree-like scheme	**0.62**	0.67	**0.43**

Table 1: Evaluation of annotation by ChatGPT using different annotation methods (on a subset of dialogs)

meaning higher randomness and diversity, turned out to work best. The longer context seems to confuse the model, as the windows of sizes 1 and 3 performed better. The results are shown in Table 2.

Overall, there has been no significant difference in performance between the models with different hyperparameters. The best performing option turned out to be the model with temperature = 0.9, masked labels, context window = 1, and no reasoning.

6.3 Full corpus evaluation

Finally, we evaluated ChatGPT with the best hyperparameters on the full corpus of 64 dialogs. As can be seen, ChatGPT performed well on a subset of 12 dialogs (see Table 2), but on the entire dataset, it performs noticeably worse for full and cut tags. We also tried to employ the voting method when utilizing ChatGPT, similar to what was done with crowdsourcing annotation, to enhance the reliability of the annotation. We ran the annotation pipeline three times, counted the votes and got the results that are also shown in Table 3b. As can be seen from the table, the implementation of voting had minimal impact on the results. ChatGPT consistently provided answers, as indicated by the Fleiss Kappa scores of 0.83 for full tags and 0.93 for cut tags, representing an almost perfect level of agreement and model consistency, despite temperature being set to 0.9 (meaning more diverse responses).

The lower quality of the annotation by ChatGPT compared to crowdsourcing can be explained by two main reasons (see Figure 6b in Appendix A). Firstly, distinguishing between close subclasses such as Extend/Enhance/Elaborate is challenging, even for humans, and it appears to be even more difficult for ChatGPT. Additionally, ChatGPT struggles with differentiating between Acknowledge/Affirm/Agree. Secondly, ChatGPT not only has difficulties in distinguishing among subclasses, but it also frequently confuses Resolve (detailed answer) with Replies (positive and negative answers). Furthermore, it often misclassifies Extend as Affirm or Agree. In general, the difference in metrics between 12 and 64 dialogs can be explained by the individuality and complexity of each dialog, with some being significantly more complicated than others.

6.4 Cost analysis

As for cost, annotation with ChatGPT varies depending of a tree length for a particular dialog from 0.03$ to 0.07$ while crowdsourcing workers need to be paid from 0.12$ to 0.22$ for one dialog annotation. Experts spend an average of 14,5 minutes annotating one dialogue, while crowdsourcers do the same for 29 minutes. Depending on whether the model is currently overloaded or not, ChatGPT's time for task completion varies. The model can typically annotate one dialogue of average length in less than 10 minutes. So, ChatGPT can be used as a silver standard of annotation instead of crowdsourcing results, which would reduce the time and money spent on experts' post-annotation. However, working with such abstract annotation classes, it is still important to rely on non-expert annotators to make the taxonomy easy to comprehend.

7 Conclusion and Future Work

We conducted several experiments on the annotation of casual conversations with speech function taxonomy performed by experts in linguistics, crowdsourcing workers, and ChatGPT. In this paper, we took a closer look at the problems of defining multilayer taxonomies in real dialogs and, furthermore, explored whether it is possible to differentiate between those classes when annotating. Experiments with ChatGPT have demonstrated the potential of using LLMs for linguistic annotation with accuracy that is close to crowdsourcing workers' performance on some dialogs. Even though guiding the model across a tree-like structure of instructions to reach the final label seems to be promising, it still falls short of non-expert performance on such tasks and does not let the researchers explore variations in how non-experts understand discourse structures.

It is important to mention that a significant drawback of the method we propose is the neces-

Experiment	Weighted Recall	Weighted Precision	Macro F1
No masking; context=1; t=0.9	**0.62**	0.67	**0.43**
Masking; context=1; t=0.9	0.61	**0.72**	**0.43**
Masking; context=1; t=0.0	0.58	0.69	0.41
Masking; context=1; t=0.5	0.58	0.69	0.4
Masking; context=1; t=0.9; reasoning	0.58	0.67	0.42
Masking; context=3; t=0.9	0.59	**0.72**	0.41
Masking; context=5; t=0.9	0.61	0.67	0.42

Table 2: Evaluation of annotation by ChatGPT using Tree-like scheme (on a subset of dialogs)

	Weighted Recall	Weighted Precision	Macro F1
Full tags	0.56	0.67	0.44
Full tags & voting	0.6	0.71	0.46
Cut labels	0.81	0.82	0.54
Cut labels & voting	**0.84**	**0.86**	**0.59**

(a) Crowdsourcers

	Weighted Recall	Weighted Precision	Macro F1
Full tags	0.41	0.59	0.34
Full tags & voting	0.42	0.6	0.33
Cut labels	**0.74**	**0.78**	**0.5**
Cut labels & voting	0.73	0.77	0.49

(b) ChatGPT

Table 3: Evaluation of final annotation by ChatGPT and crowdsourcing workers as compared to expert annotation (all dialogs)

sity of expert involvement in writing prompts and structuring them the right way. However, with LLMs, this process turned out to be extremely similar to the process of writing instructions for non-expert crowdsourcing workers and should thus pose no difficulty to a discourse researcher.

