ToD4IR: A Humanised Task-Oriented Dialogue System for Industrial Robots

CHEN LI\(^{1}\), (Member, IEEE), XIAOCHUN ZHANG\(^{2}\), (Member, IEEE), DIMITRIOS CHRYSTOMOU\(^{3}\), (Member, IEEE), AND HONGJI YANG\(^{3}\)

\(^{1}\)Department of Materials and Production, Aalborg University, 9220 Aalborg, Denmark
\(^{2}\)School of Management Science and Computer, Anhui University of Finance and Economics, Bengbu, Anhui 233030, China
\(^{3}\)School of Computing and Mathematical Sciences, University of Leicester, Leicester LE1 7RH, U.K.

Corresponding author: Xiaochun Zhang (xiaochun.zhang@aufe.edu.cn)

This work was supported in part by the Natural Science Foundation of the Education Department of Anhui Province under Grant KJ2020A0012, in part by the Program of School Scientific Research of Anhui University of Finance and Economics under Grant ACKYC22092, in part by the EU’s SMART EUREKA Programme under Grant S0218-chARmER, in part by the Innovation Fund Denmark under Grant 9118-0001B, and in part by the H2020-WIDESPREAD “Networking for Research and Development of Human Interactive and Sensitive Robotics Taking Advantage of Additive Manufacturing–R2P2” under Project 857061.

ABSTRACT

Despite the fact that task-oriented conversation systems have received much attention from the dialogue research community, only a handful of them have been studied in a real-world manufacturing context using industrial robots. One stumbling block is the lack of a domain-specific discourse corpus for training these systems. Another difficulty is that earlier attempts to integrate natural language interfaces (such as chatbots) into the industrial sector have primarily focused on task completion rates. When designing a dialogue system for social robots, the user experience is prioritized above industrial robots. To overcome these challenges, we provide the Industrial Robots Domain Wizard-of-Oz dataset (IRWoZ), a fully-labeled discourse dataset covering four robotics domains. It delivers simulated discussions between shop floor workers and industrial robots, with over 401 dialogues, to promote language-assisted Human-Robot Interaction (HRI) in industrial settings. Small talk concepts and human-to-human conversation strategies are provided to support human-like answer generation and give a more natural and adaptable dialogue environment to increase user experience and engagement. Finally, we propose and evaluate an end-to-end Task-oriented Dialogue for Industrial Robots (ToD4IR) using two types of pre-trained backbone models: GPT-2 and GPT-Neo, on the IRWoZ dataset. We performed a series of trials to validate ToD4IR’s performance in a real manufacturing context. Our experiments demonstrate that ToD4IR outperforms three downstream task-oriented dialogue tasks, i.e., dialogue state tracking, dialogue act generation, and response generation, on the IRWoZ dataset. Our source code of ToD4IR and the IRWoZ dataset is accessible at https://github.com/lcroy/ToD4IR for reproducible research.

INDEX TERMS

Natural language processing, interactive systems, data collection, neural networks, human–robot interaction.

I. INTRODUCTION

Conversational artificial intelligence (AI), such as dialogue systems and chatbots, has received much attention in recent years and reaped various positive results [1], [2], [3]. Despite its enormous success, developing robots with human-like natural language response capabilities through conversational AI is still challenging. Many studies have been conducted in four directions to construct human-like dialogue systems that improve the user experience: knowledge, empathy, engagingness, and humanness [4], [5], [6], [7], [8]. The majority of these studies, however, are focused on open-domain dialogue systems, whereas task-oriented dialogue (ToD) aims to meet the functional requirements [3], [9] of specialized domains.
Furthermore, some research works have been done on designing and developing such ToD for the manufacturing domain, especially in human-robot interaction (HRI) for industrial robots [10], [11]. While several studies have been presented to ground natural language commands for industrial robot manipulation [12], [13], the majority do not need dialogue datasets to train a neural network since they employ the keywords-matching approach, or their dialogue datasets are not publicly available. Additionally, while adopting language-enabled ToD to improve HRI for industrial jobs is a novel concept, there is currently no pre-built dialogue corpus for training such ToD.

The contributions of our work can be summarized as follows:

- As a new benchmark, we provide the Industrial Robots Wizard-of-Oz dataset (IRWoZ), a scalable, innovative manufacturing domain-centered ToD dataset that may be used to create ToD in HRI for industrial robots. IRWoZ is open to the public, facilitating successful collaboration between academics and industrial partners.
- We introduce small talk principles and human-to-human conversation strategies to assist the human-like response generation build the IRWoZ. To the best of our knowledge, it is the first effort to produce a human-like response in HRI for industrial robots.
- We offer ToD4IR, a conversational AI, fine-tuned on our IRWoZ corpus utilizing two types of State-of-The-Art (SoTA) pre-trained backbone models (GPT-2 and GPT-Neo).
- We demonstrate the robustness of our approach in a real-world factory setup.

The remainder of this work is structured as follows: Section II summarizes the related work. Section III then outlines how we created the IRWoZ dataset, the ToD4IR system architecture, and essential components. Section IV summarizes the findings of the evaluation and discussion. The paper is concluded in Section V.

II. RELATED WORK

A. TASK-ORIENTED DIALOGUE SYSTEMS

In general, ToD aims to accomplish specific tasks through dialogue with the user, such as booking a taxi or ordering food. Such systems are typically classified into two categories: pipeline and end-to-end [14]. Compared to a pipeline manner, partial or complete end-to-end dialogue systems have received more attention in recent years [15]. Each component is trained end-to-end, reducing sub-model errors’ propagation and accumulation. Furthermore, passing user feedback to a model is challenging because each module is interdependent in a pipeline pattern [16]. The evaluation of such end-to-end ToD systems mainly focuses on dialogue state tracking and response generation. Lei et al. presented a sequence-to-sequence method incorporating dialogue state tracking and response generation [2]. Li et al. proposed an end-to-end neural dialogue system for achieving targets and reaching higher accuracy [17]. Perez et al. an end-to-end memory network, a memory-enhanced neural network architecture, for dialog state tracking [18]. The proposed Alternating Roles Dialog Model (ARDM) uses the large pre-trained language model. It does not require belief states or dialog acts from human annotations [19]. Chen et al. introduced a graph attention network to extract information from utterances and graphs and leveraged a recurrent graph attention network to control state updating [20]. Wu et al. proposed a pre-trained ToD-BERT that models dialogue behavior during pre-training and outperforms downstream tasks, e.g., response selection [21]. Unlike TOD-BERT, Minimalist Transfer Learning (MinTL) used a copy mechanism to inject the previous dialogue states into the new one and improve the end-to-end response generation [22]. In SimpleTOD, Hosseini-Asi et al. took the whole ToD as a single sequence prediction problem, leveraged transfer learning from a pre-trained language model based on open-domain, and improved the performance for dialogue model [9]. Soloist used task-grounded pre-training to learn tasks while enjoying low annotation cost for the training dataset and reached a higher task success rate [23]. Human feedback is considered during the training stage in the end-to-end model [24] to improve the system performance. Bhuwan et al. presented an end-to-end differentiable KB-Infobot, which improved the robustness of the system and flexibility of question types [25].

The most related works to ours are [9], [23], which leverage the pre-trained language model to build end-to-end ToD systems. However, those works do not distinguish the dialogue belief state but treat them as a whole entity. Our work divides the belief state into four groups: database-related, task-related, required, and optional, based on the source and importance, respectively. It helps to generate system actions and responses with higher efficiency and accuracy.

B. CHIT-CHAT DIALOGUE SYSTEMS

In comparison to task-oriented dialogue systems, chit-chat systems place a higher premium on small talk, sociability, engagement, and humanness.

