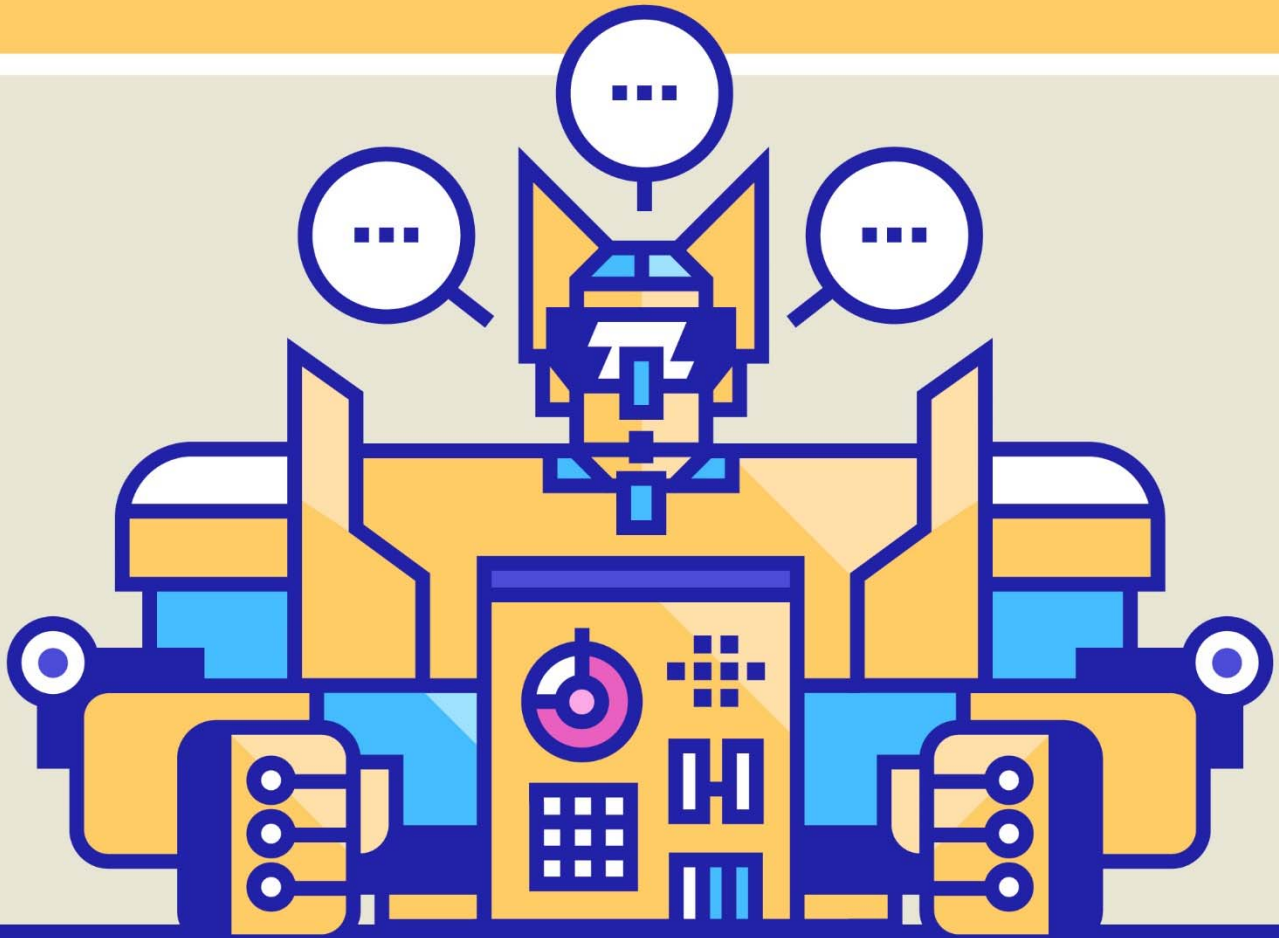
DIALOG SYSTEMS



sas
Specialty Answering Service

WWW.SPECIALTYANSWERINGSERVICE.NET

Contents

1 Introduction 2

1.1 Types of Dialog Systems 2

2 Dialog Systems in Contact Centers 2

2.1 Automated Call Centers 2

3 History 3

4 Designing Interactive Dialogues With Structured Data 4

5 Task Based Spoken Dialog Systems 4

5.1 Components of Spoken Dialog Systems 4

6 Dialog Management 6

6.1 Role of Dialog Management 6

6.2 Degrees of Initiative 6

6.3 Error Handling 7

7 Modern Approaches to Dialog Management 8

7.1 Knowledge-based Dialog Management 8

7.2 Data-driven Dialog Management 9

7.3 Hybrid Approaches to Dialog Management 9

8 Evaluation of Spoken Dialog Systems 9

8.1 User Intention Simulation 9

8.2 User Utterance Simulation 10

8.3 User Simulation Criteria 10

9 Summary 10

10 References 11

1 Introduction

A dialog system is an automated computer based system that can converse with a human being, similar to how another human being would converse. Dialog systems can be designed in several ways with different components performing different functions.

1.1 Types of Dialog Systems

Dialog systems can be classified into various categories based on different dimensions. They can be classified based on modality, by device, by style or by initiative as follows:

- **By modality**: text-based system, spoken dialog system, graphical user interface system, multi-modal system.
- **By device**: telephone-based systems, PDA systems, in-car systems, robot systems, desktop/laptop systems (native, in-browser systems, in-virtual machine), in-virtual environment, robots
- **By style**: command-based, menu-driven, natural language, speech graffiti
- **By initiative**: system initiative, user initiative, mixed initiative

2 Dialog Systems in Contact Centers

A good dialog system must be able to automate at least some of the mundane human tasks in a call center, while being responsive to the customer's needs. Ideally, menu based IVR systems used at most contact centers today, can be completely eliminated with the introduction of a trained dialog system. Dialog systems are widely used to decrease human workload in call centers.

2.1 Automated Call Centers

The most common type of dialogue in a call center revolves around seeking of information. There are specific characteristics of these kinds of interactions that can be modeled to design an automated system. In a human agent based call center, the agent acts as an intermediary between the caller and the data source. The data can be information about the caller, products, regulatory environment and so on. Most data in a call center environment is held in a structured form – mostly a database. However, some data may be in an