Tell me who you are and i'll tell you what to do: A Persona Grounded task-oriented Dialogue Generation System

Anonymous ACL submission

Abstract

Modern dialogue agents can broadly be categorized as either chit-chat or task-oriented systems. While the purpose of a chit-chat agent is to entertain and engage the user- lubricating the conversation, so to say-, the task-oriented chatbot is dedicated to fulfilling specific requests (e.g., ticket booking). Current task-oriented agents produce precise but bland and uninteresting responses. While using such agents a user may interpose personal remarks, and the failure of the agent to process and respond to such statements could be a put-off for the user. In this paper we propose a system that is persona-specific, can handle chit-chat utterances, and produces responses that add a human element to the conversation, while always remaining grounded on the task. Since current task-oriented datasets do not have persona-profiles, and do not consist of personalized remarks in utterances, we modify an existing dataset (MultiWOZ 2.1) to suit our needs. We give a semi-automated dataset creation method that uses GPT-2 model trained on the PERSONA-CHAT dataset. A small subset of the obtained data is also manually crafted to acquire a gold standard data. Our framework is based on GPT-2, Graph Convolution Network (GCN) and Memory Network that is trained on this dataset to generate persona-grounded task-oriented responses. Both automatic and manual evaluation show the effectiveness of our model and dataset[1]. Our proposed system achieves a BLEU score of 12.12 on this new dataset.

1 Introduction

The recent rise in deep learning systems has made the modelling of both task-oriented and chit-chat dialogues increasingly reliable and human-like (Hosseini-Asl et al., 2020; Adiwardana et al., 2020). This phenomenon has become even more pronounced with the introduction of transformer based generative pre-trained language models like GPT (Radford et al., 2018) and T5 (Raffel et al., 2019). task-oriented systems aim to achieve high performance in helping user achieve their goal (e.g., travel booking). This is in contrast to the chit-chat dialogue system where the user does not have a specific goal but wants to have an engaging experience. With the increasing success of task-oriented dialogue (TOD) systems, the next target of such dialogue agents should be to engage the user along with helping them achieve their goal. In real world deployment of these TOD systems, the users may sometimes even give casual remarks regarding their preferences and persona, along with their goal driven utterance. A system that just produces responses focused on goal oriented part of utterance (ignoring the part of utterance containing such personalized remarks), would still be able to fulfill the user requests, but the user experience would suffer and the response could come off as dull or even rude. On the other hand, if the TOD system starts producing such remarks on their own when the user just wants to achieve their goal, the experience might be off-putting and tedious to the user. Thus a TOD system that also produces engaging responses should adhere to the following constraints: (i). The system should produce engaging responses only when the user shows an inclination towards it (i.e when the user utters such remarks first), (ii). These responses should be grounded on some pre-assigned persona, so that the utterance does not deviate to any unforeseen or unwanted topic, (iii). The addition of such remarks in the utterances should not hinder the completion of task and should not increase the conversation length.

In this paper we propose steps to create such persona-grounded engaging responses in an existing TOD dataset. These modified system responses are more social and personalized, however they also contain some unwanted utterances that are irrelevant and can be considered noisy. This noisy dataset however, is usable for pre-training of a deep

[1]The codes and datasets will be made available

learning model. A small subset of this dataset is manually cleaned and modified to make the responses more relevant. This small dataset is used for the purpose of fine-tuning our deep-learning model. Next, we build GPT-2 (Radford et al., 2019) based system that can model this dataset. The proposed system is effective at producing relevant task-oriented responses with correct values and intents, while at the same time being engaging and grounded in their persona.

The contributions of this work are: (i). Novel task of generating human like chit-chat responses in a TOD system, where the responses are grounded on a persona-profile (ii). Novel semi-automated corpora creation method, (iii). A corpora consisting of task-oriented responses that are engaging but grounded on persona, and (iv). System that can effectively generate engaging task-oriented responses while still being grounded on background data (task-oriented data) and persona-profile of the agent.

2 Related Works

Research in dialogue systems has progressed rapidly in the past few years. This has led to the development and release of several new datasets. For task-oriented dialogues, Dialogue State Tracking Challenge (DSTC) (Williams et al., 2015; Henderson et al., 2014) provides important resource for intent detection, action prediction and response generation tasks. Another important resource in the space of task-oriented systems is the MultiWOZ corpora (Eric et al., 2019). Moon et al. (2020) propose an interesting dataset in this field named Situated Interactive MultiModal Conversations (SIMMC) dialog containing multi-modal conversations and actions. Frames dataset was proposed by Asri et al. (2017) to study the role of memory in goal-oriented dialogue systems.