He et al. investigated the effects of various pre-trained fine-tune schemes. They found that some significant language generation methods can be forgotten due to data separation. They proposed a mix-review method based on the data mixture idea and effectively alleviated language skill forgotten problems [26]. Daniel et al. employed a sequence-to-sequence model, Meena, which includes two 260 million parameters. Compared to other chatbots, Meena is an open-domain model based on a multi-turn transformer architecture; it can give unique and more reasonable responses. The paper also presented a new Sensibleness and Specificity Average (SSA) index for Manual evaluation [27]. Sun et al. integrated Chit-Chat to enhance task-oriented dialogue and achieve the goal of making virtual assistant conversations more engaging and interactive [28]. Roller and his research team think good conversation requires varied skills, including engagement, knowledge, empathy, and personality [4].
Zhang et al. developed a DIALOGPT model, trained on a massive real-world Reddit dataset [29]. Moon et al. proposed the model with light social greetings annotations for a few chit-chat dataset [30]. Another work suggests randomly sampling utterances from a chit-chat corpus to improve the out-of-domain recovery performance [31]. Xiao’s design considers intelligence and emotion together, which advances communication, engagement, and social belonging [32]. Shu proposed a Meta-Learning framework (DAML) based on random source domains with disparate label sets and achieves high performance on an unknown target domain [33]. Asma proposed a self-play metric where the dialog system talks to itself. They showed that this metric was similar to the human-rated quality for a dialog model and better than other automated metrics [34]. Hannah et al. proposed a new model based on empathetic dialogue generation and created a novel dataset of 25k emotional dialogue [5]. Mazare et al. attempted to focus on personal facts to make their chit-chat dialogue model more engagements [35]. Akasaki and Kaji annotate user utterances with chat/non-chat binary labels [3].

While the aforementioned chit-chat dialogue systems are primarily concerned with the open domain or function more like personal assistants, borrowing such features (e.g., small talk, engagement) that enable users to speak naturally in order to complete tasks more efficiently is also critical for developing robust and humanized task-oriented dialogue systems.

C. DIALOGUE DATASETS

A dialogue dataset is critical for building data-driven-based conversational AI. Such datasets may cover many different domain categories. Daily dialogue is a human-written dataset based on communication intention and emotional information and achieves good performance in multi-turn dialogues [36]. Cornell Movie-Dialogs corpus is generated in a social context from movies. This corpus contains a large metadata-rich collection of fictional conversations extracted from a raw movie, including 304,713 utterances in total [37]. Sun et al. bridge the two baselines, and compare ACCENTOR-SGD and ACCENTOR-MultiWOZ with original SGD [20] and MultiWOZ [38], [39], [40] datasets [28]. The ConvAL2 is based on large pre-trained transformers [41]. BlendedSkillTalk is a conversation dataset of about 7k entries explicitly designed to exhibit multiple conversation modes, focusing on personality, empathy, and knowledge [42]. Bookscorpa includes some fixtures and the integration of proxy usage, which downloads books from the original smashwords dataset where all books are written in English with at least 20k words. Wolf et al. used the BooksCorpus dataset to get the best perplexities and F1 scores [43]. The Ubuntu Dialogue Corpus contains over seven million utterances and 100 million words for multi-turn dialogues. It is based on neural language models with large amounts of unlabeled data [44].

While new datasets have been developed constantly, none of them are specifically targeted toward industrial robots. In our study, we create a novel dialogue corpus focusing on four common topics for HRI in industrial robotics. Additionally, we seek to increase shop floor workers’ engagement and user experience by offering a natural, humanized conversation environment. That is, to enable ToD4IR to communicate in natural human language. To do this, ToD4IR must be trained on a corpus of natural human-to-human conversations to understand how to generate a human-like response. While the SoTA work [8] incorporates chit-chat to augment its ToD and utilizes pre-trained generative models to produce freeform chit-chat data, it requires a hybrid classifier to filter candidate datasets and crowd workers to annotate data. Nevertheless, our dataset is enriched by including small talk and human-to-human conversation strategies during the dataset collection process. No extra classification or interaction by human workers is required to validate the candidate response, as we collect the discussion corpus directly using “Wizard-of-OZ,” a human-to-human technique.

III. METHOD

The approach for producing the IRWoZ corpus, which mixes small talk and human-to-human conversation strategies with task-oriented dialogue and is used in our pre-training, is described in this section. The proposed ToD4IR’s overall system architecture, as well as the backbone models trained on a dialogue history level, are then presented.

A. DATASET CREATION

Among our contributions is the development of an industrial robot-oriented dialogue corpus, IRWoZ, a fully-labeled collection of human-to-human conversations spanning over four domains (assembly and relocation tasks of industrial manipulator, delivery and positioning tasks of mobile industrial robots). It seeks to create simulated dialogues between shop floor workers and industrial robots in order to facilitate language-assisted HRI in an industrial setting, with a total of 401 dialogues.

This section describes the four-step process by which the IRWoZ dataset was created. Firstly, we investigated the most typical scenarios for HRI in industrial robots to identify areas ideal for incorporating language-enabled ToD. Secondly, we leverage the WoZ method for collecting domain-specific conversation datasets, which receive less attention and are not readily available online or on the market. Thirdly, we examined work-related small talk principles to guide the humanized response generation to boost the user experience. Finally, we explored human-to-human conversation strategies to provide a more meaningful task-oriented response and maintain a high task completion rate.

1) SCENARIOS FOR INDUSTRIAL ROBOTS

In general, industrial robots are employed in the following scenarios [45]:

- **Material handling**, including material transfer, which requires the robot to move materials or work parts from one location to another, and **machine loading and**
unloading, which utilizes a robot to load and unload parts at a production machine.
- **Processing operations** where a robot manipulates a tool to carry out a process on the work part, e.g., assembly, inspection.
- **Service operations** where a robot fulfills several operational services, e.g., repairing the manufacturing equipment, cleaning up the waste, and scrap after manufacturing tasks.

In comparison to machine loading and unloading, ToD is more involved in the material transfer scenario, e.g., internal transportation, in which the user requests a delivery task by informing a mobile robot (e.g., Mobile Industrial Robot (MIR)) the destination of the transportation, recipient, and optional information of parcels (e.g., size, color). Fig. 1 (b) depicts an example dialogue of such scenario. Human collaborates with an industrial manipulator for assembly tasks (e.g., Universal robots, Franka Emika), on the other hand, is also common in a production environment where ToD fits well. For instance, a shop floor worker may request a Franka Emika robot to assemble a product via dialogue in which the ToD system gathers the information of the product type and amount and controls the robot to accomplish the assembly operation. An example dialogue for such assembly scenario is illustrated in Fig. 1 (a). However, service operations, such as cleaning and repairing, are too task-specific to be generalized for ToD.

2) **DATA COLLECTION - A WOZ WAY**

In general, most of the ToD dataset is built based on existing dialogue systems [38]. To our knowledge, the ToD corpus for HRI in industrial robots is unavailable. Inspired by the Wizard-of-Oz framework (WOZ) [46], MultiWOZ datasets [38], [39], [40] and its succeeded validation in [15], [47], we build a dialogue corpus via human-to-human data collection method. To obtain a more reliable and diverse dialogue corpus, participants invited must have background and skills related to Robotics, Automation, or Computer Science, with an emphasis on expertise in using industrial robots. Furthermore, factory workers and engineers are also included, given that their regular tasks involve operating industrial robots. As a result, an 18-member team is created, consisting of three factory engineers, six Ph.D. students, two master’s students, and seven professors. All students have a background in Robotics/Automation and have experience in programming and manipulating industrial robots. Among seven professors, four of them work on industrial robots, one focuses on automation, and the other two have a background in computer science.

As seen in Fig. 2, the simulation environment is Aalborg University’s Learning Factory [48]. The robot utilized in the simulation is one of our autonomous industrial mobile manipulators, Little Helper (LH), currently in its eighth...
FIGURE 2. Our Little helper robot and the AAU learning factory.

generation [49]. LH combines a MiR 200 (on the bottom) and a Franka Emika Panda collaborative Robot (on the top), as seen in Fig. 2.

\subsection*{a: DIALOGUE TASK}
An ontology of a dialogue task is formed by a domain, dialogue act type, and dialogue slots. Table 1 shows the defined ontology of the IRWoZ dataset.