Possible areas for the future work are: 1) trying out other instruction-based models; 2) conducting a more comprehensive selection of hyperparameters; 3) adding criticism steps to the current pipeline, enabling self-reflection and self-correction (Kim et al., 2023); 4) evolving and adapting the developed method for solving complex problems with LLMs in other applications.

Acknowledgements

We are sincerely grateful to our esteemed colleague, Dilyara Zharikova, for all of her help and guidance during the preparation of this work.

References

James Allen and Mark Core. 1997. Damsl: Dialogue act markup in several layers (draft 2.1). In *Technical Report, Multiparty Discourse Group, Discourse Resource Initiative*.

Nicholas Asher, Julie Hunter, Mathieu Morey, Farah Benamara, and Stergos Afantenos. 2016. Discourse structure and dialogue acts in multiparty dialogue: the stac corpus. In *10th International Conference on Language Resources and Evaluation (LREC 2016)*, pages 2721–2727.

Dilyara Baymurzina, Denis Kuznetsov, Dmitry Evseev, Dmitry Karpov, Alsu Sagirova, Anton Peganov, Fedor Ignatov, Elena Ermakova, Daniil Cherniavskii, Sergey Kumeyko, et al. 2021. Dream technical report for the alexa prize 4. *4th Proc. Alexa Prize*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Harry Bunt, Jan Alexandersson, Jean Carletta, Jae-Woong Choe, Alex Chengyu Fang, Kiyong Lee, Volha Petukhova, Andrei Popescu-Belis, Laurent Romary, Claudia Soria, et al. 2010. Towards an iso standard for dialogue act annotation. In *Seventh conference on International Language Resources and Evaluation (LREC'10)*.

Harry Bunt, Jan Alexandersson, Jae-Woong Choe, Alex C Fang, Koiti Hasida, Volha Petukhova, Andrei Popescu-Belis, and David Traum. 2012. Iso 24617-2: A semantically-based standard for dialogue annotation. Technical report, University of Southern California Los Angeles.

Mikhail Burtsev, Alexander Seliverstov, Rafael Airapetyan, Mikhail Arkhipov, Dilyara Baymurzina, Nickolay Bushkov, Olga Gureenkova, Taras Khakhulin, Yuri Kuratov, Denis Kuznetsov, and Vasily Konovalov. 2018. Deeppavlov: Open-source library for dialogue systems. In *NIPS*.

Jon Z Cai, Brendan King, Margaret Perkoff, Shiran Dudy, Jie Cao, Marie Grace, Natalia Wojarnik, Ananya Ganesh, James H Martin, Martha Palmer, et al. 2023. Dependency dialogue acts–annotation scheme and case study. *arXiv preprint arXiv:2302.12944*.

Yulong Chen, Yang Liu, Liang Chen, and Yue Zhang. 2021. Dialogsum: A real-life scenario dialogue summarization dataset. *arXiv preprint arXiv:2105.06762*.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30.

Mark G Core and James Allen. 1997. Coding dialogs with the damsl annotation scheme. In *AAAI fall symposium on communicative action in humans and machines*, volume 56, pages 28–35. Boston, MA.

Malcolm Coulthard. 2014. *An introduction to discourse analysis*. Routledge.

Suzanne Eggins and Diana Slade. 2004. *Analysing casual conversation*. Equinox Publishing Ltd.

Boris Galitsky and Dmitry Ilvovsky. 2017. Chatbot with a discourse structure-driven dialogue management. In *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 87–90.

Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. 2023. Chatgpt outperforms crowd-workers for text-annotation tasks. *arXiv preprint arXiv:2303.15056*.

Xiaodong Gu, Kang Min Yoo, and Jung-Woo Ha. 2021. Dialogbert: Discourse-aware response generation via learning to recover and rank utterances. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12911–12919.

Pavel Gulyaev, Eugenia Elistratova, Vasily Konovalov, Yuri Kuratov, Leonid Pugachev, and Mikhail Burtsev. 2020. Goal-oriented multi-task bert-based dialogue state tracker.