unstructured form as well – a product brochure or a regulatory manual. The use of dialogue systems help in creating an automated call center that can mediate between the caller and the information in as effective a manner as a human-human interaction. There are several customer service situations in sectors such as education, government, entertainment, travel and healthcare where this model can be effectively used, such as:

1. Obtaining account information (telecom, financial services)
2. Responding to customer queries for product support through a call center, website or intranet portal
3. Travel reservations and enquiry
4. Weather reports

5. Guided selling where new customers are guided through the purchase process of a complex product which may require configuration choices to be made by the buyer.
6. Help Desk for internal employees
7. Website navigation where customers are guided to the right page in a complex site structure.
8. Technical support where the dialog system responds to technical problems with a product or service and assists the caller to resolve it.

The common characteristic of all these applications is that the interaction is based on locating, inserting or updating data objects in a structured database. This kind of interactions with structured data consists of the following elements:

1. **Data Structure**: This defines the set of entities and their attributes as well as how to identify references to these attributes in a user statement.
2. **List of transactions**: The transactions that are supported by the service and the method to detect references to these transactions. For example, if we consider a flight reservation system, the flight could be an entity with arrival time, departure time, destination city etc being the attributes and flight booking, flight enquiry etc being the transactions.
3. **Dialog models**: This will define how various conversational situations are handled in a human like fashion and consistent with the nature of service (for example, polite when handling a customer request, assertive when managing collections)
4. **Meta-strategy:** Handles privacy and security of the transaction. For example, a positive caller identification must happen prior to exchange of confidential information such as account balances.

Except for the dialog models, the other components mentioned above can be built using a limited amount of static data and are mostly domain independent. Even without component 3 basic mixed-initiative dialog capabilities can be built although the dialogue may not feel very 'natural'. Dialog models are essential to create a 'natural' conversation, but these can be derived only from large corpora of actual conversations between a live agent and a caller.