As mentioned earlier, four domains are identified from the two scenarios (see Appendix C).

\begin{itemize}
  \item Delivery. A transportation task, where a mobile industrial robot delivers a package.
  \item Position. A positioning task, where a mobile industrial robot edits its position/location (e.g., add a new position) on a 2D shop floor map.
  \item Assembly. A product assembly task, where an industrial manipulator assembles a requested product.
  \item Relocation. A relocation task, where an industrial manipulator assists with the relocation of an object, e.g., grasping, moving.
\end{itemize}

There are three types of dialogue acts: database-related, task-related, and general greeting. The database search conditions are set based on extracted slots if a task requires database querying. For instance, a general delivery task requires a worker to specify the destination. The ToD system should be able to verify whether or not a particular destination (i.e., location) exists in the system database. There are two types of database act: required (\textit{DB_request_req}) and optional (\textit{DB_request_opt}) (e.g., name of the recipient for a delivery task) acts. The task-related dialogue act denotes additional information that a worker needs to offer to complete a task, e.g., \textit{quantity} of an assembly task. It also includes required (\textit{T_inform_req}) and optional (\textit{T_inform_opt}) slots. The verification results of the required or optional dialogue acts are specified in the search results (i.e., search_results). Additionally, general greeting acts, like “thank you”, “goodbye”, are also provided.

Slots are required core information (e.g., product, area) extracted from a worker’s utterance to accomplish a manufacturing task. There are a total of 19 slots identified throughout the above four domains. Domains may share the slots; for example, slot \textit{color} can be used to specify the color of a product for an assembly task or the color of a package used in a delivery task.

\subsection*{b: ROLE OF PARTICIPANTS}
Human-to-human method is leveraged to collect dialogue corpus regarding the above four domains. Each participant is introduced to the LH robot and instructed on how to operate LH to perform tasks. To collect data, one participant assumes the role of an industrial robot (i.e., the wizard), while the other one acts as a shop floor worker.

The shop floor worker is asked to randomly choose a task and initiate a dialogue. By following the same approach in [38], sampled task specification (e.g., Fig. 3) crossing four domain are distributed to shop floor worker. The wizard needs to respond to the shop floor worker according to the required task. The maximum conversational depth and the number of

| Domain          | Dialogue act                                                                 |
|-----------------|-------------------------------------------------------------------------------|
| Delivery, Position, Assembly, Relocation | Delivery_DB_request_req, Delivery_DB_request_opt |
|                 | Delivery_T_inform_req, Delivery_T_inform_opt                                |
|                 | Position_DB_request_req, Position_DB_request_opt                             |
|                 | Position_T_inform_req, Position_T_inform_opt                                |
|                 | Assembly_DB_request_req, Assembly_DB_request_opt                             |
|                 | Assembly_T_inform_req, Assembly_T_inform_opt                                |
|                 | Relocation_DB_request_req, Relocation_DB_request_opt                         |
|                 | Relocation_T_inform_req, Relocation_T_inform_opt                            |
|                 | Greet                                                                        |
|                 | Search_results                                                               |

| Slots          |                                                                                   |
|----------------|-----------------------------------------------------------------------------------|
| Delivery-area, Delivery-location | Delivery-object, Delivery-color, Delivery-size |
| Delivery-sender, Delivery-recipient |                                               |
| Position-name, Position-operation |                                               |
| Assembly-producttype, Assembly-product |                                               |
| Assembly-quantity, Assembly-color, Assembly-style |                                               |
| Relocation-object, Relocation-color, Relocation-size |                                               |
| Relocation-from, Relocation-to |                                               |

\begin{itemize}
  \item You are working on an internal transportation task.
  \item You work in the lab of the AAU Smart factory.
  \item Your colleague John is expecting a package from you.
  \item John is working at the warehouse in the lab.
  \item The package is a small yellow box.
  \item You use Little Helper to deliver the package.
  \item The area (lab) and location (warehouse) should be registered in the database.
  \item The following details of the package are optional:
    \begin{itemize}
      \item recipient: John
      \item size: small
      \item Color: yellow
    \end{itemize}
\end{itemize}

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{delivery_task_template.png}
\caption{A sample task template for a delivery task.}
\end{figure}
continuous dialogue turns for a task are not limited to creating a natural dialogue environment [51]. Since individuals structure their utterances differently even when assigned the same task, the shop floor worker is encouraged to compose the sentence uniquely.

The wizard is given access to the back-end database and a list of the robot-controlling APIs. Once the shop floor worker initiates a task-related conversation, the wizard responds appropriately. If the task requires database verification, the wizard needs to provide ground truth results based on the database. They then extract task-related slots from the shop floor worker’s utterance. To construct a humanized response for the ToD, the wizards are urged to order their utterances using their own words. If preferred, wizards are also supplied with work-related small chat principles and task-related human-to-human conversation strategies.

3) SMALL TALK RESPONSE GENERATION: ARE PRINCIPLES
Unlike [28], we do not employ pre-trained models (e.g., GPT-2 [52], BlenderBot [6]) to generate candidate chit-chat responses, annotate and filter them manually. Rather than that, we collect individual human responses to the IRWoZ dataset during the conversations. The observation of human collaboration demonstrates that most tasks-related conversations may also include small talk, which can be incorporated into the ToD process to build rapport, establish trust, and increase user engagement. [53], [54], [55]. Inspired by [50], Anchor, Reveal and Encourage (ARE) principles are introduced to assist the small talk response generation. Table 2 shows the descriptions of ARE and its associated examples. Anchor facilitates mutual understanding and builds a friendly dialogue that can lead to task development. For instance, a shop floor operator may ask the mobile robot to deliver a package to the warehouse where the main storage room is located. The ToD may respond by combining task-related and small talk responses (the italic part) by saying: “Sure, I will do that. I know that place, it is quite busy area.” Establishing a trustworthy dialogue is likewise reliant on trust. Individuals trust those who share more personal information with them (i.e., Revealing). Automated assembly tasks may require the operator to instruct the robot to assemble an unfamiliar product. Therefore, instead of replying: “Sorry, I cannot.”, ToD may say: “Sorry, it is quite new to me. I have not learned how to do that yet.”. Another way to enhance user engagement is to encourage them to speak and involve them in conversation. The ToD may give a response: “Hey, how are you? Do you have a nice day?” when the operator greets the robot: “Hey robot, good morning!”.

4) TASK RELATED RESPONSE ENHANCEMENT: CONVERSATION STRATEGY
While the use of small conversation helps to increase user engagement, the primary objective of the developed ToD is to assist manufacturing tasks. Three human-to-human

#### Table 2. The “ARE” principles [50] and three conversation strategies [51] for generating humanised response with examples.

| Principle   | Definition                                                                 | Examples                                                                 | Strategy          | Definition                        | Examples                                                                 |
|-------------|-----------------------------------------------------------------------------|-------------------------------------------------------------------------|-------------------|-----------------------------------|-------------------------------------------------------------------------|
| Anchor      | the conversation with a topic that is part of the speaker’s mutual shared reality | “Yes, I know that place. It is quite a busy area.”                      | End topics with a suggestion | provide task-related suggestions | Well, I don’t know this place. Can you register it in the system first? |
| Reveal      | say something that has more information about you (using the Anchors mentioned above) | “Well, it is quite new to me. I have not learned how to do that.”         | Elicit more information | invite operator to provide more information | Can you describe a bit more of the thing you want to assemble? |
| Encourage   | invite others to speak with a question                                     | “Hey, how are you? Do you have a nice day?”                             | Clarifying         | attempt to understand what the speakers express in their messages | Do you mean you want to abort the task? |

**FIGURE 4.** Sample corpora from raw IRWoZ dataset (left) and generated delexical data (right).
conversation strategies [51], end topics with a suggestion, elicit more information and clarifying, are introduced in order to improve task-completion rate while maintaining a more humanised and task-related response. Table 2 contains the definitions of the strategies and their accompanying examples.

The ToD is expected not to give a “yes” or “no” answer for a requested task but to the mining operator’s intention to the greatest extent and provide alternative solutions to assist in completing the manufacturing task. For example, for a transportation task, the operator may accidentally give a location that is not registered in the system. The ToD may use end topics with a suggestions strategy and propose a Position task to the operator to mark the location first and say: “well, I don’t know the place. Can you register it in the system first?”. ToD should also be able to ask the operator to elicit more information if the instruction is not explicit.