In chit-chat domain, datasets have been released to control various aspects of dialogue. Dinan et al. (2018) released a chit-chat corpus where topic of conversation of grounded on a wikipedia extract. Shuster et al. (2018) release a multimodal dataset where the conversation is based around an image, with persona types assigned to the speakers. Thus the conversation is both persona and image grounded in nature. Another dataset that uses persona-profiles to ground conversation is PERSONA-CHAT (Zhang et al., 2018). In our work we follow the persona format used in this paper. We use this data to train a model that injects human like elements to a task-oriented conversation. Few dialogue datasets contain annotations for both task-oriented and chit-chat utterances. For example the task-oriented dialogue corpora constructed by Rastogi et al. (2020) contains annotations for a few chit-chat dialogue acts, but they are limited to light social greetings (e.g., “Thank you!", “Good Bye."). Another recent work by Sun et al. (2020) injects chit-chat utterances to task-oriented data using pre-trained language models. However, these utterances are not grounded to any persona type. In this work we propose a new task of enhancing the task-oriented responses with persona-grounded chit-chat utterances. We also introduce a deep learning system capable of generating such personalized task-oriented response.

3 Dataset Creation

We propose a semi-automated dataset creation method to enhance the system response with engaging and personalized utterances. Our approach uses minimal manual annotation effort in order to obtain such utterances and is able to produce diverse personalized responses. We choose the task-oriented dataset MultiWOZ 2.1 (Eric et al., 2019) and enhance its utterances to obtain the dataset $D1$. Although the creation of this dataset requires low effort, this dataset consists of many inconsistencies and noise. Therefore we select a small subset of this data and remove the noise, fix the inconsistencies and add more personalized responses to the utterances wherever possible. This small dataset $D2$ is kept for the purpose of fine-tuning deep learning models pre-trained on $D1$[2].

3.1 Semi-automated Dataset Creation

We use the following steps for semi-automated dataset creation:

3.1.1 Persona Profile Selection:

In order to generate persona grounded responses, we first need to assign persona to the user and the agent. We follow the persona format described in Zhang et al. (2018) (PERSONA-CHAT dataset), where persona-profile is a set of statements describing the likes and dislikes of a person that could come up in a conversation. Since MultiWOZ 2.1

[2]Detailed statistics and samples of the dataset is given in the appendix

dataset is related to travel and hotel/restaurant booking domains, we create 61 persona statements that would be suitable for these domains. In order to select a suitable user persona-profile from the given set of persona statements, we use BERT (Devlin et al., 2018) sentence similarity score. We use the *CLS* token embedding from pretrained BERT ('bert-base-nli-mean-tokens') and compute the cosine similarity between each user-utterance in the conversation and the persona statements. Out of all the persona statements we filter out the statements having a cosine similarity greater than 0.5 with any user utterance. From these we select top 5 most similar statements to the user utterances. While doing this we write rules to make sure that the persona statements indicating conflicting preferences do not appear together for the same conversation. These selected statements make up the user persona-profile. We use the same profile as the persona-profile of the agent as well. We use this strategy to keep the agent agreeable and non-confrontational. A similar strategy is subconsciously followed by humans as well (Chartrand and Bargh, 1999) where humans mirror the mannerisms of each other in a conversation.

3.1.2 Personalized Utterance Generation:

Once persona has been assigned to the users, the next task is generation of personalized responses. Since our system would produce personalized responses only when user utters such statements first, we first generate personalized user responses. To achieve this we train a GPT-2 (Radford et al., 2019) model on PERSONA-CHAT dataset. Given a persona-profile P of user-1 and conversation history $H = \{u_1, v_1, ..u_n, v_n\}$ where $P = \{p_1, p_2, ..., p_n\}$, p_i being statement describing user preference, u_j being j^{th} utterance from user-1 and v_j being the j^{th} utterance from user-2. We prepare input sequence to the model by appending P and H (we limit the history to previous two utterance in the conversation), where each p_i is separate by a special token '*<persona>*' and each u_i and a_i are preceded by special tokens '*<user>*' and '*<agent>*' respectively. The entire input sequence is preceded by a '*<startoftext >*' token. Our final target sequence is the utterance u_{i+1}. We append this to the input sequence as well, preceding it with the special token as discussed and appending an '*<endoftext >*' token at the end. We train the GPT-2 model for causal language modelling using this sequence. At the time of inference, only the source sequence (consisting of persona-profile and utterance history) is fed to the model and the generated token at each step is appended to the source sequence. This step is repeated until the '*<endoftext >*' token is generated or some maximum length response is generated. Doing this we obtain a BLEU score of 0.53 on the PERSONA-CHAT dataset.

Next we prepare our MultiWOZ data to generate the personalized user utterances. We use the persona-profile P selected for the user (described in Section 3.1.1) and take the dialogue history $H = \{w_1, a_1, .., w_n, a_n\}$ where w_i is the i^{th} user utterance and a_i is the i^{th} agent response. As done during training we limit the dialogue history to two utterances. We use the same special token scheme while appending the persona-profile P and dialogue history H. The trained GPT-2 model is then used to generate the next user utterance. To generate the system utterance we follow the same sequence of steps but we replace the last user utterance with the utterance generated by the trained model. Once the personalized utterances are generated we join them with the original task-oriented responses using the order given by the arranger module.