3 History

Since the early 1990s, several commercial spoken dialog systems have been developed to support various applications in telephone based services. Early spoken dialog systems functioned in restricted domains such as telephone based call routing systems (HMIHY), travel planning (DARPA Communicator) and weather information systems (JUPITER). Recently, dialog systems have become popular in embedded systems such as in-car navigation, entertainment systems and communication systems. Multi-modal systems such as the TALK project focused on developing new technologies for adaptive dialog systems using a combination of speech and graphics for car navigation. These technologies are now being used for applications such as telematics, smart home design and intelligent robots. Dialog systems have now evolved to support multiple tasks by accessing information from a variety of sources.

4 Designing Interactive Dialogues With Structured Data

The key principle of designing an interactive dialogue with structured data is to locate one or more items that meet certain specific criteria from a database. This may mean identifying the most convenient flight between point A and point B or retrieving the account balance from the caller's bank account. This overall objective is broken down into a set of sub goals that need to be satisfied to achieve the objective (for the flight example, this may include no stopovers, no early morning or late evening flights and so on). The dialogue manager needs to chart a path through the sub-goals such that A. the end objective is achieved, B. any partial constraints on the order or selection of sub-goals are met and C. the most efficient route is chosen.

The dialogue manager performs various interactions to achieve the goal of the conversation by capturing attribute values in frames that represent transactions and sub-goals. As more and more attribute values gets filled in the frames, the need for further dialog goes down. Spontaneous transactions work in this environment as the transaction values may be filled in any order and several values can be supplied simultaneously. The system will set key milestones to be reached by gathering information from the caller and these milestones may be approached in multiple ways. For example, if the system attempts to retrieve a record using a caller's last name and the database retrieves multiple records then the system may ask for a different attribute such as an address. This process is repeated until a unique record is identified. Thus most practical implementations of dialog systems have the flexibility to deal with user input arriving in any order or form as well as incomplete inputs without getting stuck on any one attribute. This is the key difference between a dialogue system and an IVR system based on system prompts. In a dialog system the exact progress of the dialog cannot be pre-defined or known in advance.

Dialog systems that are multilingual and domain-neutral store the language specific and task specific information in separate modules that can be loaded as and when required.

5 Task Based Spoken Dialog Systems

Task based dialog systems are those that are designed to accomplish a clearly defined task such as making a flight booking. Contact centers mostly use task based, spoken dialog systems. Spoken dialog systems are an advanced application of spoken language technology which aims to provide a human centric interface for users to access and manage information. As staff costs increase and agent turnover rises due to repeated monotonous tasks, dialog systems have become the preferred solution for contact centers as they offer superior performance at lower costs.

Spoken dialog systems receive speech inputs from the user and responds with the required information after performing necessary actions.

5.1 Components of Spoken Dialog Systems

The typical steps in a dialog system can be summarized as below:

1. An input decoder converts the caller's speech to plain text. The input decoder may consist of an automatic speech recognizer, handwriting recognizer and even a gesture recognizer.

2. A Natural Language Processor analyzes the plain text input and parses it for semantic and syntactic tagging.
3. The semantic data is analyzed by a dialog manager which analyzes it contextually using historic information, previous dialogs etc and manages the dialog flow.
4. The dialog manager then interacts with task managers for specific tasks such as retrieval of information, editing of data etc.
5. The dialog manager then produces the output using an output generator which may again include a natural language generator, a gesture generator and so on. The output is then rendered using an output renderer such as a text to speech engine.

In a spoken dialog system, these components work as follows:

User Input: This will be in the form of speech signals with noises

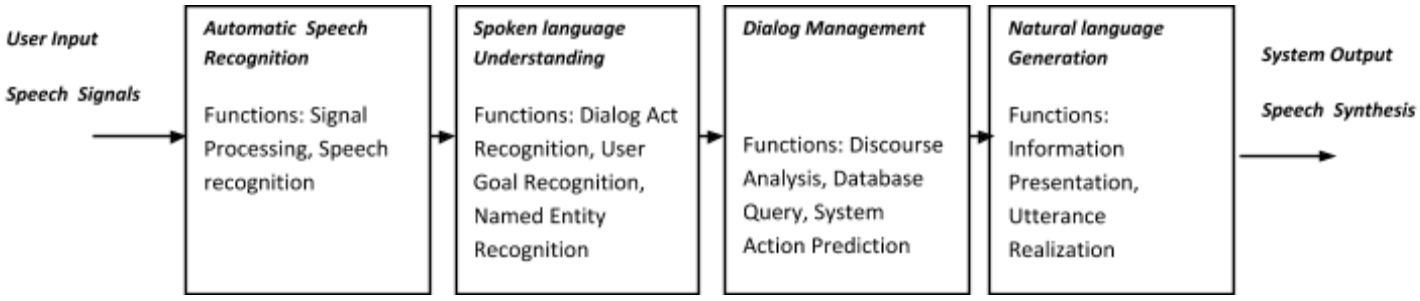
Automatic Speech Recognition (ASR): The speech waveform is transformed into a sequence of parameter vectors which is then further converted into a textual form as a sequence of words or phrases.

