Morgan & Claypool Publishers

Conversational AI

Dialogue Systems, Conversational Agents, and Chatbots

Michael McTear

Synthesis Lectures on Human Language Technologies

Graeme Hirst, *Series Editor*

Conversational AI

Dialogue Systems, Conversational Agents, and Chatbots

Synthesis Lectures on Human Language Technologies

Synthesis Lectures on Human Language Technologies is edited by Graeme Hirst of the University of Toronto. The series consists of 50- to 150-page monographs on topics relating to natural language processing, computational linguistics, information retrieval, and spoken language understanding. Emphasis is on important new techniques, on new applications, and on topics that combine two or more HLT subfields.

Conversational AI: Dialogue Systems, Conversational Agents, and Chatbots
Michael McTear
2020

Embeddings in Natural Language Processing: Theory and Advances in Vector Representations of Meaning
Mohammad Taher Pilehvar and Jose Camacho-Collados
2020

Natural Language Processing for Social Media, Third Edition
Anna Atefeh Farzindar and Diana Inkpen
2020

Statistical Significance Testing for Natural Language Processing
Rotem Dror, Lotem Peled, Segev Shlomov, and Roi Reichart
2020

Deep Learning Approaches to Text Production
Shashi Narayan and Claire Gardent
2020

Linguistic Fundamentals for Natural Language Processing II: 100 Essentials from Semantics and Pragmatics
Emily M. Bender and Alex Lascarides
2019

Cross-Lingual Word Embeddings
Anders Søgaard, Ivan Vulić, Sebastian Ruder, Manaal Faruqui
2019

Bayesian Analysis in Natural Language Processing, Second Edition
Shay Cohen
2019

Argumentation Mining
Manfred Stede and Jodi Schneider
2018

Quality Estimation for Machine Translation
Lucia Specia, Carolina Scarton, and Gustavo Henrique Paetzold
2018

Natural Language Processing for Social Media, Second Edition
Atefeh Farzindar and Diana Inkpen
2017

Automatic Text Simplification
Horacio Saggion
2017

Neural Network Methods for Natural Language Processing
Yoav Goldberg
2017

Syntax-based Statistical Machine Translation
Philip Williams, Rico Sennrich, Matt Post, and Philipp Koehn
2016

Domain-Sensitive Temporal Tagging
Jannik Strötgen and Michael Gertz
2016

Linked Lexical Knowledge Bases: Foundations and Applications
Iryna Gurevych, Judith Eckle-Kohler, and Michael Matuschek
2016

Bayesian Analysis in Natural Language Processing
Shay Cohen
2016

Metaphor: A Computational Perspective
Tony Veale, Ekaterina Shutova, and Beata Beigman Klebanov
2016

Grammatical Inference for Computational Linguistics
Jeffrey Heinz, Colin de la Higuera, and Menno van Zaanen
2015

Automatic Detection of Verbal Deception
Eileen Fitzpatrick, Joan Bachenko, and Tommaso Fornaciari
2015

Natural Language Processing for Social Media
Atefeh Farzindar and Diana Inkpen
2015

Semantic Similarity from Natural Language and Ontology Analysis
Sébastien Harispe, Sylvie Ranwez, Stefan Janaqi, and Jacky Montmain
2015

Learning to Rank for Information Retrieval and Natural Language Processing, Second Edition
Hang Li
2014

Ontology-Based Interpretation of Natural Language
Philipp Cimiano, Christina Unger, and John McCrae
2014

Automated Grammatical Error Detection for Language Learners, Second Edition
Claudia Leacock, Martin Chodorow, Michael Gamon, and Joel Tetreault
2014

Web Corpus Construction
Roland Schäfer and Felix Bildhauer
2013

Recognizing Textual Entailment: Models and Applications
Ido Dagan, Dan Roth, Mark Sammons, and Fabio Massimo Zanzotto
2013

Linguistic Fundamentals for Natural Language Processing: 100 Essentials from Morphology and Syntax
Emily M. Bender
2013

