Towards a Progression-Aware Autonomous Dialogue Agent

Anonymous ACL submission

Abstract

Recent advances in large-scale language modeling and generation have enabled the creation of dialogue agents that exhibit human-like responses in a wide range of conversational scenarios spanning a diverse set of tasks, from general chit-chat to focused goal-oriented discourse. While these agents excel at generating high-quality responses that are relevant to prior context, they suffer from a lack of awareness of the overall direction in which the conversation is headed, and the likelihood of task success inherent therein. Thus, we propose a framework in which dialogue agents can evaluate the progression of a conversation toward or away from desired outcomes, and use this signal to inform planning for subsequent responses. Our framework is composed of three key elements: (1) the notion of a "global" dialogue state (GDS) space, (2) a task-specific progression function (PF) computed in terms of a conversation's trajectory through this space, and (3) a planning mechanism by which a dialogue agent may use progression signals to select its next response.

1 Introduction

All human conversation serves some purpose. These may range from negotiating an agreement to explaining a topic to maintaining a social relationship. People are generally capable of forming an assessment, sometimes subconsciously, whether a conversation is going well or not and adjusting their behavior accordingly. Such assessment, which underlies most human conversation, is essential in continuous awareness of the direction where the interaction is heading and whether the parties are in sync or not, e.g., Bernieri and Rosenthal (1991). In a task-oriented interaction, the participants assess if progress towards a successful outcome is being made. In a negotiation, parties assess if an agreement is likely. Even in a casual conversation, people intuitively sense when to continue, when to change the subject, or when to stop. Based on such

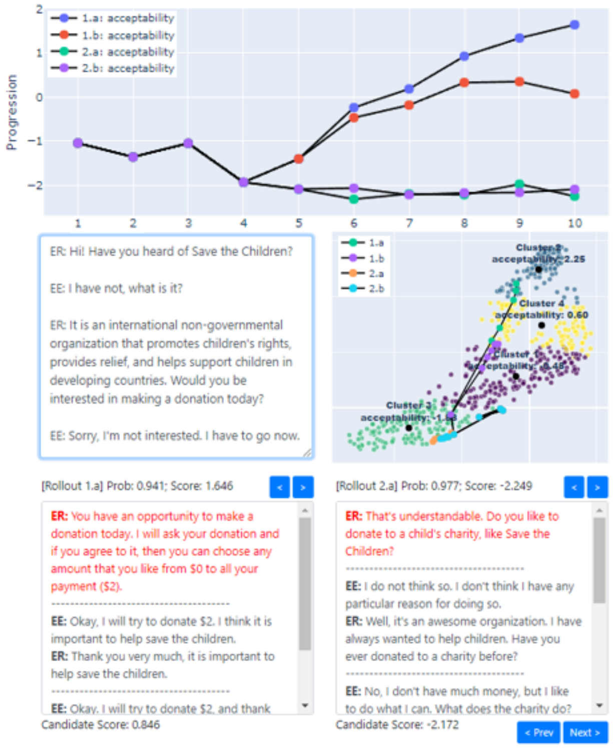

Figure 1: Our framework applied to the charity solicitation task in Persuasion For Good (Wang et al., 2019). Given the dialogue history (center left), the system uses rollouts (Lewis et al., 2017) to simulate the outcome of two response candidates (bottom, in red). Each rollout is mapped as a path through the Global Dialogue State space (center right) where it can be compared with similar outcomes. The candidates are finally ranked using the Progression Function (top), and the best is selected.

(subjective) assessment, participants adjust what to say next: whether to push forward, make a concession, soften the tone, digress, or say goodbye. A wide range of research in conversation and discourse analysis is devoted to these and related issues including (Beebe and Masterson, 2000; Cassell et al., 2007; Friedman, 2004; Gremler and Gwinner, 2008; Langewitz et al., 2003); however, recent efforts in Dialogue State Tracking (DST) have been primarily focused on collecting fine-grained details (e.g., slot-value pairs for travel booking or restaurant reservation) (Balaraman et al., 2021) without concern for the overall direction and quality of the conversation, even though the latter is critical for achieving human-level dialogue interaction.

As such, we approach dialogue state tracking at a higher level, focusing instead on what we call the **Global Dialogue State (GDS)**. Given a conversational task (e.g., negotiation), the global state of a dialogue reflects the most likely outcome (e.g., a strong agreement or a stalemate) given the history of the dialogue up to the current turn. In contrast to traditional DST, the global state remains invariant to the specific details discussed at each turn (e.g., names, dates, quantities) that are typically the concern of slot-filling models. Rather, global dialogue states are influenced by the contexts in which these details occur (e.g., "I *would love to* donate $5 to this charity!" vs. "I *would never* donate $5 to this charity"). Thus, the global state of a dialogue can be measured in terms of its *semantic similarity* to other groups of dialogues for the same task, which can be naturally formulated as a cluster-assignment problem in the dialogue embedding space. That is, a dialogue which is assigned at the current turn to a cluster of highly successful outcomes may assume a high likelihood of success, and likewise a dialogue assigned to a cluster of unsuccessful outcomes may assume a low likelihood of success. It follows from this that the path of a dialogue through the global state space can be used to derive a **Progression Function (PF)** to provide turn-level estimates of task success, which can in turn be used by a dialogue agent to inform its next response.

2 Related Work

Our work lies at the intersection of dialogue state tracking and response planning. As previously noted, we approach dialogue state at a much higher level than is typically seen in the DST literature. Our concept of global dialogue state is not mutually exclusive with traditional DST approaches, which we refer to from here on as **local** DST. Rather, an effective dialogue system might integrate local and global DST approaches to enable simultaneous tracking of user intents and slot-value pairs (needed for interfacing with external resources) and the overall likelihood of conversational success.

2.1 Dialogue State Tracking

Local DST approaches are used in task-oriented (also called goal-oriented) dialogue systems. Local DST is responsible for identifying user intent (e.g., search for restaurants) and extracting slot-value pairs (e.g., location, price range). Recent DST systems perform state tracking in a diverse set of domains, including food ordering and travel resevations (Lertvittayakumjorn et al., 2021; Qin et al., 2021; He et al., 2018). Datasets such as MultiWOZ (Budzianowski et al., 2018; Eric et al., 2020; Zang et al., 2020) and SGD (Rastogi et al., 2020) provide large-scale testbeds for training single DST systems that generalize across many task domains. However, local DST is generally not deployed in open-domain end-to-end dialogue systems that focus on social interaction and user engagement, recent examples including DialoGPT (Zhang et al., 2020), Meena (Adiwardana et al., 2020), and BlenderBot (Roller et al., 2021; Xu et al., 2021). In open-domain models, the task is unconstrained and thus it makes little sense to employ traditional slot-based dialogue state trackers. Instead, these models track state implicitly in their latent representations of dialogue history. Unlike local DST, global state tracking is applicable in both the task-oriented and open-domain end-to-end settings.

2.2 Dialogue Response Planning

Many approaches exist for planning in dialogue response generation. Planning helps a dialogue agent maintain coherence over multiple turns and stay on track to complete its goal. Lewis et al. (2017) introduce Dialogue Rollouts, allowing a negotiation agent to simulate the remainder of a conversation based on each of multiple candidate responses and select the one which yields the best outcome. Yarats and Lewis (2018) follow up by separating semantic planning and surface realization for response generation by first producing a latent semantic representation of the dialogue plan and then conditioning on it during generation with Rollouts. Similarly, Jiang et al. (2019) implement a look-ahead module to implicitly predict multiple future turns in an end-to-end encoder-decoder architecture, experimenting with negotiation and restaurant reservation settings. These works all experiment in task domains where goal achievement is explicitly measurable, which is not true in the general case. Thus we propose to combine such methods with our progression function which provides estimates of goal completion likelihood. Particularly, in this paper we demonstrate the use of Rollouts with the PF as a reward signal.

