

1.3 Dialogue Act Annotation Scheme

This document presents the dialogue act (DA) annotation scheme which was used to annotate the CROWDSS corpus. It is adapted from Pareti and Lando (2019), who do not provide guidelines for their annotation scheme. Therefore, in an effort to streamline future annotation efforts we release our scheme along with the CROWDSS dataset and the paper. Note that we had to rely on our own interpretation of Pareti and Lando's scheme as well as that minor modifications were made to the tag set (see paper).

General annotation rules

1. DA labels are assigned to functional segments (FS) which are continuous sequences of complete tokens.
2. There can be one or more FS per utterance.
3. FS may not cross turn boundaries.
4. FS may not overlap
5. Every token must be part of an FS (although it is okay if punctuation or whitespace tokens are not within FS spans)
6. Every FS must receive exactly one DA label.
 - a. if more than one label is conceivable for a given FS, the label „must correspond to the most salient intent at that point in the conversation, i.e. the intent that we need to identify to understand the dialog state and provide a conversationally appropriate reaction“ (Pareti and Lando, 2019)

About the tagset

- **Hierarchical tagset:** The tagset is hierarchical in nature with three levels of granularity. The naming of the tags follows a hierarchical approach and inherits the full path through the schema, e.g. request.instruct.cancel. The act right of the dot is a sub-act of the one on its left.
- **Domain-agnosticism:** All tags are domain-agnostic. However, domain-specific tags could be introduced on a fourth level of hierarchy, e.g. request.query.open.booking_time
- **Expectation-based annotation:** A given FS is labelled with the tag that describes best the conversational expectation that it creates (e.g., it requests a piece of information, or proposes something that the recipient is expected to accept or reject) or that it reacts to. When an FS both reacts to a past conversational expectation and creates a new one, the more explicit and salient one is chosen as the tag for the FS.
- **Precedence to task-oriented labels:** When an FS could be labeled both with a task-oriented tag such as respond.yes.accept (accepting a proposal made by the assistant, thus advancing the task at hand) and a non-task-oriented one such as social.interpersonal.feedback, the task-oriented tag is chosen, e.g., „Klingt super“ following a request.propose.offer act by the assistant would be labeled with respond.yes.accept. For the same reason, assert acts are not very typical, as in a task-oriented setting most utterances either respond to (Respond) or create (Request) a conversational expectation.

High-level tags	Medium-level tags	Low-level tags	Examples	Often preceded by	Often followed by
<p>Request</p> <p>Set of acts that are intended to elicit some reaction from the listener. Information requests usually take the form of a question (directly or indirectly formulated). Action requests are formulated as instructions to accomplish a task. Proposals are also requests since they are intended to elicit acceptance (or rejection).</p>	instruct	<p>task: the user instructs the assistant to perform a task (in our case, typically, find or book a restaurant), typically at the beginning of the dialogue, but sometimes also mid-dialogue by specifying (further) search or booking information that the assistant has not asked for by means of a request.query.x act.</p>	<p>example: „Finde mir ein Restaurant in der Nähe, das Sushi anbietet.“</p> <p>example: instructing the assistant by specifying booking information</p> <ul style="list-style-type: none"> Assistant: „<u>Möchten Sie, dass ich dort einen Tisch buche?</u>“ (request.propose.suggest) User: „Ja <u>für morgen Abend 19.00 für zwei.</u>“ (respond.yes.accept; request.instruct.task) <p>counterexample: replying to a query by the assistant by specifying booking information</p> <ul style="list-style-type: none"> Assistant: „<u>Wann darf ich den Tisch für Sie buchen?</u>“ (request.query.open) User: „<u>Buche den Tisch auf 19 Uhr.</u>“ (respond.reply.open) 		<p>request. propose. offer</p> <p>request. propose. suggest</p>
		<p>cancel: the user instructs the assistant to cancel the current task.</p>	<p>„Ich will doch nicht buchen.“</p>		

High-level tags	Medium-level tags	Low-level tags	Examples	Often preceded by	Often followed by
	query	open: assistant or user asks the other one an open question, requesting one or more pieces of information. Can be formulated as imperatives or using implicatures – if the FS can be rephrased into an open question on the surface, it should be labelled with this tag. Especially imperative cases may look like request.instruct.task acts, but should be labelled with request.query.open if they request only information, and not performing an action.	<p>„Welche Küche bevorzugen Sie?“ (standard surface form)</p> <p>„Gib mir auch die Adresse und die Telefonnummer!“ (imperative, requesting two pieces of information)</p> <p>„Kannst du mir die Telefonnummer geben?“ (implicature)</p>		respond. reply. open
		yes-no: assistant or user asks the other one a yes-no-question.	„Hat das Restaurant einen Garten?“		respond. reply. agree respond. reply. disagree
		select: the assistant presents the user with a choice of two or more options.	„Wollen Sie italienische oder deutsche Küche?“		respond. reply. select