For example, the operator may forget to mention the required task-related information for an assembly task and ask: “Hey robot, can you assemble ten pieces of this?”. The ToD may say: “Can you describe a bit more of the thing you want to assemble?” Clarification and confirmation are essential for task execution. Operators may alter tasks during the conversation, e.g., aborting a task-switching a task. For example, if the operator decides to abort a task during its execution, ToD needs to understand if that is what the operator wants or just a misunderstanding. Therefore, ToD should be able to use the Clarifying strategy to ask: “Sorry, do you mean you want to abort the task?”.

5) DATA STRUCTURE
To maintain high scalability, IRWoZ uses a data structure similar to that of the most popular MultiWOZ 2.0 [38]. Each dialogue is divided into two sections: the domain and the turn. The turn section includes multiple dialogue turns, including user and system utterances, belief states, and system actions. The dialogue is saved in JSON format. The left side of Fig. 4 illustrates an example of IRWoZ data. To extend the IRWoZ, the user needs to follow the current data structure and add new JSON elements which includes the desired domain, dialogue turns, database-related slots, task-related slots, and search results.

B. SYSTEM ARCHITECTURE
The proposed ToD4IR comprises three services: cognitive speech, human-robot dialogue, and robot control. Fig. 5 illustrates a high-level system architecture of ToD4IR.

1) COGNITIVE SPEECH SERVICE
The ToD4IR is designed in such a way to accept human speech as input and provide near-human voice as the output to create a natural and flexible communication environment.

Generally, most popular conversational AIs support a trigger word, e.g., Ok Google, Alexa, for activating the voice service. The service will terminate if no active human voice is detected within a certain period. Though most smart speakers adopt this interaction strategy, such interaction is not as natural as human-to-human communication. The experience of real-world human-to-human interaction demonstrates that there are no universal standards for constructing dialogues. The work-related dialogue among participants might be continuous or discrete, with a particular greeting at the beginning of the conversation. Furthermore, observation from our previous work [10] also shows that inaccurate human intent prediction is another concern, as a portion of a human’s speech may fall between two continuous voice detection periods. Therefore, the voice interface of ToD4IR is designed to keep listening to the human voice in the background rather than manually invoking the service using trigger words. Thus, ToD4IR can provide a near-human ability to participate in a conversation at any moment without requiring the user.
to employ trigger words, as long as the task-related user utterance is identified.

Two cognitive speech services of Microsoft,\(^4\) speech-to-text (STT) and text-to-speech (TTS), are leveraged to convert streamed human voice to the transcript and generate the near-human voice as a response, respectively.

2) HUMAN-ROBOT DIALOGUE SERVICE

As the core service of ToD4IR, the human-robot dialogue service (HRDS) is composed of five components: encoding dialogue context, decoding belief state, computing system action, encoding dialogue history, and decoding system response. SQLite database,\(^5\) which stores real-time production data, is leveraged to assist grounding response generation.

3) TASKS OF THE HRDS

There are four tasks, data preprocessing, belief state prediction, system actions generation, and system response generation, running on the HRDS.

a: TASK 0: IRWOZ DATASET PREPROCESSING

We define the dialogue context as follows:

\[ C = \{ U_1, Sys_1, \ldots, U_i, Sys_i \} \tag{1} \]

where \( U_i \) represents the user’s utterance at turn \( i \), and \( Sys_i \) is defined as:

\[ Sys_i = \begin{cases} Tres_i \oplus Sres_i, & \text{if } Sres_i \neq \emptyset \\ Tres_i, & \text{otherwise} \end{cases} \tag{2} \]

where \( \oplus \) denotes text concatenation, \( Tres_i \) and \( Sres_i \) represent task-related and small talk-related responses, respectively. As aforementioned, the wizard may either use a hybrid response (i.e., only \( Tres_i \), self-organized natural response which might mix small talk response with the task response) or concatenates \( Tres_i \) and \( Sres_i \) (guided by ARE principles and human-to-human conversation strategies). Dialogue belief \( B \) defines the dialogue states:

\[ B = \{ b_1, b_2, \ldots, b_i \} \tag{3} \]

where \( b_i \) denotes dialogue state at turn \( i \), and \( b_i \) is defined as:

\[ b_i = DB_{req_i} \oplus DB_{opt_i} \oplus T_{req_i} \oplus T_{opt_i} \tag{4} \]

where \( DB_{req_i} \) and \( DB_{opt_i} \) indicate the required and optional belief states for database retrieval at turn \( i \), \( T_{req_i} \), and \( T_{opt_i} \) for the required and optional task-related belief states. Both \( DB_{opt_i} \) and \( T_{opt_i} \) can be NULL if operator does not provide them. System actions, \( Sys_{act} \), define verified results from the operator’s utterance:

\[ Sys_{act_i} = DB_{SR_i} \oplus T_{SR_i} \tag{5} \]

where \( DB_{SR_i} \) and \( T_{SR_i} \) denote the generated system actions at turn \( i \) based on database search results and system response, \( Res_i \), at turn \( i \) will be the part of the dialogue context, \( Sys_{i+1} \), at turn \( i + 1 \).

We define a dialogue, \( D \), of HRDS as:

\[ D = \{ C, B, Sys_{act}, Res \} \tag{7} \]

Therefore, the training dataset, as raw IRWoZ (see the original format on the left side of Fig. 4), needs to be reconstructed and annotated by following the above dialogue structure before moving to HRDS. Table 3 defines the special tokens used to identify components of the raw text. One of the tasks of data preprocessing is to extract and annotate the raw text from IRWoZ and generate the expected dialogue dataset (see the right side of Fig. 4).

\[ p(C_i) \rightarrow p(B_i|C_i) \rightarrow p(Res_i|C_i, B_i, Sys_{act_i}) \tag{8} \]

Dialogue belief, \( b_i \), at turn \( i \) is predicted based on the dialogue context, \( \{ U_1, Sys_1, \ldots, U_{i-1}, Sys_{i-1}, U_i \} \), from turn 1 to turn \( i \) (only user utterance at turn \( i \)). The predicted belief states that \( b_i \), is a sequence of text which is formed as follows:

\[ b_i = \{ T \ b_{s_1} = b_{v_1}, \ldots, b_{s_m} = b_{v_m} \}, \tag{9} \]

where \( T \) represents the predicted domain, \( b_s \) and \( b_v \) stand for the belief slot and the slot value, respectively, and \( m \) is the total number of the predicted belief slots. \( b_s \) should belong to the set of predefined slots, \( slT \), of the desired task \( T \),

\[ slT = \{ sl_{t1}, sl_{t2}, \ldots, sl_{tn} \} \tag{10} \]

| Token                           | Description                      |
|---------------------------------|----------------------------------|
| <lendoftext>                    | indicate beginning and end of text |
| <boci>                          | indicate beginning of the context |
| <eoc>                           | indicate end of the context       |
| <bobi>                          | indicate beginning of the dialogue belief state |
| <ebob>                          | indicate end of the dialogue belief state |
| <user>                          | indicate user’s utterance         |
| <sys>                           | indicate final system response    |
| <dbreq>                         | indicate required slots for database query |
| <dbopt>                         | indicate optional slots for database query |
| <treq>                          | indicate required slots for target task |
| <topt>                          | indicate optional slots for target task |
| <bosysact>                      | indicate beginning of the system actions |
| <eosysact>                      | indicate end of the system actions |
| <bostdress>                     | indicate beginning of the task-oriented response |
| <bostdress>                     | indicate end of the task-oriented response |
| <bostdress>                     | indicate beginning of the small talk response |
| <bostdress>                     | indicate end of the small talk response |

\[ Res_i = Sys_{i+1} \tag{6} \]

\[ D = \{ C, B, Sys_{act}, Res \} \tag{7} \]
FIGURE 6. An overview of the ToD4IR model architecture including three tasks, predict belief state, generate system actions and generate system response.

\[ b_{s_j} \in sli_T, 1 \leq i \leq m \]  
(11)

where \( n \) is the total number of the task slots \( S \) and \( sli_T \) is a slot related to the database or the task. The training goal of belief prediction, \( \mathcal{L}(B_T^i) \), is defined as:

\[ \sum_{j=1}^{o} \log \theta(b_{s_j}(b_{s_1}, \ldots, b_{s_{j-1}}), C_T^j)) \]  
(12)

where \( o \) is the total number of slots of a belief state sequence, and \( \theta \) is the learning neural network parameters. If the \( sli_T \) is an optional slot (i.e., \( DB_{opt} \) or \( T_{opt} \)), it will not be predicted unless it is detected from operator’s utterance.