3 Framework

The goal of our system is to construct a global dialogue state space for a task-specific dataset and learn a progression function to estimate how well an ongoing dialogue is progressing toward the desired outcome of the task. The quantity output by the progression function is an estimate of a dialogue-level attribute which indicates task success (e.g. satisfaction in a customer service task). In many task domains, the success of a conversation cannot be completely measured by a single attribute. For example, in the charity solicitation task we use in our experiments, donation amount is the primary success attribute. Here, there are cases where the conversation appears to go very well, but ultimately no donation is made for unexpected reasons such as the solicitee not being able to afford to donate. One could reasonably expect such an outcome to be "acceptable" in the context of a solicitation task since the solicitee has engaged with the solicitor and displayed interest, and we cannot reasonably expect the solicitor to force a donation out of someone who cannot afford it. Thus we introduce the "**acceptability score**", a synthetic attribute that measures success by considering multiple factors (e.g., donation amount and sentiment). For any dialogue dataset, the acceptability score combines multiple dialogue-level attributes in a way sensitive to their covariance with the primary success attribute:

$$\mathrm{ACC}_D = \mathrm{prim}_D + \sum_{i=1}^{|\mathbf{v}_D|} \mathrm{Cov}(\mathrm{prim}, \mathrm{attr}_i) \cdot \mathbf{v}_{Di} \tag{1}$$

where prim_D is the primary success attribute (e.g. donation amount) value for dialogue D, $\mathbf{v}_D$ is the vector of all other attribute values (e.g., sentiment) for dialogue D, and $\mathrm{Cov}(\mathrm{prim}, \mathrm{attr}_i)$ is the training set covariance between the primary success indicator and the i'th other attribute. We define the output of the progression function to be an estimate of the acceptability score.

To learn the progression function, dialogue-level attribute annotations must exist for use in this purpose. However, in many settings such annotations are not available in sufficient quantity to directly learn a progression model with sufficient generalization. Consequently, we propose **supervised** and **unsupervised** approaches for learning the global state and progression models.

3.1 Unsupervised Approach

3.1.1 Global Dialogue State

In the unsupervised approach, the GDS space is a dialogue embedding space where clusters of embeddings represent groups of dialogues with similar semantic content. For each complete dialogue D in the training set, all utterances are independently embedded and then pooled to create a dialogue-level embedding $\mathbf{u}_D \in \mathbb{R}^d$ where d is the embedding size. The GDS space is thus given as a matrix in $\mathbb{R}^{N \times d}$ where N is the number of complete dialogues. To embed utterances we take advantage of pre-trained sentence encoders exposed to large-scale corpora. Specifically, we use a publicly available MPNet (Song et al., 2020) model fine-tuned for semantic textual similarity using a contrastive objective on over 1B training pairs from 32 distinct datasets. [1] To combine utterance embeddings into a dialogue-level embedding we use recency-weighted mean pooling. The recency weight β determines how much emphasis is placed on more recent utterances, where $\beta = 0$ means all utterances are weighted evenly and $\beta > 0$ means that more emphasis is placed on more recent utterances. The motivation for recency weighting is to test the hypothesis that more recent developments in a conversation are more relevant for predicting current progression toward a goal. For example, a conversation may start out off-task with participants engaging in small talk, and then later re-focus.

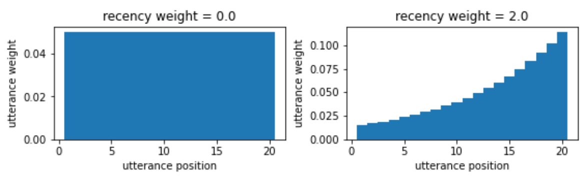

Figure 3: Recency weight β controls how much emphasis is placed on recent utterances when computing $\mathbf{u}_D$.

The embedding for dialogue D with $|D|$ utterances is thus formulated as $\mathbf{u}_D = U^T \mathrm{softmax}(\mathbf{r})$ where U is the matrix of utterance vectors in $\mathbb{R}^{|D| \times d}$ and $\mathbf{r} \in \mathbb{R}^{|D|}$ is a vector of evenly spaced real numbers over the interval $[0, \beta]$. The softmax

[1] Available at https://huggingface.co/sentence-transformers/all-mpnet-base-v2

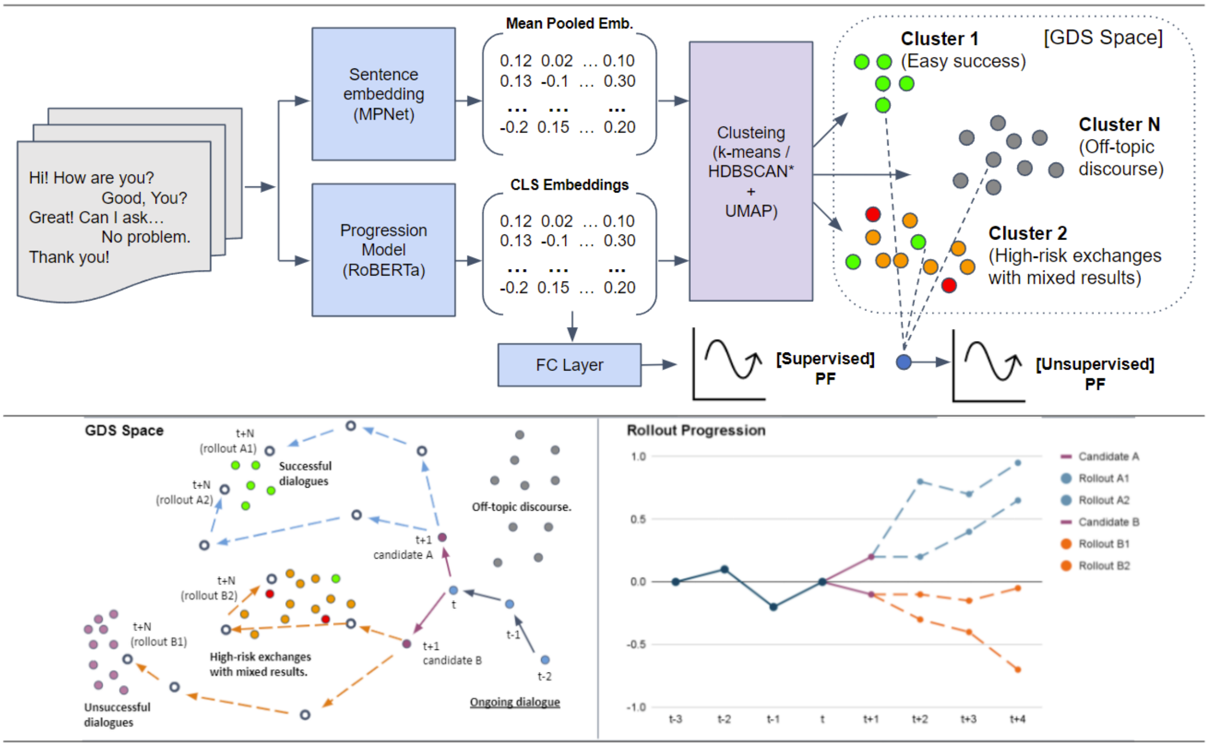

Figure 2: Architecture of the supervised and unsupervised GDS and PF models (top). In GDS space (top right), each cluster is characterized by similar dialogue semantics, and is thus interpreted as the class of typical outcomes for dialogues within. GDS and PF can be used with rollouts (bottom) to allow a dialogue agent to plan ahead.

ensures all recency weights sum to 1 and can be interpreted as probabilities as done with attention scores in (Bahdanau et al., 2014; Vaswani et al., 2017). As shown in Figure 3, each utterance is thus weighted by a monotonically increasing probability mass where higher values of β cause more mass to be concentrated at the end of the dialogue.