High-level tags	Medium-level tags	Low-level tags	Examples	Often preceded by	Often followed by
	propose	offer: the assistant offers one or more options (venues or also performing an action like booking a venue) in response to an <u>explicit</u> instruction by the user.	<ul style="list-style-type: none"> User: “<u>Finde ein französisches Restaurant in der Nordstadt</u>“ (request.instruct.task) Assistant: „<u>Ich habe das Restaurant Paris gefunden, welches Ihrem Wunsch entspricht.</u>“ (request.propose.offer) 	request. instruct. task	respond. yes. accept respond. no.reject
		suggest: the assistant suggests one or more options (venues or also performing an action like searching with different criteria) either in response to an <u>implicit</u> instruction by the user or without being instructed to do so, that is, in a <u>pro-active</u> way. This tag also encompasses cases where the assistant pro-actively tries to advance the task, e.g., by requesting further search criteria, loosening search criteria, or proposing to list other options.	<p>example 1: suggest following an implicit user instruction</p> <ul style="list-style-type: none"> User: „<u>Ich habe Hunger.</u>“ (request.instruct.task; note that for this label it does not matter whether the instruction is phrased explicitly or implicitly) Assistant: „<u>In Ihrer Nähe kann ich die Trattoria Italia empfehlen.</u>“ (request.propose.suggest) <p>example 2: suggest without being instructed to do so by the user, i.e., the assistant is pro-active</p>	request. instruct. task	respond. yes. accept respond. no.reject

High-level tags	Medium-level tags	Low-level tags	Examples	Often preceded by	Often followed by
			<ul style="list-style-type: none"> User: „<u>Finde ein billiges italienisches Restaurant in der Innenstadt.</u>“ (request.instruct.task) Assistant: „<u>Es gibt keine günstigen italienischen Restaurants in der Innenstadt. Wollen Sie außerhalb der Innenstadt suchen?</u>“ (repond.notify.failure; request.propose.suggest) 		
	check	confirm: assistant or user checks with the other one whether a certain piece of information is correct.	„Die Trattoria dolce vita liegt in der Nordstadt, richtig?“		respond. yes.agree respond. no.disagree
Respond Respond acts are complementary to request acts , which they usually follow. Respond acts are answers to	reply	open: assistant or user replies to an open question (request.query.open).	<ul style="list-style-type: none"> Assistant: „<u>Wann darf ich den Tisch für Sie buchen?</u>“ (request.query.open) User: „<u>Für 16 Uhr bitte</u>“ (repond.reply.open) 	request. query. open	

High-level tags	Medium-level tags	Low-level tags	Examples	Often preceded by	Often followed by
information requests and also reactions to an action request or a proposal.		select: the user replies to a request.query.select act by selecting one of the options.	<ul style="list-style-type: none"> Assistant: „<u>Wollen Sie lieber in der Nordstadt oder im Süden suchen?</u>“ (request.query.select) User: „<u>Im Norden</u>“ (respond.reply.select) 	request. query. select	
	yes	agree: assistant or user says „yes“ to a yes-no-question posed by the other one, („agree“ may not be the best name for this tag, though).	<ul style="list-style-type: none"> User: „<u>Gibt es Döner?</u>“ (request.query.yes-no) Assistant: „<u>Ja, das steht auf dem Speiseplan</u>“ (respond.yes.agree) 	request. query.yes -no	
		accept: the user accepts a proposal (offer or suggest) made by the assistant.	<ul style="list-style-type: none"> Assistant: „<u>Darf ich Ihnen stattdessen ein Restaurant in der Innenstadt vorschlagen?</u>“ (request.proposal.suggest) User: „<u>Das klingt toll</u>“ (respond.yes.accept) 	request. propose. offer request. propose. suggest	request. instruct. task also by the user, ac- cepting the offer, instruct- ing to go ahead and book)