Spoken language Understanding (SLU): The textual form of the user speech is analyzed using NLP (natural language processing) modules such as morphological analysis, part of speech tagging and shallow parsing. The processed speech utterance is then mapped to a semantic frame (a meaning representation) in which the dialog act, user goal and named entities are extracted by a semantic parser.

Dialog Management (DM): The Dialog Manager is the central component of a spoken dialog system as it coordinates the activities of all the other components controls the dialog flow and even interacts with external applications for information. Some of the key activities performed by the DM include discourse analysis, querying of knowledge database and predicting the next action based on context. In addition, the DM is also responsible for error handling when ASR and SLU errors occur due to noises in the user input.

Natural Language Generation (NLG): The dialog system responses are generated in natural language form with the required items from an external database (such as a customer database) that answers the specific user request.

System Output: The system output may be visualized on a display if available and is synthesized by a text to speech module or pre-recorded audio. The basic architecture of a spoken dialog system can be depicted as below:



6 Dialog Management

6.1 Role of Dialog Management

The role of a DM is to accept the user's request and output the system response at a conceptual level. The user's intentions are represented as a semantic frame of SLU results. The key roles of a DM can be summarized as below:

a. Search external knowledge databases based on current inputs and context to obtain query results
b. Ask and obtain further information to refine the query to obtain a unique response
c. Obtain user confirmation for unclear information
d. Predict the next system action at a concept level so as to output the system's utterance in NLG and TTS modules
e. Control generic conversational mechanisms such as barge-ins, multi-party dialogue etc in order to make the conversation more 'natural'.

6.2 Degrees of Initiative

Initiative refers to who directs the progression of a dialog. A dialog can be modeled as a sequence of conversations between the caller and the system, and during the course of the dialog the initiative may shift between caller and system based on the discourse context. In a spoken dialog system, the degrees of initiative can be classified as a. system-initiative, where the system guides the dialog at each step; b. user-initiative – where the user takes control of the dialogs and the system only responds to it; and c. Mixed-initiative – where the system has the overall control, but the user has the flexibility to change the progress of the dialog.

In a system initiative dialog, the system asks one or more questions to extract some information from the user that will help to refine the query in a step by step manner. Once it has obtained sufficient information, it submits a query to the external knowledge base. The questions are usually structured in such a way that the user's response is a single word or phrase from which the named entity can be easily identified. One big advantage of this model is that the user inputs can be determined in advance and the set of vocabulary and grammar for each response can be restricted thus improving the performance of the ASR and SLU components of the system. Most commercial dialog systems are designed as system-initiative models. However, the disadvantage is that the dialog itself will be time-consuming as the system has to obtain one input at a time and the conversation is not very natural as the dialog flows are pre-defined with a set of limited vocabulary.

In a user-initiative dialog, the dialog flow is controlled by the user and the system shall only ask confirmation questions, if some input is not clear. Free and naturally flowing conversation is the biggest advantage of this model. However, developing a commercial user-initiative system is complicated as the ASR and SLU would need to deal with a much larger vocabulary and grammar and the DM will have to manage more flexible dialog flows.

A mixed-initiative dialog is one which is controlled by the system, while retaining the flexibility for the user to provide additional information or even to change the task. The complexity of this model is due to the difficulty in managing turn-taking – issues such as barge-ins need to be handled. Also, there is a risk that the user may try to take over the initiative by changing the task mid-way. In such cases, the system may either chose to work with the new request or bring the user back to the original task. This is the most natural form of human-human conversation, however practical implementation is extremely difficult.