Semi-Supervised Learning and Domain Adaptation in Natural Language Processing
Anders Søgaard
2013

Semantic Relations Between Nominals
Vivi Nastase, Preslav Nakov, Diarmuid Ó Séaghdha, and Stan Szpakowicz
2013

Computational Modeling of Narrative
Inderjeet Mani
2012

Natural Language Processing for Historical Texts
Michael Piotrowski
2012

Sentiment Analysis and Opinion Mining
Bing Liu
2012

Discourse Processing
Manfred Stede
2011

Bitext Alignment
Jörg Tiedemann
2011

Linguistic Structure Prediction
Noah A. Smith
2011

Learning to Rank for Information Retrieval and Natural Language Processing
Hang Li
2011

Computational Modeling of Human Language Acquisition
Afra Alishahi
2010

Introduction to Arabic Natural Language Processing
Nizar Y. Habash
2010

Cross-Language Information Retrieval
Jian-Yun Nie
2010

Automated Grammatical Error Detection for Language Learners
Claudia Leacock, Martin Chodorow, Michael Gamon, and Joel Tetreault
2010

Data-Intensive Text Processing with MapReduce
Jimmy Lin and Chris Dyer
2010

Semantic Role Labeling
Martha Palmer, Daniel Gildea, and Nianwen Xue
2010

Spoken Dialogue Systems
Kristiina Jokinen and Michael McTear
2009

Introduction to Chinese Natural Language Processing
Kam-Fai Wong, Wenjie Li, Ruifeng Xu, and Zheng-sheng Zhang
2009

Introduction to Linguistic Annotation and Text Analytics
Graham Wilcock
2009

Dependency Parsing
Sandra Kübler, Ryan McDonald, and Joakim Nivre
2009

Statistical Language Models for Information Retrieval
ChengXiang Zhai
2008

Conversational AI: Dialogue Systems, Conversational Agents, and Chatbots

Michael McTear

www.morganclaypool.com

ISBN: 9781636390314 paperback
ISBN: 9781636390321 ebook
ISBN: 9781636390338 hardcover

DOI 10.2200/S01060ED1V01Y202010HLT048

A Publication in the Morgan & Claypool Publishers series
SYNTHESIS LECTURES ON HUMAN LANGUAGE TECHNOLOGIES

Lecture #48
Series Editor: Graeme Hirst, *University of Toronto*
Series ISSN
Print 1947-4040 Electronic 1947-4059

Conversational AI

Dialogue Systems, Conversational Agents, and Chatbots

Michael McTear
Ulster University

SYNTHESIS LECTURES ON HUMAN LANGUAGE TECHNOLOGIES #48

MORGAN & CLAYPOOL PUBLISHERS

ABSTRACT

This book provides a comprehensive introduction to Conversational AI. While the idea of interacting with a computer using voice or text goes back a long way, it is only in recent years that this idea has become a reality with the emergence of digital personal assistants, smart speakers, and chatbots. Advances in AI, particularly in deep learning, along with the availability of massive computing power and vast amounts of data, have led to a new generation of dialogue systems and conversational interfaces. Current research in Conversational AI focuses mainly on the application of machine learning and statistical data-driven approaches to the development of dialogue systems. However, it is important to be aware of previous achievements in dialogue technology and to consider to what extent they might be relevant to current research and development. Three main approaches to the development of dialogue systems are reviewed: rule-based systems that are handcrafted using best practice guidelines; statistical data-driven systems based on machine learning; and neural dialogue systems based on end-to-end learning. Evaluating the performance and usability of dialogue systems has become an important topic in its own right, and a variety of evaluation metrics and frameworks are described. Finally, a number of challenges for future research are considered, including: multimodality in dialogue systems, visual dialogue; data efficient dialogue model learning; using knowledge graphs; discourse and dialogue phenomena; hybrid approaches to dialogue systems development; dialogue with social robots and in the Internet of Things; and social and ethical issues.