Jet Hoek and Merel Scholman. 2017. Evaluating discourse annotation: Some recent insights and new approaches. In *Proceedings of the 13th Joint ISO-ACL Workshop on Interoperable Semantic Annotation (isa-13)*.

Fan Huang, Haewoon Kwak, and Jisun An. 2023. Is chatgpt better than human annotators? potential and limitations of chatgpt in explaining implicit hate speech. *arXiv preprint arXiv:2302.07736*.

Dan Jurafsky. 1997. Switchboard swbd-damsl shallow-discourse-function annotation coders manual. *www.dcs.shef.ac.uk/nlp/amities/files/bib/ics-tr-97-02.pdf*.

Ali Kashefi and Tapan Mukerji. 2023. Chatgpt for programming numerical methods. *Journal of Machine Learning for Modeling and Computing*.

Geunwoo Kim, Pierre Baldi, and Stephen McAleer. 2023. Language models can solve computer tasks. *ArXiv*, abs/2303.17491.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *arXiv preprint arXiv:2205.11916*.

Vasily Konovalov, Pavel Gulyaev, Alexey Sorokin, Yury Kuratov, and Mikhail Burtsev. 2020. Exploring the bert cross-lingual transfer for reading comprehension. In *Komp'juternaja Lingvistika i Intellektual'nye Tehnologiithis*, pages 445–453.

Tiffany H Kung, Morgan Cheatham, Arielle Medenilla, Czarina Sillos, Lorie De Leon, Camille Elepaño, Maria Madriaga, Rimel Aggabao, Giezel Diaz-Candido, James Maningo, et al. 2023. Performance of chatgpt on usmle: Potential for ai-assisted medical education using large language models. *PLoS digital health*, 2(2):e0000198.

Denis Kuznetsov, Dmitry Evseev, Lidia Ostyakova, Oleg Serikov, Daniel Kornev, and Mikhail Burtsev. 2021. Discourse-driven integrated dialogue development environment for open-domain dialogue systems. In *Proceedings of the 2nd Workshop on Computational Approaches to Discourse*, pages 29–51, Punta Cana, Dominican Republic and Online. Association for Computational Linguistics.

Alex Lascarides and Nicholas Asher. 2007. Segmented discourse representation theory: Dynamic semantics with discourse structure. *Computing meaning*, pages 87–124.

Jiaqi Li, Ming Liu, Min-Yen Kan, Zihao Zheng, Zekun Wang, Wenqiang Lei, Ting Liu, and Bing Qin. 2020. Molweni: A challenge multiparty dialogues-based machine reading comprehension dataset with discourse structure. *arXiv preprint arXiv:2004.05080*.

Lingyao Li, Lizhou Fan, Shubham Atreja, and Libby Hemphill. 2023. " hot" chatgpt: The promise of chatgpt in detecting and discriminating hateful, offensive, and toxic comments on social media. *arXiv preprint arXiv:2304.10619*.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. Dailydialog: A manually labelled multi-turn dialogue dataset. *arXiv preprint arXiv:1710.03957*.

Kaihui Liang, Austin Chau, Yu Li, Xueyuan Lu, Dian Yu, Mingyang Zhou, Ishan Jain, Sam Davidson, Josh Arnold, Minh Nguyen, et al. 2020. Gunrock 2.0: A user adaptive social conversational system. *arXiv preprint arXiv:2011.08906*.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35.

William C Mann and Sandra A Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text-interdisciplinary Journal for the Study of Discourse*, 8(3):243–281.

Nikita Mattar and Ipke Wachsmuth. 2012. Small talk is more than chit-chat: Exploiting structures of casual conversations for a virtual agent. In *Annual Conference on Artificial Intelligence*, pages 119–130. Springer.

Harsha Nori, Nicholas King, Scott Mayer McKinney, Dean Carignan, and Eric Horvitz. 2023. Capabilities of gpt-4 on medical challenge problems. *arXiv preprint arXiv:2303.13375*.

OpenAI. 2022. Introducing chatgpt. Accessed on May 13, 2023.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Merel CJ Scholman, Jacqueline Evers-Vermeul, Ted JM Sanders, et al. 2016. A step-wise approach to discourse annotation: Towards a reliable categorization of coherence relations. *Dialogue & Discourse*, 7(2):1–28.

Yuntao Shou, Tao Meng, Wei Ai, Sihan Yang, and Keqin Li. 2022. Conversational emotion recognition studies based on graph convolutional neural networks and a dependent syntactic analysis. *Neurocomputing*, 501:629–639.

Dominik Sobania, Martin Briesch, Carol Hanna, and Justyna Petke. 2023. An analysis of the automatic bug fixing performance of chatgpt. *arXiv preprint arXiv:2301.08653*.