Examples of Initiative Types	
User Initiative	User: I want to travel from New York to Chicago by flight System: There are 10 flights daily User: I need a non-stop flight in the morning
System Initiative	System: Please state your origin and destination city User: I want to travel from New York to Chicago System: Would you like a non-stop flight? User: Yes
Mixed Initiative	System: There are 10 flights daily between New York and Chicago. What time would you like to fly? User: The cheapest flight, please *(note the change of direction of the conversation initiated by the user)*

6.3 Error Handling

Practical dialog systems have been used in several different application domains as spoken language technologies such as ASR and SLU have evolved significantly in the past decades. However, robust error handling remains a key concern area for spoken dialog systems. Errors in a dialog system typically arise due to errors in speech recognition and language understanding. The caller's input may be unclear or incomplete due to inaccurate speech recognition. In some cases, even if speech recognition is accurate, the SLU module may not capture the correct contextual meaning due to lack of sufficient data. These errors can cause the dialog system to respond with inappropriate responses to user queries. While the straightforward solution to this problem is to improve the accuracy of the recognition and understanding phases of the system, it may not actually be practical due to noisy environments and unexpected user inputs.

Error handling therefore is often a responsibility of the DM module which adopts dialog mechanisms to detect and repair potential errors at the conversational level. Error detection is usually measured as a confidence score computed for the speech recognition and language understanding modules. The confidence score is then compared against a manually defined threshold and corrective action is taken when the threshold is breached. Error recovery strategies deployed by the DM would include an explicit or implicit confirmation from the user or a dialog rephrasing to repair the error. An explicit confirmation usually is in the form of a question that explicitly asks for confirmation of a target data, such as the departure date in a flight reservation system. The caller may be asked to respond with a 'yes' or 'no'. In an implicit confirmation,

the next question that the DM asks will embed the previous information provided by the caller. These strategies are useful when the initial user input is unclear and the user's confirmation helps to confirm the target slot values.

Examples of Error Handling Strategies	
Explicit Confirmation	User: I want to travel from New York to Chicago by flight System: So, you want to go to Chicago from New York? Please answer yes or no User: Yes
Implicit Confirmation	User: I want to travel from New York to Chicago by flight System: Do you want a morning flight from New York to Chicago or an evening flight? User: Morning flight
Static Rephrasal	User: I want to travel from New York to Chicago by flight System: Sorry, would you please rephrase? User: I want to know about flights to Chicago from New York.
Dynamic Rephrasal	User: I want to travel from New York to Chicago by flight System: Sorry, would you please rephrase using expression 'I'd like to fly from <origin> to <destination> ? User: I'd like to fly from New York to Chicago.

7 Modern Approaches to Dialog Management

There are mainly three approaches to dialog management in a dialog system. These are:

a. Knowledge-based dialog management
b. Data-driven dialog management
c. Hybrid dialog management

7.1 Knowledge-based Dialog Management

In the early days dialog systems were domain specific and were designed by application developers with domain-specific knowledge. As a result the rules were designed based on continuous experiments with real users and a restricted language set was used to perform highly structured tasks using system-initiated dialogs. This approach is fairly simple and easy to implement and has been used in several practical applications. However, the rule base is pre-defined and hard coded, making the dialog flow inflexible. Scenarios such as over-information (where the caller provides more information than the system asked for)

cannot be handled by such systems. In addition, these systems are not domain-portable and every time it needs to be used for a new domain, the system will have to be designed from scratch.

In order to overcome these limitations, generic dialog modeling approaches based on agenda models are being proposed. One of the most popular dialog management frameworks using this approach is RavenClaw a two-tier framework that has a clear separation between domain-dependent and domain independent aspects of dialog control logic. This model is still time consuming when it comes to designing the knowledge sources which is done manually. A further improvement to this approach is to automatically model prior knowledge from previous dialog corpus.

7.2 Data-driven Dialog Management

A more recent approach to dialog management is a data driven approach, where training is done automatically with little human intervention. Domain adaptability can be achieved in an easier manner using this approach as it requires only additional time and effort for collecting data specific to the domain. Some of the algorithms used in the data driven approach are based on Markov Decision Processes (MDPs) enabled Reinforcement Learning (RL). However data driven approaches are still mostly in the research domain and not many practical implementations are available.