\( c: \text{TASK 2: GENERATE SYSTEM ACTIONS} \)

System actions, \( Sys_{act} \), include database-related actions and task-related actions. If database querying is required based on the predicted belief states, the corresponding database-related slots of belief states are extracted as query parameters. The extracted task-related slots and the database querying results are mapped to system actions (i.e., \( T_{act} \) and \( DB_{act} \)) and are defined as follows:

\[ DB_{act}, T_{act} \in \{null, detected, undetected\} \]  
(13)

where \( null \) means the operator does not provide the slot while it can be \( detected \) from the operator’s utterance (if it is a task-related slot) or matched with database query results (if it is a database-related slot). Naturally, \( undetected \) means that the slot remains undetected.

\( d: \text{TASK 3: GENERATE SYSTEM RESPONSE} \)

The delexicalized task related system response (e.g., text between "boTres" and "eoTres" of the right side of Fig. 4), and small talk response (e.g., text between "boSres" and "eoSres" of the right side of Fig. 4). The goal of training,
ToD4IR: A Humanised Task-Oriented Dialogue System for Industrial Robots

ToD4IR, like previous state-of-the-art (SoTA) works (e.g., SOLOIST [23], SimpleToD [9] and MinTL [22]), is implemented using two auto-regressive pre-trained language models, GPT-2 and GPT-Neo [56]. Those models were trained on massive volumes of open Web material and learned how to complete a sentence in a given context. Such models are largely used in downstream NLP tasks (e.g., machine translation, answering questions, text generation in our case), where they are fed a small task-specific dataset for fine-tuning the desired final model. We briefly introduce GPT-2 and GPT-Neo, which are used in this paper.

a: GPT-2
The OpenAI\footnote{https://openai.com/blog/better-language-models/} GPT-2 follows decoder only transformer [57] architecture in a self-supervised manner. It is trained with causal language modeling (CLM) with the target of generating coherent text based on the previously given text. Hugging face\footnote{https://huggingface.co/gpt2} provides five different sizes, distilgpt-2, gpt-2, gpt2-medium, gpt2-large and gpt2-xl. Different from [8], [9], [23], ToD4IR uses gpt2-large (with 36 layers and 1280M parameters) instead of gpt-2 as one of the backbone models in our case (see Fig. 6).

b: GPT-NEO
Since the latest pre-trained language model, GPT-3 (with 175B parameters), has not yet been open-sourced, EleutherAI’s GPT-Neo is leveraged as our second backbone model. GPT-Neo is an open-source transformer model which replicates the GPT-3 architecture. It is trained on the Pile [58] dataset, which has an 825GB English text corpus. Evaluation results of linguistic reasoning and physical and scientific reasoning show the GPT-Neo (with 2.7B parameters) resembles GPT-3 in performance.\footnote{https://github.com/EleutherAI/gpt-neo}

5) ROBOT CONTROL SERVICE
Based on our previous work [10], ToD4IR is designed to be agnostic of robot hardware. The robot control service is implemented as a pair of components: a robot service management (RSM) and a robot service execution (RSE). The output of the HRDS (i.e., task-related information and commands) will be sent to RSE to ground language instructions into robot actions, which invokes robot control functionalities (e.g., package delivery) via communication protocols (e.g., TCP/IP). RSM periodically synchronizes, once per week by default, the local robot services and skills with the ToD4IR server to update the robot services/skills registered on the local client-side and the robot controlling scripts.

IV. EXPERIMENTS AND DISCUSSIONS
This section evaluates the proposed approach to address two research questions: Q1: Given that ToD4IR is a conversational AI for industrial robots, how does ToD4IR perform tasks within the four defined domains? Q2: Is the ToD4IR able to augment user experience through embedded small talk?

A. EXPERIMENTAL SETUP
We trained and evaluated our ToD4IR on Aalborg University Cloud, where each server is configured with Intel(R) Xeon(R) Gold 5118 (12 cores), 256GB of memory, and Nvidia V100 with 32GB VRAM. The ToD4IR is implemented based on HuggingFace Transformers 4.7.0,\footnote{https://huggingface.co/docs/transformers/index} Microsoft DeepSpeed 0.4.0\footnote{https://github.com/microsoft/DeepSpeed} and Torch 1.7.\footnote{https://pytorch.org/get-started/locally/}
TABLE 4. The components of the annotated dialogue corpus of IRWoZ dataset. ToD4IR trains on a sequence of such annotated dialogue corpus.

| Component       | Description                                                                 | Example                                                                 |
|-----------------|-----------------------------------------------------------------------------|-------------------------------------------------------------------------|
| Dialogue Context| It includes user utterance and system response.                             | \{oe\} <user> could you deliver the box for me please? <sys> sure, where do you want to deliver it? <user> yes, it is the office in the lab. <oe> |
| Belief State    | It contains database-related slots, task-related slots (including database and task-related), and the user’s dialogue act. | \{oe\} <DB Req> delivery area=lab location=office <T Req> delivery object=box <oe> |
| System Actions  | It comprises the database search results and task-related slots extraction. | \{osys Act\} delivery area=detected location=detected object=detected <eosys Act> |
| Task response   | It responds to the user’s spoken command.                                  | \{oeo\} ok, i found location [location] in the [area] . \{eoe\} |
| Small Talk response | It is a casual style of discourse that omits any necessary functional themes of a task. | \{oeos\} i am so happy! \{eos\} |

1) PREPARATION OF DATASETS
While the WoZ method enables a more flexible and straightforward generation of low-noise dialogue corpora, the time-consuming data collection and annotation processes need high accuracy. The primary difficulty lies in checking whether the appropriate slots are given, validating whether the slots match database search results, and manually annotating slots. Fig. 4 shows an example of raw IRWoZ dialogue and the respective annotations.

To address this issue, a Flask-based 12 IRWoZ web application is constructed. The application features two interfaces, one for the user and one for the Wizard. The user submits their utterance using the online form, and the Wizard automatically detects and checks the extracted dialogue acts (e.g., delivery, position) via the web form. The Wizard is given two distinct input text fields to deliver task-related and small-talk-related responses. When the user confirms the chat is complete, the web application automatically annotates and saves the conversations. We provide an example of the dialogue collection and annotation processes using the web application developed in the appendix B.

Eighteen individuals have been asked to participate in the gathering procedure for the IRWoZ dataset. The dataset contains 158/42/101/100 dialogues corpus span over assembly/delivery/position/relocation (see Fig. 7), with each dialogue having at least two turns and augmenting over 88.7% of system responses with small talk. Table 4 summarizes the essential components of the annotated dialogue with examples. To evaluate the proposed ToD4IR on our IRWoZ dataset, we divide it into 60%, 20%, and 20% for training, validation, and testing, respectively.

2) AUTOMATIC EVALUATION
To respond to the first research question, we train our ToD4IR based on GPT architecture with five pre-trained language models: gpt2, gpt2-large, gpt2-xl, GPT-Neo (1.3B), and GPT-Neo (2.7B). The ToD4IR follows the end-to-end dialogue pattern in which the evaluation mainly involves two aspects, dialogue state tracking, and system actions and response generation [38].

TABLE 5. Evaluation of dialogue state tracking on IRWoZ using joint accuracy and slot accuracy metrics.

| Model               | Joint Goal Accuracy | Slot Accuracy |
|---------------------|----------------------|---------------|
| ToD4IR-gpt2         | 0.789                | 0.941         |
| ToD4IR-gpt2-large   | 0.856                | 0.959         |
| ToD4IR-gpt2-xl      | 0.861                | 0.964         |
| ToD4IR-gpt neo (1.3B)| 0.845               | 0.957         |
| ToD4IR-gpt neo (2.7B)| 0.866               | 0.963         |

In order to evaluate the performance of ToD4IR, we used three metrics.