3.1.3 Arranger Module

The generated personalized utterance needs to be joined with the original task-oriented responses. To decide where in the original response the generated utterance needs to be inserted, we train an arranger module that helps us place the utterances in the correct order. To train the arranger module we make use of utterances in both MultiWOZ and PERSONA-CHAT dataset containing multiple sentences in an utterance. We randomly sample 10,000 utterances that consist of more than one sentence. We obtain another 10,000 samples by jumbling the sequence of these sentences. The utterances with the right order of sentences are labelled 'correct' and the utterances with jumbled order are labelled 'incorrect'. We train a BERT model to classify these utterances using 18,000 of the created utterances and keep 1,000 each for validation and testing. The trained model is able to classify the utterances into 'correct' and 'incorrect' sequence with an F1-Score of 0.84.

The generated personalized utterance is placed with the task-oriented response and the sentence order is shuffled. The obtained sequence is passed through the trained model and classified. The order that is classified as 'correct' is kept and the rest are discarded. In this manner the task-oriented user and agent utterances are merged with their personalized responses.

3.1.4 Utterance Filter

Upon observing the utterances obtained, it could be seen that they belonged to three categories, (i). *Engaging and persona consistent:* These utterances add to the quality of the task-oriented response while being grounded on the assigned persona, (ii). *Engaging and persona independent:* These utterance enhance the response quality, but are independent from the persona-profile of the agent. (iii). *Others:* These utterances do not add to the quality of the original response and can often be incoherent and confusing. The utterances that are coherent yet conflicting with the persona-profile are also kept in this category.

We manually annotate 2,500 samples from the obtained utterances. It was found that out of these utterances 745 belong to the *Engaging and persona consistent class*, 395 were from the *Engaging and persona independent class* while the remaining 1,360 samples were from the *Others* class. Keeping 2,000 utterances as training sample, 200 as validation and 300 as test, we train a BERT classifier. The input to the classifier is the combined sequence of the persona-profile, utterance by the previous speaker and the current utterance. Each of these sequences are separated by a BERT special token. The F1-Score of the trained classifier on test dataset was 0.78. We use this classifier to label all the utterances into these three categories. The utterances that are assigned the class *Others* are replaced by their original task-oriented utterances, while the rest of the utterances are kept as is.

3.2 Manual dataset creation

After generating the responses, finding correct order to arrange them in and filtering out the irrelevant responses, the dataset that remains still consists of some noise. Since the classifiers used are not 100% accurate, the irrelevant responses and invalid orders exist. Along with these, the persona assigned for the conversation does not always match and can be irrelevant in many cases. This often happens in case of conversations where the topic of conversation can span multiple domains. Here matching persona statements from one domain can often be over-represented, resulting in persona from other domains to be ignored. For example if a conversation spans over the domains of travel and restaurant, often the statements related to travel like ‘i enjoy travelling’, ‘i love visiting museums’ etc. can be selected and those related to food preferences can be left out due to the limit of statements in the persona-profile.

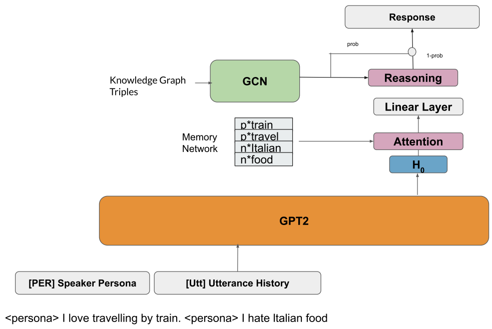

Figure 1: Architectural diagram of the proposed model based on GPT-2, Memory Network and Graph Convolution Network (GCN). Here ‘p’ and ‘n’ are trainable positive and negative embeddings.

We create a small dataset consisting of 2,504 utterances, where these issues in the dataset are fixed and wherever possible, engaging persona grounded responses are added. We distribute this task to 3 annotators, each having masters degree with proficiency in English. A total of 2,504 utternaces from the training set of the created dataset was sampled. The annotators were first asked to read an entire conversation and assign appropriate persona profile of the user that can be inferred by the conversation. Apart from domain specific persona statements we also asked them to add generic and common persona statements (like ‘i am a student’, ‘i am old’ etc.) wherever possible. Once persona-profile is assigned the user and system utterances, were modified to make the conversation more interesting but persona grounded.

4 Methodology

The model consists of five parts (c.f. Figure 1): (i). Causal Language modelling with GPT-2 model, (ii). Memory Network based persona injection, (iii). Graph Encoding, (iv). Reasoning module and (v). Generation with copy mechanism.