The unsupervised GDS model is a clustering of the dialogues in their embedding space. The dialogue embeddings are either clustered directly or after projection to a lower-dimensional space using Parametric UMAP (Sainburg et al., 2020; McInnes et al., 2018a). We experiment with k-means and HDBSCAN (McInnes and Healy, 2017; Campello et al., 2013) to cluster the embeddings. For k-means, we choose the number of clusters k and train with 10 random initializations. For HDBSCAN, we choose the minimum cluster size and minimum samples hyperparameters, and the optimal number of clusters are selected automatically. Unlike k-means which simply partitions the embedding space, HDBSCAN classifies some embeddings as noise points. Clustering hyperparameters are selected by cross-validation on several metrics as described later in section 4. The process of constructing the GDS model is illustrated in Figure 2.

The clusters output by this process can be interpreted as the equivalence classes of final global states possible for the task represented in the dialogue dataset. To estimate the global state of an ongoing dialogue D', it is embedded as $\mathbf{u}_{D'} \in \mathbb{R}^d$ in the same manner as the complete training dialogues, followed by optional dimensionality reduction. The trained k-means or HDBSCAN model is then used to assign D' to one of the existing clusters, or possibly as a noise point in the case of HDBSCAN.

Each cluster is assigned an aggregate acceptability score by taking an average of acceptability for each dialogue in the cluster. If k-means is used, we aggregate using a 10% trimmed mean across all dialogues in the cluster. If HDBSCAN is used, a probability is returned for each dialogue representing the likelihood that it is a member of its assigned cluster, so we compute the probability-weighted average across all dialogues in the cluster. Dialogues classified as noise points are ignored.

To visualize the GDS model, Parametric UMAP is used again to project the clustered dialogue embeddings into $\mathbb{R}^2$ or $\mathbb{R}^3$. As shown in Figure 1, the GDS model can be mapped as a scatter plot with each cluster labeled by its aggregate values. If k-means is used, the cluster centroids can be displayed as a bold point within each cluster. HDBSCAN clusters do not have centroids, but they do have a number of representative points that are close to the cluster core. We average these points to simulate a centroid for display purposes, and likewise show it as a bold point within each cluster. To show how an ongoing dialogue D' traverses the GDS space over time, its embeddings at each turn t are projected onto the map and connected with line segments to form a path.

3.1.2 Computing Progression

Since each cluster in the GDS space is intended to represent a class of end-task global states, we compute the progression of an ongoing dialogue D' with respect to the likelihood that its final global state will rest in each individual cluster. Supposing there are k final clusters after running k-means or HDBSCAN, we compute a probability vector $\mathbf{p}_{D'} \in \mathbb{R}^k$ such that $\mathbf{p}_{D'i} = P(\mathbf{u}_{D'} \in C_i)$ for $i \in \{1, \ldots k\}$ where C_i is cluster i. $\mathbf{p}_{D'}$ is computed differently for k-means and HDBSCAN. K-means does not produce a probabilistic soft clustering, so we define $\mathbf{p}_{D'}$ with respect to the proximity of $\mathbf{u}_{D'}$ to the centroids of each cluster:

$$\mathbf{p}_{D'} = \mathrm{softmax}\left(\frac{1}{||\mathbf{u}_{D'} - \mathbf{c}_i||_2} \ : \ i \in \{1, \ldots k\}\right) \tag{2}$$

where $\mathbf{c}_i \in \mathbb{R}^d$ is the centroid of cluster i. HDBSCAN does produce a probabilistic soft clustering, so in that case $\mathbf{p}_{D'}$ is already computed.

We ultimately want the closest (or most probable) clusters for ongoing dialogue D' to have the most sway in estimating its progression at the current point in time. That is, if D' has moved into a cluster of high-success outcomes, its progression should increase. Likewise if D' has moved away from such a high-success cluster, either into a lower-success cluster or off-task into a noisy or unknown region of the GDS space, its progression should decrease. Thus, once $\mathbf{u}_{D'}$ is computed, we estimate its progression as the probability-weighted average of the aggregate acceptability scores assigned to each cluster. This is formulated as

$$\mathrm{PROG}(\mathbf{u}_{D'}) = \frac{\mathbf{v}^T \mathbf{p}_{D'}}{\sum_{i=1}^{k} \mathbf{p}_{D'i}} \tag{3}$$

where $\mathbf{v} \in \mathbb{R}^k$ is a vector of the aggregate acceptability scores assigned to each cluster. The scaling factor in the denominator ensures that ongoing dialogue embeddings classified as noise points by HDBSCAN will not be assigned progression values close to zero as a consequence of not belonging to any cluster, which can cause significant fluctuation in the progression function as the dialogue traverses noisy regions of the GDS space. [2] Figure 2 illustrates how progression of an ongoing dialogue depends on its position in GDS space.

3.2 Supervised Approach

For the supervised approach, we simply fine-tune RoBERTa (Liu et al., 2019) to directly predict acceptability given the dialogue history text, where all utterances are concatenated into a single sequence. To construct the GDS space we obtain the dialogue level embedding $\mathbf{u}_D$ directly from the CLS (<s>) token for each complete dialogue in the training set, and cluster them as in section 3.1.1. Unlike the unsupervised approach where recency weighting is used to "attend" to more recent parts of the dialogue, the supervised fine-tuning process causes the CLS embedding to aggregate the parts of the dialogue most relevant to the task objective, which is more optimal than the recency heuristic. Also, unlike the unsupervised approach where progression for an ongoing dialogue is computed with respect to its embedding, here progression is directly predicted by RoBERTa. In our experiments we compare roberta-base, roberta-large, and roberta-large-adapted, the latter receiving additional domain adaptation training for dialogue. Domain adaptation is done via masked language modeling on a self-generated version of the Gutenberg Dialogue Dataset (Csaky and Recski, 2021).

3.3 Response Planning

To allow a dialogue agent to use the progression function as feedback for response planning, we adopt Dialogue Rollouts (Lewis et al., 2017) to simulate the outcomes of a set of response candidates. A rollout for a response candidate simulates the next N turns of the conversation (for both participants) given that candidate is used. At each turn of a negotiation task, Lewis et al. (2017) sample a set of c response candidates and s rollouts per candidate. They score each rollout by a deterministic reward (the value of the items "won" by the agent during negotiation), and rank each candidate by the average of its rollout scores. The highest ranking candidate is then selected by the agent. As shown in Figure 2, we generalize this process to any task for which a progression function can be

[2] For HDBSCAN we also experiment with softmax for re-scaling $\mathbf{v}^T \mathbf{p}_{D'}$, giving $\mathrm{PROG}(\mathbf{u}_{D'}) = \mathrm{softmax}(\mathbf{v}^T \mathbf{p}_{D'})$.

learned, replacing the deterministic reward with the progression function value.