7.3 Hybrid Approaches to Dialog Management

RL based data-driven DMs need a large amount of dialog corpora to learn an optimal policy due to the large state space and policy space. In order to address this, user simulation techniques are used to generate a large number of simulated dialogs from limited real corpora. Recent research has focused on hybrid approaches that integrate reinforcement learning and supervised learning in order to optimize dialog policies with a fixed dialog corpus thus eliminating the need for large amount of dialog corpora which a traditional RL based model requires. In this approach, RL is used to optimize a measure of dialog reward and supervised learning is used to restrict the learnt policy to the portions where data is available.

8 Evaluation of Spoken Dialog Systems

In order to improve the system's performance, it is necessary that the dialog systems are periodically evaluated using quantitative metrics such as dialog success rate, task success rate etc which measure dialog success. Dialog costs are measured using metrics such as average turn length, number of turns in the dialog and so on. Humans users are typically used for evaluating dialog systems, however this is an expensive proposition and can also lead to delays in case the evaluators are not available. In order to overcome this, dialog simulation is used, where a simulated user interacts with the dialog system. There are three main levels of simulation that is required for a robust evaluation of a spoken dialog system – user intention simulation, user surface simulation and error simulation. In general a user simulator is split into two levels – user intention simulator and user utterance simulator.

8.1 User Intention Simulation

The purpose of user intention simulation is to produce subsequent user intentions for a given discourse context. This is usually represented as user's goals and is included as information on the user's utterance

(surface). A general user intention simulation in a spoken dialog system can be represented by the probabilistic formula P(user intention|discourse context). User intention simulators may be implemented using a knowledge-based approach or a data-driven approach. In the former approach, the model is built up from human discourse knowledge and the developer can create different rules that determine how the simulated user behaves given specific discourse information. However, designing all possible rules that cover diverse situations caused by ASR, SLU and DM errors is a difficult and time consuming proposition.

If enough data is available, it is easier to build a data-driven user intention simulator as most probabilistic methods are domain and language neutral.

8.2 User Utterance Simulation

Since a spoken dialog system's performance is dependent on the performance of the SLU as well as the DM, it is necessary to simulate user utterances as well and not restrict the simulation to user intentions. User utterance simulation generates surface level utterances that express a given user intention. Since there can be many semantically equivalent sentences that express a given user intention, the user utterance simulation can be formulated probabilistically as follows:

P(W|user intention), where $W=\{w_1, w_2,...,w_N\}$ is a word (w_N) sequence.

8.3 User Simulation Criteria

Since simulated users are expected to replace real users in the test environment, the main criterion of user simulation is how closely the simulated patterns mimic natural conversation scenarios. In user intention simulation, the simulated intention should be natural to the relevant discourse context. In the user surface simulation, the simulated utterance should closely resemble a real user's utterance. In an ASR channel simulation, the simulated sentence and the added noise together should be similar to the noise levels that are expected to be recognized by a real ASR.

Other than naturalness and accuracy which are important in any natural language problem, user simulation also requires variety. Since the main goal of user simulation is to evaluate the robustness of the dialog system in various environments, it is essential that the simulation generates diverse intention, surface and error patterns.

Yet another key criterion of user simulation is controllability with allows the developers to manipulate the characteristics of the simulated users, such as the user type, language fluency levels and so on.

9 Summary

Dialog management techniques and robust error handling are key aspects of dialog systems. These control the interaction with the caller as well as with external data sources, while improving the robustness of the system in a noisy environment. Data driven approaches are being used to build dialog management systems that can overcome the problems of traditional knowledge based approaches. While the adoption levels of spoken dialog systems will increase in the future, there are still several challenges that need to be overcome

such as improving the speech recognition accuracy and better error handling techniques. Spoken dialog systems will be used in new types of applications that go beyond just information access through telephone into wider application areas such as learning assistance. Self-learning dialog systems may evolve in future which will learn from its own experience to improve performance over time.

10 References

1. Recent Approaches to Dialog Management for Spoken Dialog Systems. Lee et.al
2. Considerations in the Design and Evaluation of Spoken Language Dialog Systems. Lamel, Rosset and Gauvain.
3. Dialogue Management for an Automated Multilingual Call Center Hardy, Strzalkowski and Wu