4.1 Problem Definition

Given a conversation history of the form $C = \{u_1, s_1, u_2, s_2, ..., u_k\}$, persona-profile $P = \{p_1, p_2, ..., p_n\}$ and the knowledge graph triples $T = \{(h_1, r_1, t_1), (h_2, r_2, t_2), ..., (h_n, r_n, t_n)\}$, where u_i and s_i are the i^{th} user and system utterances respectively, p_j is a persona statement describing the persona of the agent; h_i, r_i and t_i are respectively the head concept, relation and tail concept of the i^{th} triple. The task is to predict the next system utterance u_{k+1} that is grounded on the persona-profile P.

The example below illustrates the task of the system:

Input Sequence:
Persona: i own a car, i love travelling with friends, i enjoy gambling
User: am looking for a place to to stay that has cheap price range it should be in a type of hotel . i've been travelling a lot .
Input Graph: ('rosa's bed and breakfast', 'pricerange', 'cheap'), ('the cambridge belfry', 'pricerange', 'cheap')
Desired Personalized Output: thats great ! i love travelling . do you have a specific area you want to stay in ?[3]

4.2 Causal Language Modelling with GPT-2

We fine-tune the GPT-2 model (Radford et al., 2019) for the causal language modelling task on our dataset. The concatenation of persona-profile, source and target sequences, each separated by their own special tokens (as discussed in section 3.1.2), is fed as the input sequence $IP = (P, U_1, S_1, U_2, S_2 ... U_N, S_N)$ to the model during training . Here, U_i and S_i are the i^{th} utterance of the user and system. The S_N sequence is the target sequence to be generated by the system. At each decoding step t a hidden representation h_t is produced containing the previous context and persona information. The inference steps are the same as described in section 3.1.2.

4.3 Memory Network based persona injection

The persona statements used in the dataset can be categorized into three types: (i). showing positive sentiment of the person towards some entities or concepts. Eg: 'i love train travel', (ii). showing negative sentiment of the person towards some entities or concepts. Eg: 'i find museums boring', and (iii). not showing any sentiment towards any entity or concept. Eg: 'i am a student'.

We label each of the persona statements used in the dataset to positive, negative or neutral depending on the sentiment it represents. Next we extract all the nouns and adjectives from each statement using nltk pos tagger[4]. For each persona statement we align the nouns and adjectives with their sentiment and obtain the tuple list $T = \{(t_1, s_1), ..., (t_n, s_n)\}$ for a persona-profile, here t_i are the nouns and adjectives and $s_i \epsilon \{positive, negative, neutral\}$ is the corresponding sentiment. To model the sentiments in the persona-profiles we make use of an external memory network M and train $v_{positive}$, $v_{negative}$ and $v_{neutral}$ sentiment embeddings. We obtain fastText embeddings for each t_i in a persona-profile and multiply it with its sentiment embedding v_{s_i} to obtain sentiment enriched embeddings m_i. For every conversation, the memory network would store the list of sentiment enriched embeddings $M = [m_1, m_2, ..., m_n]$.

At each decoding step we attend to the the matrix M using the hidden representation h_t produced by the model as described by equation 1 and 2.

$$c_t = \sum_{i=1}^{i=n} \alpha_{t,i} m_i \tag{1}$$

$$\alpha_{t,i} = softmax(v_a^T tanh(W_a[h_t; m_i])) \tag{2}$$

Here v_a and W_a are trainable weight matrices and c_t is the attended memory vector. The attended memory vector is passed through a linear layer and added to h_t to obtain the sentiment enriched hidden representation.

4.4 Graph Encoding

We use the Graph Convolution Network (GCN) Vashishth et al. (2018) for encoding the Knowledge Graph (KG) obtained from background data of the MultoWOZ dataset (knowledge-graph creation and subgraph extraction is explained in detail in section A.2 and A.3 of the appendix). A non-parametric compositional operation $\psi(.)$ is defined to combine node embedding and relation embedding. Given an input graph $G = (V, E)$ and a GCN with L layer, for each node $v \in V$ the node embedding at the $l + 1^{th}$ layer is updated by aggregating information from its local neighbours $N(v)$. These neighbours consist of pairs of node u connected with relation r. The hidden representation of node v for $l + 1^{th}$ layer is obtained by the following equations:

$$e_v^l = \frac{1}{|N(v)|} \sum_{(u,r) \in N(v)} W_N^l \psi(h_u^l, h_v^l) \tag{3}$$

$$h_v^{l+1} = ReLU(e_v^l + W_F^l h_v^l) \tag{4}$$

[3]personalized response shown in green, the rest of the response is task-oriented