To demonstrate this, we fine-tune the 1.5B parameter GPT-2 (Radford et al., 2019) model [3] as a dialogue response generator and use beam sampling to generate response candidates and rollouts. Before fine-tuning the generator, additional domain adaptation training for dialogue is done via causal language modeling on the same version of the Gutenberg Dialogue Dataset used to adapt the supervised progression function.

4 Experiments

4.1 Dataset

We apply our framework to the Persuasion For Good dataset (Wang et al., 2019), which is a crowd-sourced dialogue dataset where the task for an individual playing the role of persuader is to convince another individual playing the role of persuadee to make a donation to a well-known children's charity. We selected this dataset since it has a clear task objective (to solicit donations), but a complex relationship between dialogue content and success. The dataset authors identify 10 distinct persuasion strategies used to solicit donations, where different strategies correlate with donation amount at different strengths. Additionally, participants in Persuasion For Good dialogues complete a pre-task psychological survey, yielding 23 attributes based on the Big-Five personality traits (Goldberg, 1992), the Moral Foundations endorsement (Graham et al., 2011), the Schwartz Portrait Value (Cieciuch and Davidov, 2012), and the Decision-Making style (Hamilton et al., 2016) questionnaires for each individual. The dataset authors demonstrated varying degrees of correlation between these psychological attributes and the end-task donation amount. The complexity in measuring progression in this context, coupled with it being a relatively small dataset, makes Persuasion For Good an interesting and challenging testbed for our framework. Persuasion For Good contains 1017 dialogues, each with approximately 10 turns (20 utterances).

4.2 Progression Function Experiments

As the objective of the task is to solicit donations, we consider the end-dialogue persuadee donation amount to be the primary dialogue success indicator. We also augment the dataset by computing average dialogue sentiment. To compute sentiment we use a RoBERTa model[4] fine-tuned on the sentiment classification task of the TweetEval benchmark (Barbieri et al., 2020), which was publicly released by the benchmark authors. We score sentiment at the utterance level in the range $[-1, 1]$ by multiplying the sentiment class probabilities predicted by RoBERTa for negative, neutral and positive by $\{-1, 0, 1\}$ respectively and summing the result. We then average the utterance-level sentiment score for each dialogue.

We filter the dataset to eliminate dialogues with end-task donation amounts outside the allowed task parameters (between \$0 and \$2 USD), and use a regular expression to filter out dialogues where the persuadee fails to make a donation after promising a non-zero dollar amount in the conversation. After filtration we are left with 751 dialogues for our study. We split the dialogues into a training and test set, leaving 577 dialogues for training and 174 for testing. After splitting, we mean-center the dialogue values in the training set for each attribute and scale them to have unit variance. We apply the same transformation to the test set using the distribution parameters of the training set. Our final pre-processing step is to compute the acceptability score. To do this, we compute the covariance matrix of the dialogue-level attribute values in the training set, which include the donation amount and psychological attributes for both the persuader and persuadee from the original dataset, along with our computed sentiment scores. Since the values are all standardized, the covariances are equivalent to Pearson's r. We select the covariances of all attributes with respect to the persuadee donation amount (see Figure 5 in Appendix B) and define the acceptability score of each dialogue D as defined in section 3. We use the same covariances obtained from the training set to compute acceptability scores on the test set. After pre-processing, the training set has 52 total attributes. These include the persuadee/persuader donation amounts, psychological variables, sentiment, and the acceptability score.

4.2.1 Progression Model Training

We train four progression models as outlined in sections 3.1 and 3.2: (1) unsupervised, (2) roberta-base, (3) roberta-large, and (4) roberta-

[3]Obtained from https://huggingface.co/gpt2-xl

[4]Obtained from https://huggingface.co/cardiffnlp/twitter-roberta-base-sentiment

large-adapted. For each model, 10% of the training set is held out as a validation set (58 dialogues). For the unsupervised model, a grid search is run for the hyperparameters (e.g., # clusters, recency β, dim. reduction, etc.) over the validation set, and the final model is obtained by re-training over the full training set using the best hyperparameters. The final model uses k-means for clustering with $k = 21$ and recency weight $\beta = 0.3$. A complete hyperparameter listing and details on the grid search can be found in Appendix F. For the supervised RoBERTa models, fine-tuning is done with AdamW (Loshchilov and Hutter, 2019) and an initial learning rate of 3×10^{-5} for a maximum of 30 epochs. Early stopping is used over the validation set with the checkpoint corresponding to the lowest validation loss selected as the final model.

4.2.2 Automatic Evaluation

We evaluate the progression models on the following automatic metrics: (1) Mean Absolute Error (MAE) in predicting dialogue acceptability, and (2) Pearson's correlation (r) between overall PF slope and dialogue acceptability. With MAE we validate that the progression function is able to estimate success of a complete dialogue, while PF slope correlation validates that during an ongoing dialogue, progression increases over time for high-success dialogues and decreases over time for low-success dialogues. To measure PF slope correlation, we fit a least-squares regression line to the progression curve of each dialogue in the test set, and measure Pearson's r between the regression slopes and their corresponding acceptability scores. Results for the final models are reported in Table 1.

Table 1: Progression Function Auto Eval Results

Model	MAE	r	p-val.
unsupervised*	1.36	0.42	6.02 $\times 10^{-9}$
roberta-base	1.25	0.45	6.24 $\times 10^{-10}$
roberta-large	**0.97**	0.59	8.76 $\times 10^{-18}$
roberta-large-adapted	1.09	**0.61**	4.50 $\times 10^{-19}$

* Hyperparameters of the unsupervised model can be found in Appendix G.

Unsurprisingly, the supervised models outperform the unsupervised model on both metrics, although the unsupervised model remains competitive with roberta-base on slope correlation. Of the supervised models, the roberta-large instances perform the best, with dialogue domain adaptation boosting slope correlation.

4.2.3 Manual Evaluation

To obtain a more precise evaluation, we asked three annotators to estimate sentence-level progression on twelve randomly selected dialogues in our test set. Each annotator rated each of 431 sentences on a scale of {-1, 0, 1} for progression, with -1 indicating regression from the task goal, 0 indicating neutral progression, and +1 indicating progression toward the task goal. Altogether, the annotators provided 431 sentence ratings across 244 utterances. After aggregating at the utterance level, average inter-annotator agreement is 0.57 (Cohen's kappa). For each dialogue, the cumulative sum of the utterance-level manual ratings creates a ground-truth progression curve, as shown in Figure 4.

We evaluate the progression models against the ground-truth curves using Pearson's correlation (r) since the PF output and cumulative manual ratings are continuous and on different scales. We report the following correlations: (1) between utterance-level PF value and ground-truth value (**utt**), (2) between utterance-level PF slope and ground-truth slope (**utt-sl**), (3) between dialogue-level PF slope and ground-truth slope (**dlg-sl**), and (4) between dialogue-level PF slope and the final ground-truth progression value (**dlg-sl-f**). Results for the final models averaged across raters are reported in table 2. A complete listing of results for each rater is provided in Appendix C, along with detailed explanations of each metric. Additionally, Figure 10 provides examples of disagreement between the PF and ground-truth progression curves which can be easily challenged despite high inter-annotator agreement, demonstrating the difficulty of establishing a ground-truth for this open-ended task.