High-level tags	Medium-level tags	Low-level tags	Examples	Often preceded by	Often followed by
	no	disagree: the user says „no“ to a yes-no-question posed by the assistant („disagree“ may not be the best name for this tag, though).	<ul style="list-style-type: none"> User: „<u>Hat diese Pizzeria einen Garten?</u>“ (request.query.yes-no) Assistant: „<u>Nein, leider nicht</u>“ (respond.yes.disagree) 	request.query.yes-no	
		reject: the user rejects a proposal (offer or suggest) made by the assistant.	<ul style="list-style-type: none"> Assistant: „<u>Ich empfehle Ihnen die Salatbar am Tor.</u>“ (request.proposal.suggest) User: „<u>Nein, da will ich nicht hin!</u>“ (respond.yes.reject) 	request.propose.offer request.propose.suggest	
	notify	success: the assistant notifies the user of the success of a task like a completed search or booking. Note that if the assistant mentions specific venues when returning search results, the utterance is more likely to be a request.propose.offer/suggest act.	<p>example 1: „Ich habe fünf Restaurants für Sie gefunden“</p> <p>example 2: „Ich habe die Buchung für Sie vorgenommen“</p> <p>counterexample:</p> <ul style="list-style-type: none"> User: „<u>Ich brauche ein kinderfreundliches Restaurant um die Ecke.</u>“ (request.instruct.task) Assistant: „<u>Die Pizzeria Dolce Vita und der Ratskeller</u> 		

High-level tags	Medium-level tags	Low-level tags	Examples	Often preceded by	Often followed by
			<u>erfüllen Ihre Kriterien</u> “ (request.propose.offer)		
		failure: the assistant notifies the user of the failure of a task like a search or booking, or also when the assistant cannot answer a user’s query, e.g., because the requested piece of information is not stored in the assistant’s database. Failed search also means no matching results.	<p>example 1: „Über diese Information verfüge ich leider nicht.“</p> <p>example 2: failure due to too many search results</p> <ul style="list-style-type: none"> Assistant: „<u>Ich konnte kein genaues Restaurant finden.</u> <u>Wollen Sie mehr Kriterien angeben, um ein passendes zu finden oder wollen Sie sich alle Optionen anhören?</u>“ (respond.notify.failure; request.query.select). Note that here the search did yield results (but too many) and it is the way the assistant communicates this – „kein genaues“ (<i>no matching</i>) – that leads to the failure tag. 		
		buy_time: assistant or user stalls the actual utterance. Not	„Hm“		

High-level tags	Medium-level tags	Low-level tags	Examples	Often preceded by	Often followed by
		very typical in our case due to the written modality (but sometimes stalling is mimicked).			
Assert Assert acts cover the transmission of information that is not requested in previous discourse, unlike respond acts, nor expect the listener to provide any sort of reaction . These are usually expressions of opinion or statements setting the ground for further interaction.	provide	statement : the user provides an unsolicited statement where the assistant is not expected to react to in any way.	<ul style="list-style-type: none"> Assistant: „<u>Zwei Restaurants entsprechen Ihren Vorstellungen.</u> <u>Würden Sie ein Restaurant mit Live Musik bevorzugen?</u>“ (respond.notify.success; request.query.yes-no) User: „<u>Nein,</u> <u>ich bevorzuge ein Restaurant mit klassischer Musik.</u>“ (respond.no.disagree; assert.provide.statement) 		
Social Social acts have purely social intent and are usually expressed in natural conversation to conform to social expectations. These include greetings, politeness expressions and expressions of sympathy and agreement.	greetings	opening : addressing of the interlocutor to open a conversation. Wake words like „Computer“ or „Sprachassistent are not included and tagged with „other“.	„Hallo“		
		closing : addressing of the interlocutor in closing the conversation.	„Tschüss“		

High-level tags	Medium-level tags	Low-level tags	Examples	Often preceded by	Often followed by
These expressions are often formulaic and they can be omitted from the conversation without affecting its structure or comprehension.	politeness	apology: assistant or user apologizes for something; often assistant turns communicating failure (e.g., when there is no search result) also include „leider“ which expresses an apology, but this is not labelled since respond.notify.failure is more salient, FS need to be continuous sequences and there can only be one label per FS.	„ <u>Entschuldigung</u> , <u>diese Information steht mir leider nicht zur Verfügung</u> “ (social.politeness.apology; respond.notify.failure)		
		thanks: assistant or user expresses their gratitude.	„Vielen Dank“		
		ack_thanks: assistant or user acknowledges the other one's gratitude.	„Bittesehr“		
	interpersonal	feedback: assistant or user expresses feedback or a phatic reaction concerning the other one's utterance, both content-wise and for signaling reception and understanding of the previous utterance.	„Das ist super“, „Okay“		
Other					

High-level tags	Medium-level tags	Low-level tags	Examples	Often preceded by	Often followed by
Currently used for anything that the annotator deems not covered by the current tag set. FS tagged with other can be revised later to refine the tag set. Wakewords are currently tagged with other.					

Reference: Pareti, Silvia, and Tatiana Lando. 2019. "Dialog Intent Structure: A Hierarchical Schema of Linked Dialog Acts." In LREC 2018 - 11th International Conference on Language Resources and Evaluation, 2907–14. <https://www.aclweb.org/anthology/L18-1460/>.