[4]https://www.nltk.org/book/ch05.html

Experiment	Pre-training	Fine-tuning	Persona Profile	BLEU	BLEURT	PPL	METEOR
GPT-2+MN+GCN	✓	✓	✓	**12.12**	**0.449**	**66**	**0.253**
GPT-2+GCN	✓	✓	✓	11.76	0.442	71	0.230
GPT-2+MN+GCN	✓	✗	✓	10.17	0.431	83	0.214
GPT-2+MN+GCN	✓	✓	✗	10.46	0.433	81	0.226
GPT-2+MN+GCN	✓	✗	✗	9.93	0.428	86	0.207
GPT-2+MN+GCN	✗	✓	✓	4.16	0.226	107	0.112

Table 1: BLEU, BLEURT, Perplexity (PPL) and METEOR scores for the different experiments. The results show the significance of each module, along with the persona-profile and fine-tuning dataset $D2$. All the results are statistically significant with a confidence interval of 95%

In the above equations, h_v is initialized by its word embedding and h_r is initialized by its relation embedding at layer 0. W_N^L and W_F^L are trainable weight matrices specific to the L^{th} layer. The composition operation is defined as $\psi(h_u, h_r) = h_u - h_r$, similar to the TransE model (Bordes et al., 2013). The relation embedding is also simultaneously updated following Equation 5.

$$h_r^{l+1} = W_R^l h_r^l \tag{5}$$

The weight matrix W_R^l is also trainable for the l^{th} layer. Finally, the node embedding h_v^{LG} and relation embedding h_r^{LG} are obtained from the final layer.

4.5 Reasoning Module

We adapt the method proposed by Ji et al. (2020) to perform reasoning on our knowledge-graph. At each decoding step contextual information is used along with concept and relation information in the knowledge graph to predict the next token. First, the nodes in the graph G corresponding to the concepts in the conversation history is given a score of 1 and all the unvisited nodes are assigned a score of 0. The final node score $ns(v)$ of the unvisited node $v \in V$ is computed by aggregating score from the visited nodes u in the neighbour $N_{in}(v)$ connected with a relation r.

$$ns(v) = \theta_{(u,v) \in N_{in}(v)}(\gamma ns(u) + R(u, r, v) \tag{6}$$

In Equation 6 θ is the mean(.) function, γ is the discount factor $R(u, r, v)$ is the relevance score of the triple obtained by using equation 7

$$R(u, r, v) = \sigma(h_{u,r,v}^T W_{sim} h_t^{LD}) \tag{7}$$

where, $h_{u,r,v}$ is the concatenation of the final GCN layer representation of head, relation and tail concepts, $h_{u,r,v} = [h_u^{LG}; h_r^{LG}; h_v^{LG}]$. All the concept $ns(v)$ scores are finally summed up and passed through a softmax function.

$$P(c_t | s < t, G) = softmax_{v \in V}(ns(v)) \tag{8}$$

where c_t is the selected node at the t^{th} time-step. The reasoning module, thus, learns to select the correct concept given the current decoder state.

4.6 Generation with Copy Mechanism

The final generation uses the copy mechanism to either generate a token or copy a concept from the knowledge-graph. A soft gate probability p_t is used to weigh the distribution of vocabulary and concepts (See et al., 2017).

$$p_t = \sigma(W_p h_t) \tag{9}$$

Here, W_p is a trainable weight matrix. The final output distribution is given by the following equation.

$$P(y_t | x, G) = p_t P(c_t | s_{t-1}, G) + (i - p_t) P(s_t | s_{t-1}) \tag{10}$$

In the equation 10, y_t is the probability distribution of the final output at the t_{th} time-step. s is the context representation obtained from GPT-2; c is the concept representation, given the context and sub-graph G.

The model is finally trained to by minimizing the negative log-likelihood of generating the target response sequence $y_{target} = \{y_1, y_2, y_3, ..., y_N\}$ given in equation 11

$$L = \sum_{t=1}^{N} -logP(y_t^{target} | y_{<t}^{target}, x, G) \tag{11}$$

5 Experiments

We conduct experiment on our proposed model using our created datasets $D1$ and $D2$. In our main experiment we use the dataset $D1$ for pre-training and once the model is trained, we fine-tune the model using $D2$. To show the significance of the pre-training step, we obtain results by training our model only on $D2$. We conduct ablations on our memory network based persona injection module

Experiment	Pre-training	Fine-tuning	Persona Profile	PC	Adequacy	Fluency	DB Consistency
GPT-2+MN+GCN	✓	✓	✓	**76%**	**3.24**	3.7	60%
GPT-2+GCN	✓	✓	✓	73%	3.09	**3.8**	**63%**
GPT-2+MN+GCN	✓	✗	✓	69%	2.93	3.3	61%
GPT-2+MN+GCN	✓	✓	✗	65%	2.88	**3.8**	58%
GPT-2+MN+GCN	✓	✗	✗	60%	2.61	3.5	60%
GPT-2+MN+GCN	✗	✓	✓	49%	1.98	3.5	24%

Table 2: Manually evaluated Persona Consistency (PC), Adequacy, Fluency and Database Consistency scores for the different experiments. All the results are statistically significant with a confidence interval of 95%

and highlight its importance. We also perform ablation on persona-profile and show that the persona-profiles help in grounding the responses based on persona. The implementation details of the experiments is elaborated in section A.4 of the appendix.