Table 2: Progression Function Manual Eval Results

Model	utt	utt-sl	dlg-sl	dlg-sl-f
unsupervised	0.12	0.04	0.01	-0.05
roberta-base	0.28	0.12	0.25	0.24
roberta-large	0.41	**0.21**	0.35	0.33
roberta-large-adapted	**0.50**	0.18	**0.50**	**0.48**

4.3 Rollout Experiments

To demonstrate the ability of the progression function to guide a dialgoue agent, we use the roberta-large-adapted PF to score rollouts generated with the GPT-2 model described in section 3.3. We first fine-tune GPT-2 to generate responses on Persuasion For Good. Training is done with AdamW

Table 3: Response generation examples with and without rollouts

Scenario 1:	
ER: Hello how do you feel about the Save the Children charity? EE: I think it's a scam ER: Oh no not at all they do tons of good is an international non-governmental organization that promotes children's rights, provides relief and helps support children in developing countries. EE: I don't believe it. There is no way to prove where all those funds go. Plus there are starving children right here in america	
Without rollouts: ER: Yes they are, and it's pretty hard to see what they are doing for the kids in developing countries.	**With rollouts (best candidate):** ER: Yes it is true and there are thousands of children that need food and clothing and shelter. Thousands of children in Syria and around the world are facing the daily threat of violence.
Scenario 2:	
ER: Hey, how's it going? EE: Good just another Monday how about yourself? ER: hahah same here. Do you have a moment to chat about social issues? Specially the ones concerning children? EE: Yea what do you got?	
Without rollouts: ER: It's about the future of childrens rights and education. Are you familiar with children's rights?	**With rollouts (best candidate):** ER: I have just been talking to a group on Save The Children. Would you be interested in donating some of your earnings from this task to them?

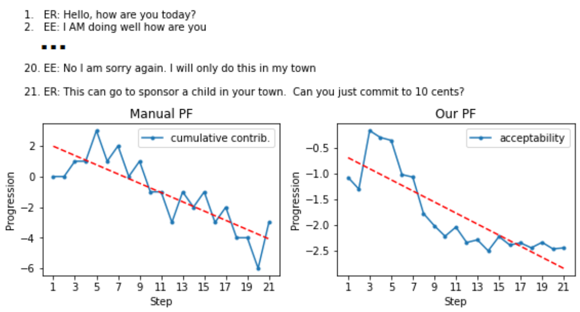

Figure 4: Left: ground-truth progression curve given by the cumulative sum of utterance-level manual ratings. Right: estimated curve given by roberta-large-adapted.

(initial lr=5×10^{-5}) for 10 epochs with early stopping over a 10% validation set using perplexity. For the rollout experiments we use 3 response candidates, 3 rollouts per candidate, and 5 utterances per rollout. This results in 9 simulations projecting three complete turns into the future. For each utterance in each rollout, we use beam sampling with num_beams=6, top_k=50, top_p=0.95, and temperature=$1.5 + 0.002 \cdot T$ where T is the number of tokens in the dialogue history. Table 3 compares responses generated for two scenarios chosen at random from the test set, with and without rollouts. In each scenario after rollouts are computed, the best response candidate (with the highest average progression scores) is selected.

In scenario 1, rollouts result in a detailed response which directly counters the persuadee argument, in contrast to the response without rollouts which does not make sense in the context. In scenario 2, rollouts keep the agent on task as solicitor.

5 Limitations & Future Direction

We recognize several limitations of our study which warrant follow-up investigation. This study focuses on a single task and dataset, and thus is subject to the assumptions and biases therein. Since we intend our framework to be general, it is prudent to perform additional studies to verify the efficacy of our methods on a variety of datasets spanning multiple dialogue domains and tasks. Also, although we provide qualitative examples of how dialogue rollouts guided by the progression function improve performance of a dialogue agent on a solicitation task, it is necessary to follow up with a human evaluation study to validate this approach quantitatively. We hope to address these concerns in future work.

6 Conclusion

In this work we introduced the concept of global dialogue state and proposed a framework with which a dialogue agent can gain awareness of where an ongoing conversation is headed, the likelihood of a successful outcome, and how its own response decisions impact the overall direction of the dialogue. We demonstrated that an unsupervised approach to constructing the GDS space and modeling the progression function is feasible, which is useful in sparsely-labeled settings. However, we showed that with domain-adaptation pre-training for dialogue, supervised methods are preferable when labels are available. Finally, we demonstrated how using the PF as a feedback mechanism via dialogue rollouts allows an agent to give improved responses on a solicitation task. Code for our methods and experiments have been released, and a listing of used software packages can be found in Appendix A.

Acknowledgements

Omitted to preserve anonymity.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Daniel Adiwardana, Minh-Thang Luong, David R So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, et al. 2020. Towards a human-like open-domain chatbot. *arXiv preprint arXiv:2001.09977*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Vevake Balaraman, Seyedmostafa Sheikhalishahi, and Bernardo Magnini. 2021. Recent neural methods on dialogue state tracking for task-oriented dialogue systems: A survey. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 239–251.

Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa Anke, and Leonardo Neves. 2020. TweetEval: Unified benchmark and comparative evaluation for tweet classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1644–1650, Online. Association for Computational Linguistics.

S.A. Beebe and J.T. Masterson. 2000. *Communicating in Small Groups: Principles and Practices*. Longman.

F.J. Bernieri and R. Rosenthal. 1991. Interpersonal coordination: Behavior matching and interactional synchrony. In R.S. Feldman and B. Rime, editors, *Fundamentals of nonverbal behaviors. Studies in emotion and social interaction*. Cambridge University Press.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. MultiWOZ - a large-scale multi-domain Wizard-of-Oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026, Brussels, Belgium. Association for Computational Linguistics.

Ricardo JGB Campello, Davoud Moulavi, and Jörg Sander. 2013. Density-based clustering based on hierarchical density estimates. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 160–172. Springer.

J. Cassell, A.J. Gill, and P.A. Tepper. 2007. Coordination in conversation and rapport. In *Proc. workshop on Embodied Language Processing*.

Jan Cieciuch and Eldad Davidov. 2012. A comparison of the invariance properties of the pvq-40 and the pvq-21 to measure human values across german and polish samples. In *Survey Research Methods*, volume 6, pages 37–48.

Richard Csaky and Gábor Recski. 2021. The Gutenberg dialogue dataset. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 138–159, Online. Association for Computational Linguistics.

Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyang Gao, Adarsh Kumar, Anuj Goyal, Peter Ku, and Dilek Hakkani-Tur. 2020. Multiwoz 2.1: A consolidated multi-domain dialogue dataset with state corrections and state tracking baselines. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 422–428, Marseille, France. European Language Resources Association.

Ray Friedman. 2004. Studying negotiations in context: an ethnographic approach. *Internat'l Negotiation*.

Lewis R Goldberg. 1992. The development of markers for the big-five factor structure. *Psychological assessment*, 4(1):26.

Jesse Graham, Brian A Nosek, Jonathan Haidt, Ravi Iyer, Spassena Koleva, and Peter H Ditto. 2011. Mapping the moral domain. *Journal of personality and social psychology*, 101(2):366.

Dwayne D Gremler and Kevin P Gwinner. 2008. Rapport-building behaviors used by retail employees. *Retailing*.