Haoye Tian, Weiqi Lu, Tsz On Li, Xunzhu Tang, Shing-Chi Cheung, Jacques Klein, and Tegawendé F Bissyandé. 2023. Is chatgpt the ultimate programming assistant–how far is it? *arXiv preprint arXiv:2304.11938*.

Petter Törnberg. 2023. Chatgpt-4 outperforms experts and crowd workers in annotating political twitter messages with zero-shot learning. *arXiv preprint arXiv:2304.06588*.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Yang Yang, Juan Cao, Yujun Wen, and Pengzhou Zhang. 2022. Multiturn dialogue generation by modeling sentence-level and discourse-level contexts. *Scientific Reports*, 12(1):20349.

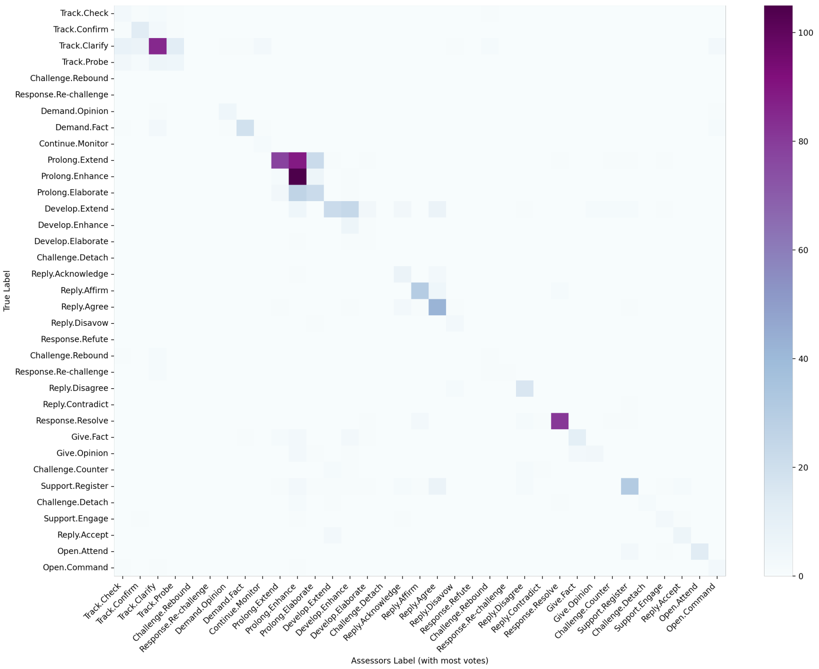
Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models.

Dian Yu and Zhou Yu. 2019. Midas: A dialog act annotation scheme for open domain human machine spoken conversations. *arXiv preprint arXiv:1908.10023*.

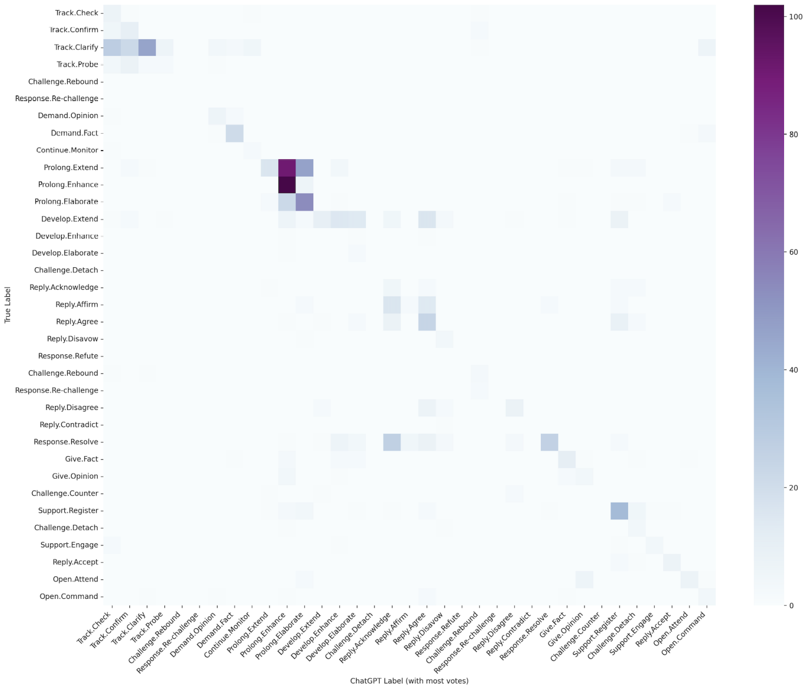
Frances Yung, Vera Demberg, and Merel Scholman. 2019. Crowdsourcing discourse relation annotations by a two-step connective insertion task. In *Proceedings of the 13th Linguistic Annotation Workshop*, pages 16–25.