- Joint goal accuracy. The output of the dialogue state tracker is compared to the ground truth label at the end of each discourse. The proportion of dialogue turns in which the value of each slot is correctly predicted is known as the joint goal accuracy.
- Slot accuracy. It compares each (domain, slot, value) triplet with the corresponding ground-truth label. Compared with the joint goal accuracy, its evaluation granularity is more refined.
- Bilingual evaluation understudy (BLEU) [59]. It is mainly used for measuring the fluency of the generated text.

Among the above three evaluation metrics, joint goal accuracy and slot accuracy are commonly used in dialogue state tracking tasks, and BLEU is leveraged for response generation.

a: DIALOGUE STATE TRACKING
This task aims to assess ToD4IR’s ability to predict dialogue state in the given dialogue context, which includes domain, slot, and value. Table 5 compares the ToD4IR’s joint objective and slot accuracy to that of various backbone models on the IRWoZ dataset. ToD4IR-gpt2-xl outperforms other models at 0.964 of slot accuracy, whereas ToD4IR-gpt2-neo (2.7B) outperforms at 0.866 of joint goal accuracy.

b: SYSTEM ACTIONS AND RESPONSE GENERATION
In this task, ToD4IR should generate system actions and responses given the ground truth dialogue states and database search results. Due to the WoZ data collection method, the ground truth of database search results is manually incorporated into the system actions of the IRWoZ dataset.

\[12\] https://flask.palletsprojects.com/en/2.0.x/
As a result, ToD4IR eliminates the requirement for database searches during the training and validation processes. Table 6 compares the ToD4IR’s BLEU 1-4 scores on the IRWoZ dataset’s five models. At 0.6013, 0.5349, 0.5032, and 0.4763, ToD4IR-gpt2-large outperforms other models.

3) HUMAN EVALUATIONS

Human assessments provide a complete picture of response-generating performance, particularly in the generation of human-like response [60]. We leverage the human evaluation questions from [8], which cover engaging, interesting, human-like, and knowledgeable, for response generation assessments.

As mentioned, 20% dialogue corpus is selected as the test data set. We feed those data samples to the five models, ToD4IR-gpt2, ToD4IR-gpt2-large, ToD4IR-gpt2-xl, ToD4IR-gpt-neo (1.3B) and ToD4IR-gpt-neo(2.7B) to obtain the responses. Task-related responses and small talk responses are highlighted in each of the dialogues.

To verify the validity of the assessment findings, domain specialists within robotics, computer science, linguistics, and humanities and various end-users such as shop floor workers and lab engineers were invited to serve as evaluators. The evaluators are asked to rate their answers on a predetermined scale, where each number on the scale denotes a different quality level, ranging from Not at all to Absolutely. For instance, the inquiry How much would you prefer to talk to the ToD4IR? is used to determine whether the user is engaged and willing to speak with the ToD4IR.

In some circumstances, particularly when evaluating the engaging and knowing competence, the dialogue context is essential to evaluate the generated system response. However, the reference context is not required to evaluate fluency (without grammatical errors), interest, and human likeness. We give context for each generated response in our evaluation.

We showcase each model’s individual score, in Table 7, and summarize how engaging, interesting, human-like, and knowledgeable the dialogue was. Table 8 reports the overall score of ToD4IR.

| TABLE 6. Context-to-response evaluation on IRWoZ. |
| --- | --- | --- | --- | --- |
| Model | BLEU-1 | BLEU-2 | BLEU-3 | BLEU-4 |
| ToD4IR-gpt2 | 0.4513 | 0.3852 | 0.3386 | 0.2968 |
| ToD4IR-gpt2-large | 0.6013 | 0.5349 | 0.5032 | 0.4763 |
| ToD4IR-gpt2-xl | 0.5709 | 0.5059 | 0.4727 | 0.4432 |
| ToD4IR-gpt-neo (1.3B) | 0.5183 | 0.4410 | 0.4001 | 0.3645 |
| ToD4IR-gpt-neo (2.7B) | 0.5599 | 0.4853 | 0.4469 | 0.4140 |

| TABLE 7. Human evaluation results of the generated response of ToD4IR. |
| --- | --- | --- | --- | --- |
| Model | Engaging | Interesting | Humanlike | Knowledgeable |
| ToD4IR-gpt2 | 50% | 25% | 37.5% | 37.5% |
| ToD4IR-gpt2-large | 12.5% | - | 12.5% | - |
| ToD4IR-gpt2-xl | 12.5% | 50% | - | 12.5% |
| ToD4IR-gpt-neo (1.3B) | - | - | 12.5% | - |
| ToD4IR-gpt-neo (2.7B) | 25% | 25% | 37.5% | 50% |

| TABLE 8. Human evaluation results of overall engaging, interesting, human-like, and knowledgeable score. |
| --- | --- | --- | --- |
| Score | Engaging | Interesting | Humanlike | Knowledgeable |
| 8.0 | 8.5 | 7.0 | 8 |

interaction in the manufacturing sector, it is a small-scale dialogue corpus with limited domain coverage. Additionally, an imbalanced data distribution (i.e., delivery occupies only 11% of the conversation corpus) affects ToD4IR’s performance on the delivery job.

2) NOISE-LABELS

In comparison to other datasets, such as MultiWoZ 2.0, IRWoZ contains far fewer mis-annotations, which adds to the high accuracy of dialogue state tracking (see table 5). The first reason is that the IRWoZ is compiled by 18 individuals, including students, academics, and shop floor workers with backgrounds ranging from robotics to computer science. As a result, the corpora of dialogue collected are significantly more pure and professorial. Second, a web application (source code can be found on our Github) is created to support the data collection. The user may directly engage with the application with the four domain tasks. The application’s back end automatically annotated and verified each dialogue, including annotations and database search results. Additionally, human verification is performed as the last step, ensuring that the collected dialogue corpora contain fewer mis-annotations than other datasets.

3) DIALOGUE RESPONSE

One of the objectives of our method is to train the ToD4IR to generate more natural and human-like responses. As a result, natural human responses are expected during the dialogue simulation. However, examining the collected data, we realize that when users communicate with the system, they use a less complicated version of their language. For instance, a simple response of “Ok.” is typically used when they affirm a system-generated inquiry, whereas “Yes, I believe so.” or “yes, you are correct.” is frequently used when they speak with a human. The participants confirm this trend during the collection of data. The other observation is that ten out of the eighteen persons prefer to structure their small talk responses instead of directly using the ARE principles.

4) PRE-TRAINED MODELS

The ToD4IR is mainly driven by gpt2-large and gpt-neo(2.7B) models with pre-trained weights. Each model is...
FIGURE 8. Identified four domains for ToD4IR.
FIGURE 9. Web application for IRWoZ dialogue corpus collection.
TABLE 9. ToD4IR end-to-end response generation on the test set of IRWoZ. We show the generated task-related and small talk-related responses by different models.

| Example 1 | Context | Task Response | Small Talk Response |
|-----------|---------|---------------|---------------------|
| Model ToD4IR-gpt2 | Task Response | Sure. What kind of them? | Hi guy! Can you assemble 270 legos for me please? |
| Model ToD4IR-gpt2-large | Task Response | No problem. What kind of lego? | |
| Model ToD4IR-gpt-xl | Task Response | No problem. | |
| Model ToD4IR-gpt-neo(1.3B) | Task Response | No problem. What kind of lego? | |
| Model ToD4IR-gpt-neo(2.7B) | Task Response | No problem. | |

Example 2: I need you to run down the workbench.

| Model ToD4IR-gpt2 | Task Response | Yes. I knew that place. | I have been to that place many times. |
| Model ToD4IR-gpt2-large | Task Response | Yes, I have been to that place many times. | |
| Model ToD4IR-gpt-xl | Task Response | No problem. | |
| Model ToD4IR-gpt-neo(1.3B) | Task Response | Yes. | |
| Model ToD4IR-gpt-neo(2.7B) | Task Response | Sure. | |

Example 3: Let’s go to the maker space, robot close-by well, where is that? it is quite a new place to me

| Model ToD4IR-gpt2 | Task Response | Sure. | I am glad to help |
| Model ToD4IR-gpt2-large | Task Response | Sure, let me check it out. | |
| Model ToD4IR-gpt-xl | Task Response | No. It is beyond my ability. | |
| Model ToD4IR-gpt-neo(1.3B) | Task Response | No. it is not possible. | |
| Model ToD4IR-gpt-neo(2.7B) | Task Response | Sure. | |