KEYWORDS

conversational interface, dialogue system, voice user interface, embodied conversational agent, chatbot, deep learning, data-driven, statistical, end-to-end learning, evaluation metrics, performance evaluation, usability, multimodality, hybrid systems, ethical issues

Contents

Preface

Conversational AI has been defined as "the study of techniques for creating software agents that can engage in natural conversational interactions with humans" [Khatri et al., 2018a]. While the idea of interacting with a computer using text or voice has been around for a long time, it is only recently that it has become a reality. Nowadays, people can talk to digital personal assistants on their smartphones, they can ask questions or issue commands to voice-enabled smart speakers, and they can navigate using voice-based systems in their cars. In other words, Conversational AI has become ubiquitous. Various terms are used in the literature to describe these systems, for example, *Dialogue System*, *Voice User Interface*, *Conversational Agent*, and *Chatbot*. In this book, the generic term *Dialogue System* will be used.

The aim of the book is to provide a readable introduction to the various concepts, issues, and technologies of Conversational AI along with a comprehensive list of references for those who wish to delve further. The book is mainly targeted at researchers and graduate students in Artificial Intelligence, Natural Language Processing, Human-Computer Interaction, and Conversational AI.

The structure of the book is as follows. Chapter 1 provides an introduction to dialogue systems, beginning with a brief history of different types of systems and then looking at present-day systems and what is required to model conversational interaction. Traditionally, dialogue systems were handcrafted: designers created rules for conversational interaction based on best practice guidelines. Chapter 2 describes the rule-based approach, presenting first the sub-components of a typical modularised architecture, then looking at the processes of development and reviewing some currently available toolkits that can be used to develop dialogue systems.

While the rule-based approach is still used extensively, particularly for commercially deployed dialogue systems, the current trend in academic research as well as in industrial research laboratories is dominated by statistical approaches using machine learning and large corpora of conversational data. Chapter 3 describes statistical approaches applied to the sub-components of the typical modular architecture, including Reinforcement Learning for dialogue management.

Chapter 4 looks at the question of how to evaluate a dialogue system, beginning with a comparison of laboratory-based evaluations and those conducted in more realistic settings. Until recently, task-oriented dialogue systems have been the focus of academic research and also in commercially deployed voice user interfaces, and a variety of evaluation metrics have been devised for these applications. With a new focus on open-domain non-task-oriented dialogue systems new metrics have been proposed and applied in various challenges and competitions for Conversational AI. The chapter concludes by reviewing some frameworks that aim to integrate some of these metrics into a unified system of evaluation.

Chapter 5 reviews the latest research in neural dialogue systems that have come to dominate the field. Neural dialogue systems are trained end-to-end using the Sequence-to-Sequence (seq2seq) approach. The chapter provides a fairly non-technical overview of the technology of neural dialogue and examines what has been achieved within this new paradigm as well as issues that are the focus of ongoing research.

Conversational AI is a rapidly developing field, with a lot of open issues and opportunities. Chapter 6 explores some challenges and areas for further development, including multimodal dialogue systems, visual dialogue, data efficiency in dialogue model learning, the use of external knowledge, how to make dialogue systems more intelligent by incorporating reasoning and collaborative problem solving, discourse and dialogue phenomena, hybrid approaches to dialogue systems development, dialogues with social robots and with the Internet of Things, and social and ethical issues.

Michael McTear
October 2020

Acknowledgments

During the course of writing this book, I have seen how Conversational AI has developed into a fast-moving field with new research papers appearing on a weekly basis. The field has come a long way since when I published a book on conversational computers in the 1980s. At that time, I was seen as a bit of an oddity for thinking that some day we would be able to talk to computers the way we now interact with our smartphones and smart speakers.