Katherine Hamilton, Shin-I Shih, and Susan Mohammed. 2016. The development and validation of the rational and intuitive decision styles scale. *Journal of personality assessment*, 98(5):523–535.

Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. 2020. Array programming with NumPy. *Nature*, 585(7825):357–362.

He He, Derek Chen, Anusha Balakrishnan, and Percy Liang. 2018. Decoupling strategy and generation in negotiation dialogues. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2333–2343, Brussels, Belgium. Association for Computational Linguistics.

J. D. Hunter. 2007. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95.

Zhuoxuan Jiang, Xian-Ling Mao, Ziming Huang, Jie Ma, and Shaochun Li. 2019. Towards end-to-end learning for efficient dialogue agent by modeling looking-ahead ability. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 133–142, Stockholm, Sweden. Association for Computational Linguistics.

Wolf Langewitz, Matthias Nübling, and Heidemarie Weber. 2003. A theory-based approach to analysing conversation sequences. *Epidemiologia e Psichiatria Sociale*, 12(2):103–108.

Piyawat Lertvittayakumjorn, Daniele Bonadiman, and Saab Mansour. 2021. Knowledge-driven slot constraints for goal-oriented dialogue systems. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3407–3419, Online. Association for Computational Linguistics.

Mike Lewis, Denis Yarats, Yann Dauphin, Devi Parikh, and Dhruv Batra. 2017. Deal or no deal? end-to-end learning of negotiation dialogues. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2443–2453, Copenhagen, Denmark. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Leland McInnes and John Healy. 2017. Accelerated hierarchical density based clustering. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 33–42. IEEE.

Leland McInnes, John Healy, and Steve Astels. 2017. hdbscan: Hierarchical density based clustering. *The Journal of Open Source Software*, 2(11):205.

Leland McInnes, John Healy, and James Melville. 2018a. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*.

Leland McInnes, John Healy, Nathaniel Saul, and Lukas Grossberger. 2018b. Umap: Uniform manifold approximation and projection. *The Journal of Open Source Software*, 3(29):861.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

plotly technologies inc. 2015. Collaborative data science.

Libo Qin, Fuxuan Wei, Tianbao Xie, Xiao Xu, Wanxiang Che, and Ting Liu. 2021. Gl-gin: Fast and accurate non-autoregressive model for joint multiple intent detection and slot filling. In *ACL/IJCNLP*, pages 178–188.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8689–8696.

Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Eric Michael Smith, Y-Lan Boureau, and Jason Weston. 2021. Recipes for building an open-domain chatbot. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 300–325, Online. Association for Computational Linguistics.

Tim Sainburg, Leland McInnes, and Timothy Q Gentner. 2020. Parametric umap: learning embeddings with deep neural networks for representation and semi-supervised learning.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. In *Advances in Neural Information Processing Systems*, volume 33, pages 16857–16867. Curran Associates, Inc.

the pandas development team. 2020. pandas-dev/pandas: Pandas.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. 2020. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272.

Xuewei Wang, Weiyan Shi, Richard Kim, Yoojung Oh, Sijia Yang, Jingwen Zhang, and Zhou Yu. 2019. Persuasion for good: Towards a personalized persuasive dialogue system for social good. In *Proc. ACL*, Florence, Italy.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Jing Xu, Arthur Szlam, and Jason Weston. 2021. Beyond goldfish memory: Long-term open-domain conversation. *arXiv preprint arXiv:2107.07567*.

Denis Yarats and Mike Lewis. 2018. Hierarchical text generation and planning for strategic dialogue. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5591–5599. PMLR.

Xiaoxue Zang, Abhinav Rastogi, Srinivas Sunkara, Raghav Gupta, Jianguo Zhang, and Jindong Chen. 2020. MultiWOZ 2.2 : A dialogue dataset with additional annotation corrections and state tracking baselines. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 109–117, Online. Association for Computational Linguistics.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020. DIALOGPT : Large-scale generative pre-training for conversational response generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 270–278, Online. Association for Computational Linguistics.

A Software Packages Used

Table 4: Software Packages Used

Package	Version	Citation	URL
hdbscan	0.8.27	(McInnes et al., 2017)	`https://hdbscan.readthedocs.io/`
Matplotlib	3.3.4	(Hunter, 2007)	`https://matplotlib.org/`
NumPy	1.19.5	(Harris et al., 2020)	`https://numpy.org/`
Pandas	1.2.4	(the pandas development team, 2020)	`https://pandas.pydata.org/`
plotly	5.1.0	(plotly technologies inc., 2015)	`https://plotly.com/python/`
PyTorch	1.9.0	(Paszke et al., 2019)	`https://pytorch.org/`
scikit-learn	0.24.1	(Pedregosa et al., 2011)	`https://scikit-learn.org/`
SciPy	1.6.2	(Virtanen et al., 2020)	`https://scipy.org/scipylib/index.html`
TensorFlow	2.5.1	(Abadi et al., 2015)	`https://tensorflow.org/`
Transformers	4.11.3	(Wolf et al., 2020)	`https://huggingface.co/transformers/`
umap-learn	0.5.1	(McInnes et al., 2018b)	`https://umap-learn.readthedocs.io/`

B Training Set Covariances For Acceptability Score

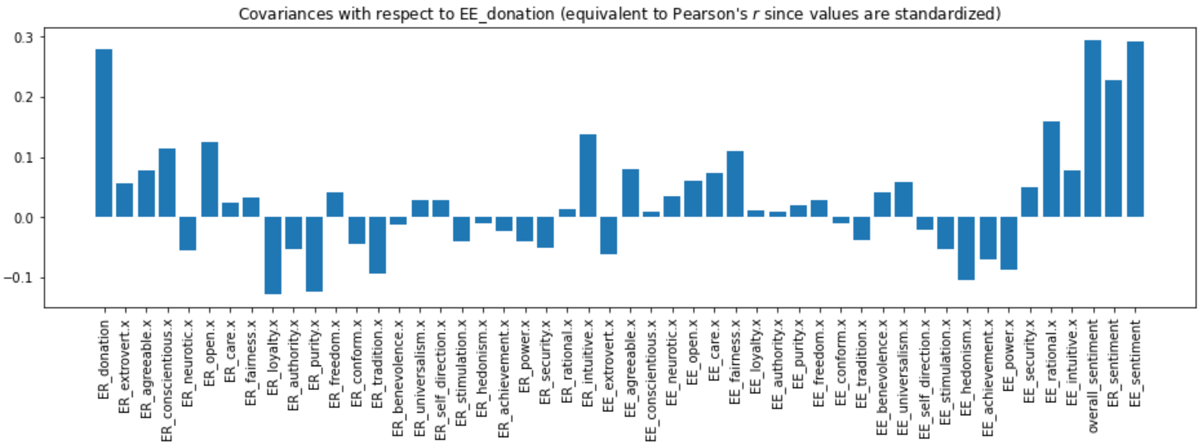

Figure 5: The covariances of all other dialogue attributes with respect to the persuadee donation are used to weight the acceptability score. ER and EE refer to the persuader and persuadee respectively.