5.1 Evaluation Metrics

We evaluate the results of our experiments using both automatic and human evaluation methods [5]. For automatic evaluation we use: (i) **BLEU**, (ii). **BLEURT**, (iii) **Perplexity (PPL)**, and (iv). **METEOR**.

For human evaluation we use the following metrics: (i). **Persona Consistency (PC)**, (ii). **Fluency (Gra)**, (iii). **Adequacy (Con)**, and (iv). **Database Accuracy**.

Three human experts with post-graduate qualifications were asked to rate 100 responses generated from the proposed model. The rating for Fluency and Adequacy was done on a Likert scale (1 to 5), with 1 being the lowest and 5 being the highest. For database accuracy the experts had to cross-verify the concepts appearing in the response with the database for calculating the accuracy (0% to 100%). The experts had to label if a generated utterance is consistent with the persona profile, and percentage of utterances that were consistent was computed for Persona-Consistency. A multi-rater Kappa (McHugh, 2012) agreement ratio of approximately 80% was observed for persona-consistency, fluency and adequacy, which may be considered as reliable.

6 Results and Analysis

The results obtained by automatic and manual evaluation are shown in Tables 1 and 2 respectively. Automatic evaluation shows that fine-tuning our model with $D2$ along with using the persona-profile at input gives the best results. Removing the fine-tuning step the performance drops by almost 2 points in terms of BLEU. Similarly removing persona-profile from input sequence drops the BLEU score by 1.36 points. Removing both of these steps results in a big drop in BLEU score ($\downarrow$ 2.19 points), giving the lowest score of all the experiments. Similar results can be observed in BLEURT, perplexity and METEOR scores. The removal of our persona-injection memory network results in a slight drop in performance. This is because the system is better able to capture positive and negative sentiments towards an aspect, thus resulting in more accurate persona grounding of responses. The biggest drop in performance in terms of all the metrics occurs on removal of pre-training step. This shows the significance of our automated dataset $D1$. Although noisy, this dataset is very useful in training the model to understand intents, fetching correct data from the knowledge-graph and generation of coarse personalized response.

Manual evaluation confirms the effectiveness of use of persona-profile and fine-tuning (on dataset $D2$) in terms of persona-consistency and adequacy scores. However adding the persona-injection memory network results in a slight drop in DB-consistency ($\downarrow$ 3 percentage points), even though it results in better persona consistency ($\uparrow$ 3 percentage points). The best fluency and DB-consistency is obtained excluding the memory network and using persona-profile and fine-tuning. The importance of pre-training the model on $D1$ is again highlighted by huge drop performance w.r.t each evaluation metric. The big drop in persona-consistency ($\downarrow$ 27 percentage points), DB-consistency ($\downarrow$ 36 percentage points) and adequacy ($\downarrow$ 1.26 points) shows that dataset $D2$ alone is inadequate for training. The pre-training of our model on $D1$ is consequential in training of GCN module and GPT-2 for implicitly understanding the intent taking actions.

Table 3 compares the outputs produced by different systems, highlighting the importance of each component. In the first example, it can be observed

[5]Detailed descripion of metrices is given in section A.5 of the appendix

Input	Persona Profile: [I am a college student , I like historical aritfacts, I enjoy travelling by trains, I have a lot of money, i love italian food] user: hi, i'm a student . i would like to visit a museum in the center of town .
Gold	agent: cool what is your major ? do you have a particular museum in mind or would you like suggestions ?
GPT-2+MN+GCN [without persona profile]	agent: there are 11 museums in the centre of town. is there a particular type of museum you are interested in?
GPT-2+MN+GCN [with persona profile]	agent: i love history too. there are 23 museums in the centre of town. is there a particular type of museum you are interested in?
Input	Persona Profile: [I enjoy travelling by trains, I cannot stay without internet] user: that sounds good , how much is a ticket ?
Gold	agent: the price is 12.80 pounds . would you like me to book it ?
GPT-2+MN+GCN [without fine-tuning]	agent: it is a lot of money. what do you do for a living? the price of the ticket is 12.80 pounds. would you like me to book it for you?
GPT-2+MN+GCN [with fine-tuning]	agent: the price is 12.80 pounds. would you like me to book it for you?
Input	Persona Profile: [i am a busy person, i do not like train journey, i like jamaican food] user: alright, can you find me a train to leicester on tuesday after 20:15 ? i hate travelling by trains.
Gold	agent: i am sorry , my mind is not all there today . i have not found a train for you yet . where are you departing from ? i hate train journey as well .
GPT-2+GCN	agent: i love travelling too . i am sorry i could not find the train. where are you departing from ?
GPT-2+MN+GCN	agent: i find train travel difficult . where will you be departing from ?

