**Abstract :** During the past decade, several areas of speech and language understanding have witnessed substantial breakthroughs from the use of data-driven models. In the area of dialogue systems, the trend is less obvious, and most practical systems are still built through significant engineering and expert knowledge. Nevertheless, several recent results suggest that data-driven approaches are feasible and quite promising. To facilitate research in this area, we have carried out a wide survey of publicly available datasets suitable for data-driven learning of dialogue systems. We discuss important characteristics of these datasets, how they can be used to learn diverse dialogue strategies, and their other potential uses. We also examine methods for transfer learning between datasets and the use of external knowledge. Finally, we discuss appropriate choice of evaluation metrics for the learning objective.

1. **Introduction:**

Dialogue systems, also known as interactive conversational agents, virtual agents or sometimes chatterbots, are useful in a wide range of applications ranging from technical support services to language learning tools and entertainment. Large scale data-driven methods, which use recorded data to automatically infer knowledge and strategies, are becoming increasingly important in speech and language understanding and generation. A wide range of data driven methods have shown to be effective for tasks involved in dialog systems. We hypothesize that, in general, much of the recent progress is due to the availability of large public datasets, increased computing power, and new machine learning models, such as neural network architectures. To facilitate further research on building data-driven dialogue systems, this paper presents a broad survey of the goal oriented dialog systems that has been implemented.

1. **Approaches:**

**Online Approach:** Training Dialog Systems through the online approach happens through live interaction with humans

**Offline Approach:** Training Dialog Systems through the offline approach happens through reinforcement learning methods

**Overview of Goal Oriented Dialogue Systems:**

In the past decade, goal-oriented dialogue systems have been the most prominent component in today’s virtual personal assistants, which allow users to speak naturally in order to finish tasks more efficiently. Practical dialogue systems consists of several components.

**NLU Module:** Maps free texts to structured semantic frames of utterances . The module is trained to predict the intent of the user utterances. Training this model requires examples of pairs of user utterances and intentions. The dataset used are written dialogues between humans who have carried out a task and annotated each utterance with its intention label.

**NLG Module:** Maps the structured representations back into a natural-language form. Given the user’s dialogue actions, the NLG module generates natural language texts. To control the quality of user simulation given limited labeled data, a hybrid approach including a template-based NLG and a model-based NLG is employed, where the model-based NLG is trained on the labeled dataset with a sequence-to-sequence model. It takes dialogue acts as input, and generates sentence sketch with slot placeholders via an LSTM decoder. Then a post-processing scan is performed to replace the slot placeholders with their actual values (Wen et al., 2015). In the LSTM decoder, we apply beam search, which iteratively considers the top k best sub-sentences when generating the next token.

**Language Understanding:**

The Language Understanding module consists of LSTM that performs LSTM and also fills the slot simultaneously. For example, for a question” find all the action movies available this weekend”.The find\_movie goes into the request slot.

**Dialog Manager:**

DM keeps monitoring the belief distribution over all possible user states underlying current user behaviors, and predicts responsive system actions. For example, given a user utterance “any action movies recommended this weekend?”, LU predicts intent request movie and slots genre and date; thereafter, DM predicts system action request location. The state tracker will be updated based on the available result from the database and the latest user dialogue action. The DM includes Dialog state tracking where the state of the dialog is represented and policy learning. Policy Learning represents the next available system action based on the value of the state tracker.

**Discussion:**

Next, we will discuss about the recent research papers that have been carried out on goal oriented dialog systems .

1. **End-to-End Task-Completion Neural Dialogue Systems – Microsoft**

**Summary:**

The objective of this project is to explore neural network architectures for question answering and goal oriented dialog systems. This paper presents a task oriented dialogue system which leverages supervised and reinforcement learning with various deep-learning models. We consider a dialogue system for helping users to book movie tickets or to look up the movies they want, by interacting with them in natural language. In this paper. The system interacts with the database and interacts with the users for helping them access information about movies and performing a task such as booking a movie ticket. The environment then assesses a binary outcome (success or failure) at the end of the conversation, based on (1) whether a movie is booked, and (2) whether the movie satisfies the user’s constraints.

**Dataset :** Movie knowledge base dataset which has the details of all the available movies, show timing and all the theaters nearby.

**Architecture:**

1. **User Simulator**: In the task oriented dialog system, the first step is to simulate a user goal where the agent knows nothing about the goal but the objective is to help the user accomplish the goal. The mechanism is to extract all the slots and aggregate them into one user goal. We dump these goals into a file as the user goal database for the simulator. The user simulator converts the movie database into a user goal template with structured conversations. There are three stages for a dialogue: no\_outcome\_yet, success and failure. The stage is no\_outcome\_yet if the agent has not issued the inform(taskcomplete) action and if the number of turns of the conversation has not exceeded the maximum value; otherwise, the dialogue is finished with either a success or a failure outcome. To be a success dialogue, the agent must answer all the questions (a.k.a. requestable slots of the user) and book the right movie tickets finally, within the maximum number of turns. All other cases are failure dialogues. For example, the whole dialogue exceeds the limit of max turns, or the agent books the wrong movie tickets for the user.
2. **NLU Component:** The natural language understanding (NLU) component is a recurrent neural network model with long-short term memory (LSTM) cells. This single NLU model [7] can do intent prediction, and slot filling simultaneously.

**Tasks of a Dialog System:**

Dialogue systems have been built for a wide range of purposes. So many non goal oriented dialog systems were built in the early 90s . However, work on goal oriented dialog systems started with the SUNDIAL project which was capable of providing the timetable information about trains and airplanes. However, after the advent of machine learning, those techniques were used to classify the intention of the user as well as to bridge the gap between the text and speech. Large number of datasets(tasks specific) have started to come in place which has become a lot easier for training the dialog system .

**Reinforcement Learning:** Reinforcement Learning comes in place where we general the optimal action by learning the dialog policy. Reinforcement Learning is applied to the dialog manager where it is trained using the deep Q network. The Deep Q network takes the input as the state tracker and gives the Q value for all the possible actions available. The epsilon greedy exploration policy is used . Q values will be updated and In the training process, at every simulation epoch, we estimate the success rate of the current DQN agent.