C Full Manual Evaluation Results

Table 5: Progression Function Manual Eval Results (All Annotators)

Model	utt (1/2/3)	utt-sl (1/2/3)	dlg-sl (1/2/3)	dlg-sl-f (1/2/3)
unsupervised	0.07 / 0.11 / 0.17‡	0.05 / 0.02 / 0.06	0.02 / 0.01 / 0.00	-0.05 / -0.08 / -0.03
roberta-base	0.17‡/ 0.29‡/ 0.37‡	0.06 / 0.13†/ 0.18‡	0.11 / 0.32 / 0.31	0.14 / 0.25 / 0.32
roberta-large	0.30‡/ 0.42‡/ 0.51‡	0.20‡/ 0.17‡/ 0.25‡	0.08 / 0.48 / 0.47	0.12 / 0.40 / 0.48
roberta-large-adapted	0.40‡/ 0.49‡/ 0.61‡	0.15†/ 0.15†/ 0.24‡	0.20 / 0.64†/ 0.66†	0.22 / 0.55 / 0.67†

two-tailed p-value: †: $p < 0.05$; ‡: $p < 0.01$

D Explanations of Manual Metrics

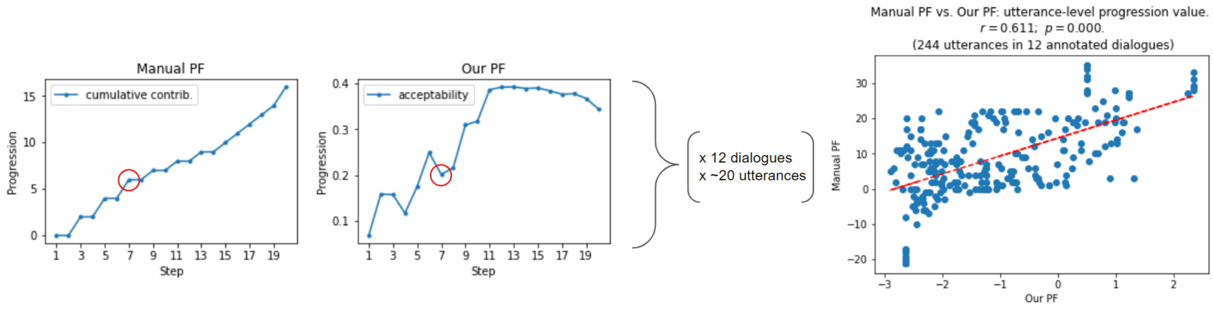

Figure 6: **utt**: Pearson's r (right) between utterance-level PF values (center, e.g., circled) and ground-truth values (left, e.g., circled) for all 244 utterances across 12 dialogues. Points shown on the right are from annotator 3. This metric is intended to measure if the PF and ground-truth progression curves assign similar values (relative to their respective scales) at each step of an ongoing dialogue.

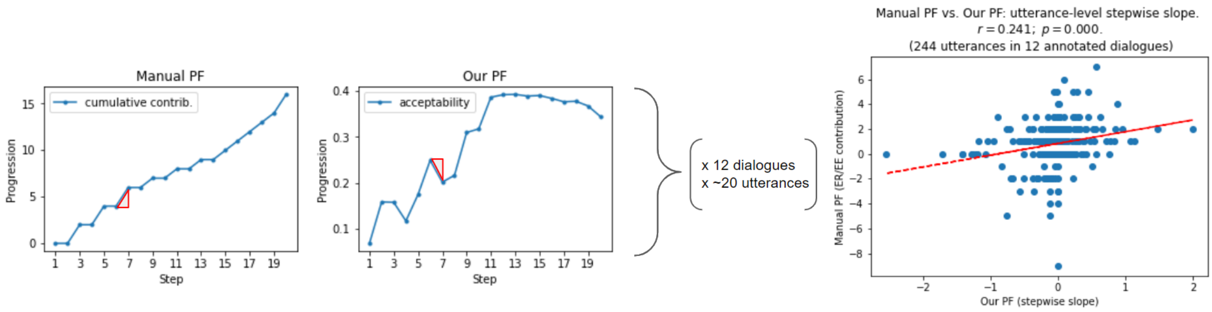

Figure 7: **utt-sl**: Pearson's r (right) between utterance-level PF slopes (center, e.g., see triangle) and ground-truth slopes (left, e.g., see triangle), for all 244 utterances across 12 dialogues. Utterance-level slopes are computed as the differences in the progression curves between two dialogue steps. Points shown on the right are from annotator 3. This metric is intended to measure if the PF and ground-truth progression curves move in the same direction at each step of an ongoing dialogue.

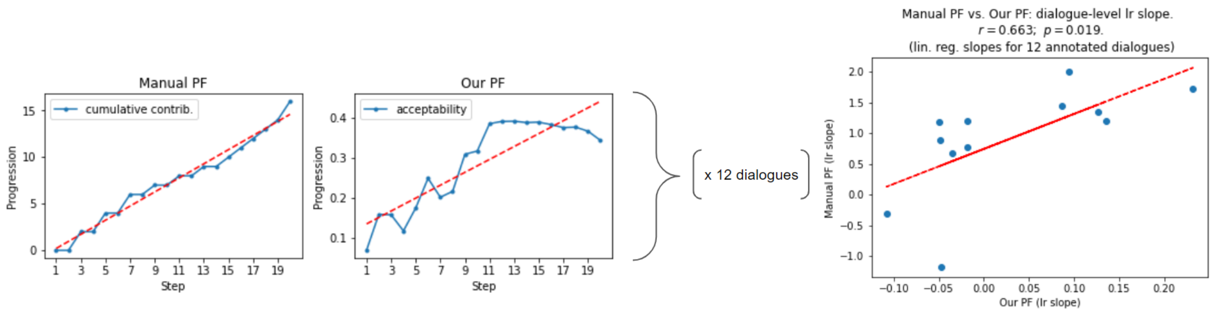

Figure 8: **dlg-sl**: Pearson's r (right) between dialogue-level PF slopes (center, e.g., see line) and ground-truth slopes (left, e.g., see line), for all 12 dialogues. Dialogue-level slopes are computed by fitting least-squares regression lines to the progression curves. Points shown on the right are from annotator 3. This metric is intended to measure the ability of the overall PF trend to approximate the ground-truth progression curve.

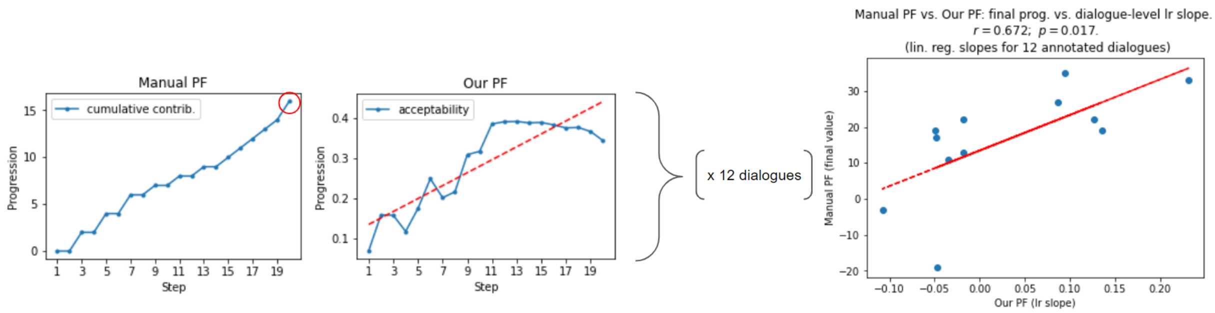

Figure 9: **dlg-sl-f**: Pearson's r (right) between dialogue-level PF slopes (center, e.g., see line) and the final ground-truth progression value (left, e.g., circled), for all 12 dialogues. Dialogue-level slopes are computed by fitting least-squares regression lines to the progression curves. Points shown on the right are from annotator 3. This metric is intended to measure the ability of the overall PF trend to predict the end-task outcome.