Table 3: Output quality comparison of our final system with systems trained without persona-profile, fine-tuning or memory-network

that without using persona-profile as input to the model the personalized response is missed in this case. When persona-profile is used, the response produced is more engaging and persona grounded even though different persona aspect (to that of gold) is used to produce this response. The numbers produced do not match as the model is incapable of performing counting operations. In the second example we see that without fine-tuning the model on $D2$ the model generates engaging response, but it is not completely appropriate for a task-oriented agent. After fine-tuning, only the task-oriented response is generated which is more suitable in this case, as it is not possible or appropriate to produce such remarks at every utterance. In the final example, the model without memory-network produces response that conflicts with its persona-profile. In this case the persona had a statement 'i do not like train journey', while in response the agent says 'i love travelling too'. This is fixed by the model using memory-network based persona injection, and the agent response ('i find train travel difficult') becomes consistent with the persona-profile.

The current method suffers from database mismatch problem where the entities produced do not match the entities in the gold sequence, but still fulfill the user request. As shown in the first example of Table 3, the model could generate responses that are personalized but use different persona statement from that of gold sequence. These are some factors that reduce the performance of the system even while generating valid utterances.

7 Conclusion

In this paper we propose a new task of enhancing the response of task-oriented system to make it more engaging and personalized. The agent should thus be able to produce response that are persona-grounded and task-specific at the same time. We present a novel semi-automated dataset creation method that modifies the MultiWOZ 2.1 dataset by assigning persona and rewriting the responses. The new responses are more personalized and interesting while still being task-oriented. A small subset of this new dataset is cleaned and modified manually. We also propose a deep learning model that makes use of GPT-2, GCN and Memory-Network to model this agent. Through different evaluations and experiments we show the effectiveness of our datasets and models.

In future it would be interesting to explore systems that can easily adapt to new personas and new task domains.

8 Ethical Declaration

We use two freely available datasets to create our new dataset. These datasets have only been used for the purpose of academic research. The dataset created in this work will be made available only after filling and signing an agreement declaring that the data will be used only for research purposes. The annotation for manual evaluations was done by human experts, who are the regular employee of our research group. There are no other issues to declare.

References

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*.

Layla El Asri, Hannes Schulz, Shikhar Sharma, Jeremie Zumer, Justin Harris, Emery Fine, Rahul Mehrotra, and Kaheer Suleman. 2017. Frames: a corpus for adding memory to goal-oriented dialogue systems. *arXiv preprint arXiv:1704.00057*.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 26.

Tanya L Chartrand and John A Bargh. 1999. The chameleon effect: the perception–behavior link and social interaction. *Journal of personality and social psychology*, 76(6):893.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2018. Wizard of wikipedia: Knowledge-powered conversational agents. *arXiv preprint arXiv:1811.01241*.

Mihail Eric, Rahul Goel, Shachi Paul, Adarsh Kumar, Abhishek Sethi, Peter Ku, Anuj Kumar Goyal, Sanchit Agarwal, Shuyang Gao, and Dilek Hakkani-Tur. 2019. Multiwoz 2.1: A consolidated multi-domain dialogue dataset with state corrections and state tracking baselines. *arXiv preprint arXiv:1907.01669*.

Matthew Henderson, Blaise Thomson, and Jason D Williams. 2014. The second dialog state tracking challenge. In *Proceedings of the 15th annual meeting of the special interest group on discourse and dialogue (SIGDIAL)*, pages 263–272.

Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. 2020. A simple language model for task-oriented dialogue. *arXiv preprint arXiv:2005.00796*.

Haozhe Ji, Pei Ke, Shaohan Huang, Furu Wei, Xiaoyan Zhu, and Minlie Huang. 2020. Language generation with multi-hop reasoning on commonsense knowledge graph. *arXiv preprint arXiv:2009.11692*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Mary L McHugh. 2012. Interrater reliability: the kappa statistic. *Biochemia medica*, 22(3):276–282.

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in pre-training distributed word representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Seungwhan Moon, Satwik Kottur, Paul A Crook, Ankita De, Shivani Poddar, Theodore Levin, David Whitney, Daniel Difranco, Ahmad Beirami, Eunjoon Cho, et al. 2020. Situated and interactive multimodal conversations. *arXiv preprint arXiv:2006.01460*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

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.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

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

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*.

Thibault Sellam, Dipanjan Das, and Ankur P Parikh. 2020. Bleurt: Learning robust metrics for text generation. *arXiv preprint arXiv:2004.04696*.

Kurt Shuster, Samuel Humeau, Antoine Bordes, and Jason Weston. 2018. Image chat: Engaging grounded conversations. *arXiv preprint arXiv:1811.00945*.