Example 4: Can you reach the injection molding machine

| Model ToD4IR-gpt2 | Task Response | Yes, I can not. | It is a difficult task for me |
| Model ToD4IR-gpt2-large | Task Response | No. This is beyond my ability. | |
| Model ToD4IR-gpt-xl | Task Response | No. It is not possible. | |
| Model ToD4IR-gpt-neo(1.3B) | Task Response | No. It is quiet new to me. | |
| Model ToD4IR-gpt-neo(2.7B) | Task Response | Sure. | |

Example 5: Hi friend! can you assemble eight blue phones? Of course. When do you want them? Look, I still have one parcel at hand! Maybe you need to wait.

| Model ToD4IR-gpt2 | Task Response | All right. I will finish them in the afternoon. | Trust me! |
| Model ToD4IR-gpt2-large | Task Response | I must finish them before that. | |
| Model ToD4IR-gpt-xl | Task Response | I must finish them before that. | |
| Model ToD4IR-gpt-neo(1.3B) | Task Response | All right. | |
| Model ToD4IR-gpt-neo(2.7B) | Task Response | Sure. I must finish them before that. | |

Example 6: OK, have you been to the lego smart production line?

| Model ToD4IR-gpt2 | Task Response | Well, I do not know that. Can you register it in the system first? | |
| Model ToD4IR-gpt2-large | Task Response | No, I have not been to that place. | |
| Model ToD4IR-gpt-xl | Task Response | No, I do not know that place. | |
| Model ToD4IR-gpt-neo(1.3B) | Task Response | No, can you register it in the system first? | |
| Model ToD4IR-gpt-neo(2.7B) | Task Response | No, but I have not been to that place. | |

trained for 20 epochs. However, the largest model, ToD4IR-gpt-neo(2.7B), only surpasses ToD4IR-gpt2-large by a factor of 0.01 on joint goal accuracy while around 0.05 less on BLEU 1-4. Additionally, we compare the ToD4IR against three other gpt type models. The overall evaluation results indicate that shallow neural networks outperform deep
neural networks when ToD4IR is trained on a small-scale dataset.

V. CONCLUSION

In this study, we present ToD4IR, a humanized task-oriented dialogue system aimed at industrial robots. The first industrial-oriented dialogue corpus, IRWoZ, is constructed with 401 dialogues spanning four industrial tasks: delivery, assembly, position, and relocation. To aid ToD4IR in generating natural and humanized responses, we use small talk principles, known as ARE, along with human-to-human conversation strategies. Driven by a task-based neural network GPT, ToD4IR can predict the dialogue state and generate system actions using real-time database search results.
Additionally, it can generate work completion and user experience enhancement responses that include task-related (based on user goals) and small talk-related responses. Experiments demonstrate that ToD4IR achieves high accuracy in the dialogue state tracking and fluency in the generated response.

We hope that the proposed IRWoZ will inspire the dialogue research community and industrial partners to continue investigating language-assisted human-robot interaction in manufacturing, contributing dialog corpora for new industrial domains, and fine-tune pre-trained pre-trained ToD4IR for new industrial tasks.

We intend to conduct extensive user research in the future to gather input on the naturalness and coherence of ToD4IR-generated dialogue and responses when COVID-19 constraints are removed.

**APPENDIX A**

**IDENTIFIED FOUR DOMAINS FOR ToD4IR**

Fig. 8 shows the identified four domains, position, delivery, assembly, and relocation, and the corresponding scenario constraints are removed.

**APPENDIX B**

**WEB APPLICATION - IRWoZ DATA COLLECTION**

Fig. 9 displays the designed web interface for collecting dialogue corpora. One operator uses User Mode to start the dialogue based on the task specification. The other operator takes on the role of Wizard and responds to inquiries in System mode.

**APPENDIX C**

**GENERATED EXAMPLES**

**A. GENERATED EXAMPLES BY DIFFERENT MODELS**

Table 9 shows six examples of generated responses by ToD4IR based on five models. The response is generated (based on user goals) and small talk-related responses. Experience enhancement responses that include task-related and work completion and user experience enhancement responses.

**B. WEB INTERFACE FOR ToD4IR**

Fig. 10 shows the end-to-end response generation examples by ToD4IR-gpt2-large.

**REFERENCES**

[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” 2018, arXiv:1810.04805.

[2] W. Lei, X. Jin, M.-Y. Kan, Z. Ren, X. He, and D. Yin, “Sequicity: Simplifying task-oriented dialogue systems with single sequence-to-sequence architectures,” in Proc. 56th Annu. Meeting Assoc. Comput. Linguistics, Melbourne, VIC, Australia: Association for Computational Linguistics, Jul. 2018, pp. 1437–1447. [Online]. Available: https://aclanthology.org/P18-1133

[3] S. Akasaki and N. Kaji, “Chat detection in an intelligent assistant: Combining task-oriented and Non-task-oriented spoken dialogue systems,” in Proc. 55th Annu. Meeting Assoc. Comput. Linguistics, Vancouver, BC, Canada: Association for Computational Linguistics, Jul. 2017, pp. 1308–1319. [Online]. Available: https://aclanthology.org/P17-1120

[4] S. Roller, E. Dinan, N. Goyal, D. Ju, M. Williamson, Y. Liu, J. Xu, M. Ott, E. M. Smith, Y.-L. Boureau, and J. Weston, “Recipes for building an open-domain chatbot,” in Proc. EACL, 2021, pp. 300–325.

[5] H. Rashkin, E. M. Smith, M. Li, and Y.-L. Boureau, “Towards empathetic open-domain conversation models: A new benchmark and dataset,” in Proc. 57th Annu. Meeting Assoc. Comput. Linguistics, Jan. 2019, pp. 5370–5381.

[6] S. Roller, E. Dinan, N. Goyal, D. Ju, M. Williamson, Y. Liu, J. Xu, M. Ott, K. Shuster, E. M. Smith, Y.-L. Boureau, and J. Weston, “Recipes for building an open-domain chatbot,” 2020, arXiv:2004.13637.

[7] L. Gao, S. Biderman, S. Black, L. Golding, T. Hoppe, C. Foster, J. Pfang, H. He, A. Thate, N. Nabeshima, S. Presser, and C. Leaby, “The pile: An 800GB dataset of diverse text for language modeling,” 2021, arXiv:2101.00027.

[8] K. Sun, S. Moon, P. Crook, S. Roller, B. Silvert, B. Liu, Z. Wang, H. Liu, E. Cho, and C. Cardie, “Adding chit-chat to enhance task-oriented dialogues,” 2020, arXiv:2010.12757.

[9] E. Hosseini-Asl, B. Mccann, C. S. Wu, S. Yavuz, and R. Socher, “A simple language model for task-oriented dialogue,” in Proc. Adv. Neural Inf. Process. Syst., 2020, pp. 20179–20191.

[10] C. Li, J. Park, H. Kim, and D. Chrystostomou, “How can I help you? An intelligent virtual assistant for industrial robots,” in Proc. Companion ACM/IEEE Int. Conf. Hum.-Robot Interact. New York, NY, USA: Association for Computing Machinery, Mar. 2021, pp. 220–224, doi: 10.1145/3434074.3447163.

[11] G. Li and H. J. Yang, “Bot-X: An AI-based virtual assistant for intelligent manufacturing,” Multiagent Grid Syst., vol. 17, no. 1, pp. 1–14, Apr. 2021.

[12] D. Evangelista, W. Villa, M. Imperoli, A. Vanzo, L. Iocchi, D. Nardi, and A. Prett, “Grounding natural language instructions in industrial robotics,” in Proc. IEEE/RSJ IROS Workshop, Hum.-Robot Interact. Collaborative Manuf. Environ., Nov. 2017, pp. 1–6.

[13] J. Jungbluth, R. Krieger, W. Gerke, and F. W. Plapper, “Combining virtual and robot assistants—A case study about integrating Amazon’s Alexa as a voice interface in robotics,” in Proc. Robotics-Acad. Conf. Ind. Robot., 2018, p. 5.