Yiming Zhu, Peixian Zhang, Ehsan-Ul Haq, Pan Hui, and Gareth Tyson. 2023. Can chatgpt reproduce human-generated labels? a study of social computing tasks. *arXiv preprint arXiv:2304.10145*.

A Confusion matrices comparing crowdsourced/ChatGPT annotation with true labels



(a) Crowdsourced annotation



(b) ChatGPT annotation

B Speech Functions list

Cut labels	Full labels	Definition
Open.Demand.Fact	Open.Demand.Fact	Demanding factual information.
Open.Demand. Opinion	Open.Demand.Opinion	Demanding judgment or evaluative information from the interlocutor.
Open.Give.Fact	Open.Give.Fact	Providing factual information.
Open.Give.Opinion	Open.Give.Opinion	Providing judgment or evaluative information.

Open.Command	Open.Command	Making a request, an invitation or command to start a dialog or discussion of a new topic.
Open.Attend	Open.Attend	These are usually greetings.
React.Rejoinder. Confront.Response	React.Rejoinder.Confront. Response.Re-challenge	Offering an alternative position, often an interrogative sentence.
React.Rejoinder. Support.Track	React.Rejoinder.Support.Track. Probe	Requesting a confirmation of the information necessary to make clear the previous speaker's statement.
	React.Rejoinder.Support.Track. Check	Getting the previous speaker to repeat an element or the entire statement that the speaker has not heard or understood.
	React.Rejoinder.Support.Track. Clarify	Asking a question to get additional information on the current topic of the conversation. Requesting to clarify the information already mentioned in the dialog.
	React.Rejoinder.Support.Track. Confirm	Asking for a confirmation of the information received.
Sustain.Continue. Prolong	Sustain.Continue.Prolong. Extend	Adding supplementary or contradictory information to the previous statement.
	Sustain.Continue.Prolong. Enhance	Adding details to the previous statement, adding information about time, place, reason, etc. Clarifying / rephrasing the previous statement or giving examples to it.
	Sustain.Continue.Prolong. Elaborate	
React.Rejoinder. Confront.Challenge. Rebound	React.Rejoinder.Confront. Challenge. Rebound	Questioning the relevance, reliability of the previous statement, most often an interrogative sentence.
React.Respond. Support.Reply	React.Respond.Support.Reply. Affirm	A positive answer to a question or confirmation of the information provided. Yes/its synonyms or affirmation.
	React.Respond.Support.Reply. Acknowledge	Indicating knowledge or understanding of the information provided.
	React.Respond.Support.Reply. Agree	Agreement with the information provided. In most cases, the information that the speaker agrees with is new to him. Yes/its synonyms or affirmation.
React.Respond. Support.Develop	React.Respond.Support.Develop. Extend	Adding supplementary or contradictory information to the previous statement.
	React.Respond.Support.Develop. Enhance	Adding details to the previous statement, adding information about time, place, reason, etc.
	React.Respond.Support.Develop. Elaborate	Clarifying / rephrasing the previous statement or giving examples to it. A declarative sentence or phrase (may include for example, I mean, like).
React.Respond. Confront.Reply	React.Respond.Confront.Reply. Disagree	Negative answer to a question or denial of a statement. No, negative sentence.
	React.Respond.Confront.Reply. Contradict	Refuting previous information. No, sentence with opposite polarity. If the previous sentence is negative, then this sentence is positive, and vice versa.
	React.Respond.Confront.Reply. Disavow	Denial of knowledge or understanding of information.
Sustain.Continue. Monitor	Sustain.Continue.Monitor	Checking the involvement of the listener or trying to pass on the role of speaker to them.
Sustain.Continue. Command	Sustain.Continue.Command	Making a request, an invitation or command to start a dialog or discussion of a new topic.
React.Respond. Support.Register	React.Respond.Support.Register	A manifestation of emotions or a display of attention to the interlocutor.
React.Respond. Support.Engage	React.Respond.Support.Engage	Drawing attention or a response to a greeting.
React.Respond. Support.Reply. Accept	React.Respond.Support.Reply. Accept	Expressing gratitude.
React.Rejoinder. Support.Response. Resolve	React.Rejoinder.Support. Response.Resolve	The response provides the information requested in the question.
React.Respond. Command	React.Respond.Command	Making a request, an invitation or command to start a dialog or discussion of a new topic.
React.Rejoinder. Confront.Challenge. Detach	React.Rejoinder.Confront. Challenge.Detach	Terminating the dialog.