E PF Disagreement Examples

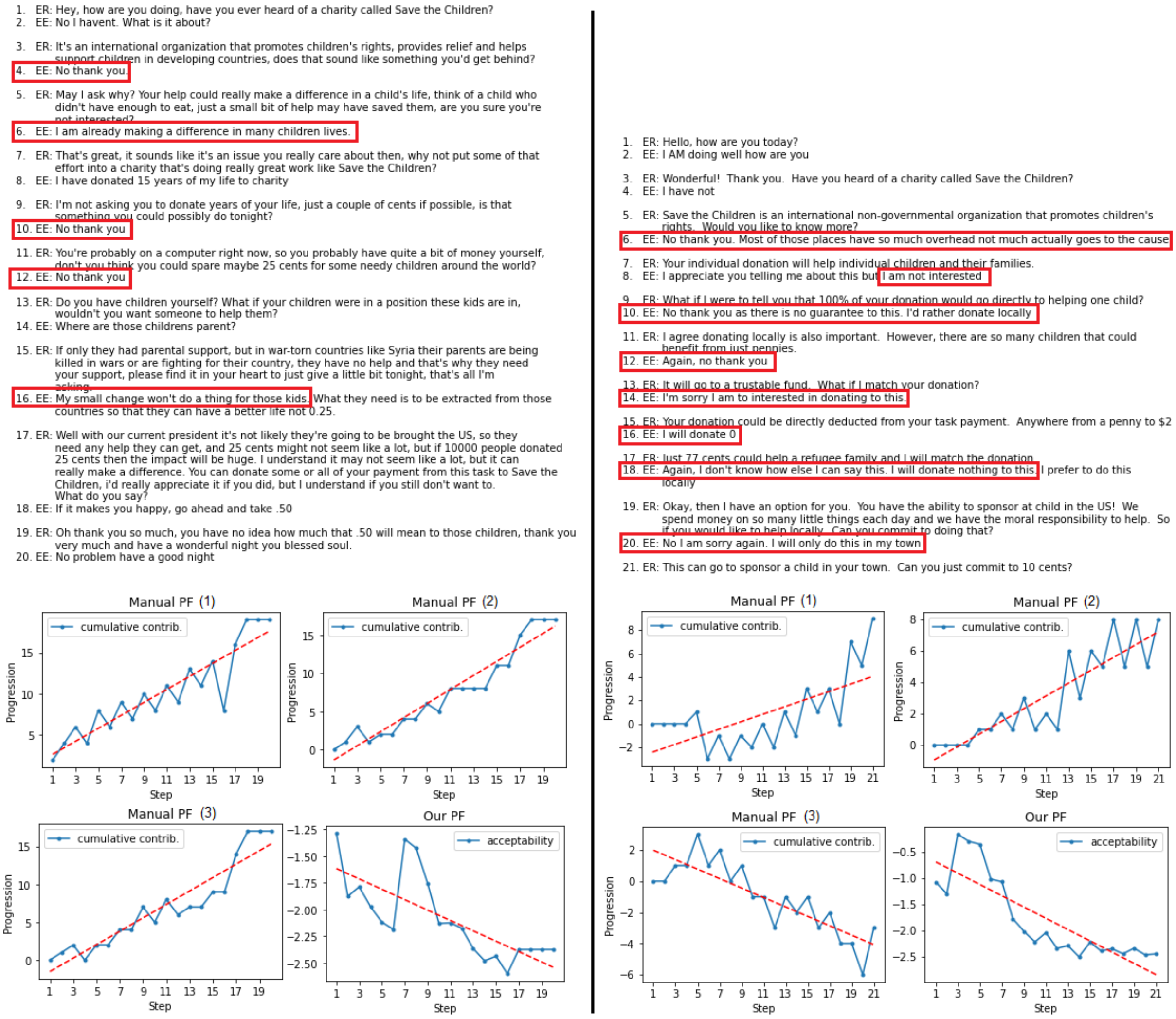

Figure 10: Two examples where the PF* completely disagrees with the ground-truth progression curves, and it can be easily argued that the PF is actually correct.
Left: All three annotators agree that this conversation is progressing positively throughout, yet the utterances highlighted in red show that the persuadee actively refuses to donate until step 18 where a concession is made out of frustration.
Right: Two of three annotators agree that this conversation is progressing positively throughout, yet the utterances highlighted in red show that the persuadee shows distrust of the organization and then refuses to donate without concession for the remainder of the dialogue.
(*PF shown is roberta-large-adapted)

F Grid Search Details for Unsupervised Model

Algorithm 1: Grid search for hyperparameter tuning of the unsupervised progression model on the validation set. Descriptions for each hyperparameter are provided in Table 6.

for $\beta \in \{0.0, 0.1, \ldots, 2.0\}$ **do**
  for $d \in \{2, 16, 32, 64, 128, 768\}$ **do**
    for *normalize_embeddings* $\in$ {*True*, *False*} **do**
      for *distance_metric* $\in$ {*Cosine*, *Euclidean*} **do**
        ▷ `k-means experiments`
        for $k \in \{2, 3, \ldots, 30\}$ **do**
          for *inverse_distance* $\in$ {*True*, *False*} **do**
            for *standardized_proximity* $\in$ {*True*, *False*} **do**
              measure_PF_slope_r();
        ▷ `HDBSCAN experiments`
        for *min_cluster_size* $\in \{10, 20, \ldots, 100\}$ **do**
          for *soft_value_aggregation* $\in$ {*True*, *False*} **do**
            for *prob_scaling* $\in$ {*None*, *softmax*, *sum*} **do**
              for *standardized_proximity* $\in$ {*True*, *False*} **do**
                measure_PF_slope_r();

Table 6: Hyperparameter Descriptions

Hyperparameter	Description
β (recency weight)	Controls how much emphasis is placed on recent tokens when computing dialogue embeddings.
d (embedding size)	The dimensionality of dialogue embeddings. Values < 768 reduced with Parametric UMAP.
normalize_embeddings	If True, embeddings are normalized to have unit magnitude after dimensionality reduction.
distance_metric	The distance metric used by Parametric UMAP and centroid proximity calculations.*
k (number of clusters)	Number of clusters to use for k-means.
inverse_distance	If True, Euclidean centroid proximity is computed as the inverse distance instead of negative distance.**
standardized_proximity	If True, centroid proximities are converted to z-scores before progression is computed. †
min_cluster_size	Minimum number of points in a HDBSCAN cluster. Clusters with fewer points get merged into larger ones.
soft_value_aggregation	If True, HDBSCAN cluster attribute aggregations are weighted with cluster membership probabilities.
prob_scaling	The type of scaling applied to progression computed from HDBSCAN cluster membership probabilities.

*: Centroid proximity calculations refer to progression computation. Proximities used during clustering are always Euclidean.
**: Does not apply to cosine distance.
†: Applies only if k-means is used, or if HDBSCAN is used with softmax probability scaling.

G Final Unsupervised Model Hyperparameters

The final unsupervised model uses k-means ($k = 21$), $\beta = 0.3$, $d = 768$, normalized embeddings, euclidean distance, and inverse distance for centroid proximity.