[14] Z. Zhang, R. Takanobu, Q. Zhu, M. Huang, and X. Zhu, “Recent advances and challenges in task-oriented dialog systems,” Sci. China Technol. Sci., vol. 63, no. 10, pp. 2011–2027, Oct. 2020.

[15] T.-H. Wen, D. Vandyke, N. Meksis, M. Gasic, L. M. Rojas-Barahona, P.-H. Su, S. Ultes, and S. Young, “A network-based end-to-end trainable task-oriented dialogue system,” 2016, arXiv:1604.04562.

[16] H. Chen, X. Liu, D. Yin, and J. Tang, “A survey on dialogue systems: Recent advances and new frontiers,” ACM SIGKDD Explor. Newslett., vol. 19, no. 2, pp. 25–35, Dec. 2017, doi: 10.1145/3166054.3166058.

[17] X. Li, Y.-N. Chen, L. Li, J. Gao, and A. Celikyilmaz, “End-to-end task-completion neutral dialogue systems,” in Proc. 8th Int. Joint Conf. Natural Lang. Process., 2021, pp. 733–743.

[18] J. Perez and F. Liu, “Dialog state tracking, a machine reading approach using memory network,” 2016, arXiv:1606.04052.

[19] Q. Wu, Y. Zhang, Y. Li, and Z. Yu, “Alternating recurrent dialog model with large-scale pre-trained language models,” 2019, arXiv:1910.03756.

[20] L. Chen, B. Lv, C. Wang, S. Zhu, B. Tan, and K. Yu, “Schema-guided multi-domain dialogue state tracking with graph attention neural networks,” in Proc. AAAI, 2020, pp. 7521–7528.

[21] C.-S. Wu, S. Hoi, R. Socher, and C. Xiong, “TOD-BERT: Pre-trained natural language understanding for task-oriented dialogue,” 2020, arXiv:2004.06871.

[22] Z. Lin, A. Madotto, G. I. Winata, and P. Fung, “MinTL: Minimalist transfer learning for task-oriented dialogue systems,” 2020, arXiv:2009.12005.

[23] B. Peng, C. Li, J. Li, S. Shayandeh, L. Liden, and J. Gao, “SOLQIST: Building task bots at scale with transfer learning and machine teaching,” 2020, arXiv:2005.05298.

[24] B. Liu, G. Tür, D. Hakkani-Tür, P. Shah, and L. Heek, “Dialogue learning with human teaching and feedback in End-to-End trainable task-oriented dialogue systems,” in Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol. New Orleans, LA, USA: Association for Computational Linguistics, Jun. 2018, pp. 2060–2069. [Online]. Available: https://aclanthology.org/N18-1187

[25] B. Dhirenga, L. Li, X. Li, J. Gao, Y.-N. Chen, F. Ahmed, and L. Deng, “Towards end-to-end reinforcement learning of dialogue agents for information access,” in Proc. 55th Annu. Meeting Assoc. Comput. Linguistics, Vancouver, BC, Canada: Association for Computational Linguistics, Jul. 2017, pp. 484–495. [Online]. Available: https://aclanthology.org/P17-1045
T. Wolf, V. Sanh, J. Chaumond, and C. Delangue, “Transferto: A transfer learning approach for neural network based conversational agents,” CoRR, vol. abs/1901.08149, pp. 1–6, Jan. 2019.

R. Lowe, N. Pow, I. Serban, and J. Pineau, “The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems,” in Proc. 16th Annu. Meeting Special Interest Group Discourse Dialogue. Prague, Czech Republic: Association for Computational Linguistics, Sep. 2015, pp. 285–294. [Online]. Available: https://aclanthology.org/W15-4640

S. Bög, M. Hvilshøj, M. Kristiansen, and O. Madsen, “Identifying and evaluating suitable tasks for autonomous industrial mobile manipulators (AIMM),” Int. J. Adv. Manuf. Technol., vol. 61, nos. 5–8, pp. 713–726, Jul. 2012.

J. F. Kelley, “An iterative design methodology for user-friendly natural language office information applications,” ACM Trans. Inf. Syst., vol. 2, no. 1, pp. 26–41, Jan. 1984, doi: 10.1145/357471.357420.

L. El Asri, H. Schulz, S. Sharma, J. Zumer, J. Harris, E. Fine, R. Mehrotra, and K. Suleman, “Frames: A corpus for adding memory to goal-oriented dialogue systems,” 2017, arXiv:1704.00057.

M. Nardello, O. Madsen, and C. Möller, “The smart production laboratory: A learning factory for industry 4.0 concepts,” in Proc. CEUR Workshop, vol. 1899, 2017, pp. 1–6.

C. Schou, R. S. Andersen, D. Chrysostomou, S. Bög, and O. Madsen, “Skill-based instruction of collaborative robots in industrial settings,” Robot. Comput.-Integr. Manuf., vol. 53, pp. 72–80, Oct. 2018, doi: 10.1016/j.rcim.2018.05.008.

C. Fleming, It’s the Way You Say It: Becoming Articulate, Well-Spoken, and Clear, San Francisco, CA, USA: Berrett-Koehler Publishers, 2013.

Z. Yu, Z. Xu, A. W. Black, and A. Rudnicky, “Strategy and policy learning for non-task-oriented conversational systems,” in Proc. 17th Annu. Meet- ing Special Interest Group Discourse Dialogue, 2016, pp. 404–412.

A. Radford, J. Wu, R. Child, D. Luan, A. Amodei, and I. Sutskever, “Language models are unsupervised multitask learners,” OpenAI Blog, vol. 1, no. 8, p. 9, 2019.

T. Bickmore and J. Cassell, “Small talk and conversational storytelling in embodied conversational interface agents,” in Proc. AAAI Fall Symp. Narrative Intel., 1999, pp. 87–92.

T. Küürer, “I like your shirt”-dialogue acts for enabling social talk in conversational agents,” in Proc. Int. Workshop on Intelligent Virtual Agents. Berlin, Germany: Springer, 2011, pp. 14–27.

E. Hoffman, Forms of Talk. Philadelphia, PA, USA: Univ. of Pennsylvania Press, 1981.

S. Black, G. Leo, P. Wang, C. Leahy, and S. Biderman, “GPT-Neo: Large scale autoregressive language modeling with mesh-TensorFlow,” Zenodo, Eur. Org. Nucl. Res., Cern, Switzerland, Tech. Rep., Mar. 2021, doi: 10.5281/zenodo.5297715.

K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “BLEU: A method for automatic evaluation of machine translation,” in Proc. 40th Annu. Meeting Assoc. Comput. Linguistics, 2001, pp. 311–318.

A. B. Sai, A. K. Mohankumar, and M. M. Khapra, “A survey of evaluation metrics used for NLG systems,” ACM Comput. Surv., vol. 55, no. 2, pp. 1–39, Mar. 2023, doi: 10.1145/3485766.

C. Li et al.: ToD4IR: A Humanised Task-Oriented Dialogue System for Industrial Robots
XIAOCHUN ZHANG (Member, IEEE) received the Ph.D. degree from the School of Computer Science and Technology, Nanjing University of Science and Technology, in 2014. She is currently working with the School of Management Science and Computer, Anhui University of Finance and Economics, China. Her research interests include computer vision, time series prediction, and dialogue systems.

DIMITRIOS CHRYSOSTOMOU (Member, IEEE) received the Diploma degree in production engineering and the Ph.D. degree in robot vision from the Democritus University of Thrace, Greece, in 2006 and 2013, respectively. Since 2013, he has been working with the Robotics and Automation Group, Department of Materials and Production, Aalborg University, Denmark, as a Postdoctoral Researcher, from 2013 to 2016; an Assistant Professor, from 2016 to 2019; and since 2020, as an Associate Professor. His research interests include robot vision, skill-based programming, and human–robot interaction for intelligent robot assistants.

HONGJI YANG received the B.S. and M.S. degrees in computer science from Jilin University, Changchun, China, in 1982 and 1985, respectively, and the Ph.D. degree in computer science from Durham University, Durham, U.K., in 1994. He is currently working at the School of Computing and Mathematical Sciences, University of Leicester. He has published 500 papers. His main research interests include knowledge modeling and creative computing. In 2010, he became a Golden Core Member of the IEEE Computer Society.