Over the years, I have learned a lot about conversational computers. I have benefited from meeting colleagues and friends at academic conferences such as Interspeech, SigDial, and IWSDS, including: Zoraida Callejas, David Griol, Kristiina Jokinen, Ramón López-Cózar, Wolfgang Minker, Sebastian Möller, Oliver Lemon, Verena Rieser, Alex Rudnicky, Gabriel Skantze, David Traum, Stefan Ultes, Jason Williams, and Steve Young.

In recent years, I have attended and presented at industrial conferences such as SpeechTEK, Conversational Interaction, ProjectVoice, and the RE·WORK AI Assistant Summits, where I have met and learned from colleagues including: David Attwater, Gérard Chollet, Deborah Dahl, Jim Larson, William Meisel, Robert J. Moore, Andy Peart, Wolf Paulus, and Bill Scholz.

I have also received support and encouragement from colleagues at Ulster University, in particular Maurice Mulvenna, Raymond Bond, Paul McKevitt, and Jonathan Wallace, as well as researchers Gillian Cameron, Sam Holmes, Paddy McAllister, Courtney Potts, Bronagh Quigley, and Shay van Dam.

A first draft of the book was reviewed by Oliver Lemon and an anonymous second reviewer, who both provided detailed comments and suggestions for improvements. I also received helpful comments from my close colleagues Zoraida Callejas and David Griol. Kristiina Jokinen provided comments on an earlier version of Chapters 1 and 2. Any deficiencies that remain are mine.

I would also like to acknowledge the ongoing support I received from Morgan & Claypool, including Michael Morgan, Publisher, Christine Kiilerich, Assistant Editor, and Graeme Hirst, Series Editor of the Synthesis Lectures on Human Language Technologies. Dr. C.L. Tondo helped me out on numerous occasions with problems I was encountering in LaTeX.

My wife, Sandra, supported and encouraged me throughout and never complained when I withdrew to my study in the attic to work on the book.

Finally, I would like to dedicate the book to my children Siobhan, Paul, Anna, and Rachel, and grandchildren Dylan, Sofia, Ethan, Olivia, and Cillian.

Michael McTear
October 2020

Glossary

alignment Alignment is where two words, phrases, or sentences are compared to discover relationships and similarities between them. Alignment is used extensively in machine translation to find relationships between the words and phrases in two languages and to evaluate the accuracy of the translation. Alignment is less applicable in dialogue systems as typically there can be a wide range of responses to a previous utterance in a dialogue and the responses do not usually use the same words as the previous utterance. For this reason alignment is less useful for evaluating the system's responses in a dialogue. 136

AMT Amazon Mechanical Turk is a crowdsourcing application for researchers and businesses through which they can recruit temporary workers to conduct research and other tasks virtually. 95

anaphoric reference Anaphoric reference is when a word or phrase refers to something mentioned previously. The use of pronouns and words such as *there* to refer back are examples of anaphoric reference. 15, 21, 32, 39, 64, 160

backchannels In conversation a backchannel is when one participant is speaking and the other produces verbal expressions such as *yeah*, *uh-huh*, and *right*, or nonverbal behaviors such as nodding, or combinations of verbal and nonverbal behaviors. Backchannel behaviors indicate agreement, understanding, acknowledgement, or attention and are not intended as attempts to take the turn from the current speaker. 176

beam search Beam search is an algorithm that expands the most promising node in a graph, optimizing the search and reducing memory requirements. Beam search is often used to maintain tractability in applications where the search would otherwise become intractable. For example, beam search may be used in response generation in a dialogue system where decoding the most likely sequence would require searching through all possible output sequences. Beam search returns a list of the most likely output sequences up to a predetermined limit. In contrast, greedy search selects the most likely word at each step in the decoding, with the result that the search is fast but not necessarily optimal. 144