Kai Sun, Seungwhan Moon, Paul Crook, Stephen Roller, Becka Silvert, Bing Liu, Zhiguang Wang, Honglei Liu, Eunjoon Cho, and Claire Cardie. 2020. Adding chit-chats to enhance task-oriented dialogues. *arXiv preprint arXiv:2010.12757*.

Shikhar Vashishth, Prateek Yadav, Manik Bhandari, Piyush Rai, Chiranjib Bhattacharyya, and Partha Talukdar. 2018. Graph convolutional networks based word embeddings. *arXiv preprint arXiv:1809.04283*.

Jason D Williams, Matthew Henderson, Antoine Raux, Blaise Thomson, Alan Black, and Deepak Ramachandran. 2015. The dialog state tracking challenge series. *BE A PART OF AN*, page 121.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing dialogue agents: I have a dog, do you have pets too? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2204–2213.

A Appendix

A.1 Dataset Details

Dataset	Dialogues	Utterances	Enhanced Utterances
D1	10,907	143,048	24,226
D2	367	2,504	491

Table 4: Detailed statistic of the semi-automaically created dataset $D1$ and manually created dataset $D2$

A.2 Knowledge Graph Creation

The MultiWOZ dataset contains the database of booking information of all the domains. The database consists of dictionaries for each entry, the key of which is treated as relations (edges) and the values as concepts (nodes). The name value in the dictionary is used as the head node in 'restaurant', 'hotel' and 'attraction' database. The rest of the values are related to the head node with the relation given by the dictionary keys of the corresponding values. For taxi database, all the values in the key *taxi_types* are connected with all the keys in *taxi_colors* with the relation being *taxi_colors* between them. In training database, the value under key *trainID* is treated as head for each item, and all other values in the dictionary are connected to the head with the relation being the respective key in the dictionary.

Finally, all the graphs are represented together (for each dataset) as $G_{total} = (V_{total}, E_{total})$ where V_{total} and E_{total} are the vertices and edges of the graph. Total concepts (nodes) in the final knowledge graph is 3,803; and total relation types in the knowledge graph is 26.

A.3 Sub-graph Extraction

Since the original knowledge graph G_{total} is huge, we select a sub-graph G w.r.t the context of the conversation to feed to the GCN. Part-of-Speech (PoS) tagging is done by Spacy[6] for all the user utterances in the conversation history. We first filter out the stop-words from the user utterances of the conversation text and store all the nouns, verbs, adjectives and numbers in a list L. From the knowledge graph G_{total}, we extract all the vertices with value matching any item in L and store it in V. Selecting each vertex we select all the neighbouring nodes and add it to the list L, and V. Using all the vertices V and the edges between them as E (relations between concepts) we form the sub-graph $G = (V, E)$.

A.4 Implementation Details

We use the small version of GPT-2 consisting of 12 layers, 768 dimension hidden state and 12 attention heads. The GCN used in our experiment consists of 2 layers. We limit the number of triples to 120 during sub-graph extraction, and context size (dialogue history) to 2 previous utterances by the user. Length of the generated response is limited to 150 tokens. Adam optimizer (Kingma and Ba, 2014) was used to train the model with the values of β_1 = 0.9, β_2 = 0.999 and $\epsilon = 1 \times 10^6$. To obtain the concept embedding during training, we use GPT-2, while we train a separate embedding for the relation type. The dimension of the relation embedding is kept as 768 to match the embedding dimension produced by GPT-2-small. FastText embeddings (Mikolov et al., 2018) are used to represent the words in the memory network. The best model is selected based on the obtained BLEU score on the validation set. All the implementations are done using pytorch (Paszke et al., 2019) library. The huggingface implementation of GPT-2 is used in our experiments.

A.5 Evaluation Metrics Details

(i). **BLEU:** We compute and report BLEU scores against the ground truth response. This score could be treated as a measure of content preservation

[6] https://spacy.io

from the input. (ii). **Perplexity:** We compute perplexity (PPL) to measure how likely the occurrence of a sentence is. Lower the perplexity, higher the probability of occurrence. (iii). **BLEURT:** It is machine learning-based automatic evaluation metric (Sellam et al., 2020) that has the ability to capture non-trivial semantic similarities between sentences. Its value varies roughly between 0 and 1; where 0 signifies random output and 1 denotes perfect match. We use BLEURT-20 checkpoint to compute the score[7]. (iv). **METEOR:** Meteor considers exact word (unigram) mapping, followed by stemmed-word matching, and finally synonym and paraphrase matching for computation of score with a reference document.

(i). **Persona Consistency (PC):** It measures the percentage of generated responses that are consistent to the persona-profile (ii). **Fluency (Gra):** It measures the grammatical correctness of the response. (iii). **Adequacy (Con):** It measures if the information in the predicted output is semantically same as that of ground truth. (iv). **Database Accuracy:** For every concept generated at the output, it measures the percentage of other concepts appearing along with them and is consistent with the database.

[7]code found in the link: `https://github.com/google-research/bleurt`