CFG A Context-Free Grammar, also known as Phrase Structure Grammar, analyses sentences in terms of phrases and words and models sentence structure hierarchically in a parse tree. For examples, see Figures 2.2 and 3.3. 46

coherence Coherence is what makes a text meaningful where ideas are connected logically to create meaning and maintain consistency. 64, 99, 109, 111, 145, 146, 148

cohesion Cohesion refers to features in a text that link items together. Lexical cohesion involves the use of repetition or words that are synonyms, hyponyms, or antonyms. Grammatical cohesion involves the use of expressions for anaphoric reference such as pronouns. 64

conditional random fields Conditional random fields (CRFs) are used in pattern recognition and machine learning for structured prediction, for example, in sequence tagging to identify elements in a sentence such as parts-of-speech or entities. 75

conversation flow Conversation flow, also known as dialogue flow, can be represented visually as a graph. Conversation flow specifies the flow of a conversation from a starting point to a finishing point, including branches where the conversation can go in different directions. Conversation flow graphs are used widely by designers to specify the actions and steps in a conversation. 40, 53–55, 64, 107, 136

cosine similarity Cosine similarity is used to calculate the similarity between word vectors in multidimensional space by measuring the cosine of the angle between them. Vectors that are more similar have a smaller angle between them. 132, 147

deictic A deictic expression is a word or phrase that points to the time, place, or situation in which a speaker is speaking. Resolving deictic expressions depends on context. In a physical setting a deictic expression may be accompanied by pointing. Deixis is expressed in English using demonstratives such as *this* and *that*, personal pronouns such as *he* and *she*, adverbs such as *here* and *there*, expressions of time, e.g., *now* and *when* and tense, e.g., *he had gone*, which describes a time relative to some other time. 174

dialogue flow Dialogue flow, which can be represented visually as a graph, specifies the flow of a dialogue from a starting point to a finishing point, including branches where the dialogue can go in different directions. Dialogue flow graphs are used widely by designers to specify the actions and steps in a dialogue. 25, 49, 54, 61, 94

dialogue policy Dialogue policy refers to the decision-making component of a Dialogue Manager (DM). A policy is a mapping from states in the dialogue to actions to be taken in those states. DM takes an action in each state and receives a reward. The aim is to maximize the final reward. Reinforcement Learning (RL) is used to learn a policy that maximizes the rewards by learning to take the best actions in each state of the dialogue. 66, 71, 72, 79, 88, 99

dialogue state tracking Dialogue state tracking, also known as belief state tracking, is a core element of the dialogue management component. The dialogue state represents all aspects of the interaction that are relevant for the system's choice of its next action. The dialogue state tracker updates the dialogue state as a result of new observations, such as the user's utterances or other events that are relevant to the dialogue. Dialogue state tracking becomes intractable in large dialogue systems as the number of states to be tracked becomes very large. Addressing this issue has been a major focus in dialogue systems research. Dialogue state tracking has been investigated in a number of challenges Dialogue State Tracking Challenge (DSTC) in which different approaches to Dialogue state tracking are compared and evaluated. 18, 72, 77, 84, 86, 99, 153, 154

discriminative A discriminative model is used for classification and regression tasks in supervised machine learning tasks. The model finds a decision boundary between classes. For example, if a classification task involves distinguishing between pictures of cars and vans, the model will be able to decide if a given picture shows a car or a van based on the most similar examples from the training data. Discriminative models are contrasted with generative models. 4, 99

entity In Natural Language Processing an entity is an object that is named in an utterance that is relevant to the application. For example, in a flight booking application there would be entities for **destination**, **origin**, **date**, **time**, and so on. The Natural Language Understanding component extracts values for these entities during the dialogue. Some platforms use the term *slot*. Extracting entities in a text is called *Entity Extraction*. See also named entity. 5, 32, 68, 69, 97, 112

F1 In statistical analysis F1 (also known as F-score or F-measure) is a measure of the accuracy of a test. F1 measures the balance (or harmonic mean) between precision and recall, where adjusting one of the measures to improve it will often result in lowering the score for the other. For example, improving precision can result in a lower score for recall, and vice versa. The highest possible value of F1 is 1, indicating perfect precision and recall. 98, 113, 121, 180

few-shot learning Few-shot learning in natural language processing refers to the learning of tasks using a small number of examples. Few-shot learning addresses the problem of data sparseness in machine learning. A more extreme case is one-shot learning in which one, or a few, examples are used to create a model capable of classifying many examples in the future. 78, 144, 166

FIA The form interpretation algorithm (FIA) is used in VoiceXML to process forms. There are three phases. In the *select* phase the next unfilled form is selected. In the *collect* phase the user is prompted for input using appropriate grammars and the system waits for the user's input. In the *process* phase the input is processed by filling form items (e.g., fields) and executing other actions such as input validation or response to events. Platforms such as Dialogflow, Amazon Lex, and IBM Watson Assistant use a similar a algorithm in slot-filling dialogues. 57

finite state dialogue In a finite state dialogue the flow of the dialogue is modeled as a finite-state-machine that specifies the complete dialogue flow. The finite state machine specifies all the states that the dialogue can be in and all possible transitions between the states. A dialogue flow graph is a visual representation of a finite state dialogue. 18

finite state grammar A finite state grammar is a simple grammar that analyses sentences word by word in a linear fashion. A finite state grammar can be used to analyze simple inputs in a dialogue system. 45

Gaussian Process A Gaussian process (GP) is a nonparametric model that is used when the distribution of data cannot be defined in terms of a finite set of parameters. GP is useful for modeling uncertainty. 88

generative A generative model is used in classification tasks in supervised machine learning. In contrast to a discriminative model that distinguishes between classes, e.g., to separate pictures of cars from pictures of vans, a generative model is able to produce a new picture of either class. Generative models are often used to find the hidden parameters of a distribution, whereas discriminative models are used to find the boundary that separates the data into different classes. 3, 99

intent An intent is a representation of the intention of the user in an application. For example, the utterance *I want to book a flight to London* could be labeled as the intent **book_flight**. Generally, the developer defines the intents to be used in an application and supplies a list of utterances that can express each intent. Most toolkits offer system intents for commonly used intents. Intents are detected from utterances using machine learning, i.e., classification, which is a probabilistic approach with the advantage that the utterances of the user do not have to exactly match the utterances defined for an intent, thus providing greater flexibility than a rule-based grammar. 47, 60–62, 65, 67, 68, 70, 75, 76, 85, 97, 139

knowledge graph A knowledge graph represents relationships between objects, concepts, or events in a domain. A knowledge graph models specific items and their relationships in contrast to an ontology which models general items and their relationships. In other words, a knowledge graph is an ontology with data. Large knowledge graphs model billions of entities and their relationships and are usually multi-domain. 5, 49, 167, 168, 170, 177

language model A language model is a probability distribution over sequences of words. A language model is used to estimate how probable different words and phrases are. Language models have been used extensively in speech recognition and in natural language understanding to help distinguish between words and phrases that sound similar. For example, *to*, *too*, and *two* sound the same but have different meanings and different usage in a text. Very large language models have been created, such as Google's BERT, Microsoft's Turing NLG, and OpenAI's GPT-3, and are now being used in a wide variety of Natural Language Processing tasks. These language models are able to produce text that is in many cases indistinguishable from human-produced text. 5, 45, 128, 149, 158

language modeling See language model. 134, 139, 144

maximum likelihood Maximum likelihood estimation (MLE) is a method (or function) that is used to determine the values of the parameters of a machine learning model from a data sample so that under the model the observed data is most probable. 129, 147

N-best An N-best list is an ordered list of hypotheses, for example, of recognition results from the speech recognition component. In many cases the 1st-best hypothesis is selected by default, but it is also possible to re-score the N-best list to retrieve the correct word based on information from other component of the system, such as semantic or contextual information. 44, 45, 48, 83

N-gram An N-gram is a sequence of N words, e.g., a bigram is a sequence of two words, a trigram is a sequence of three words, etc. N-grams are used in language modeling to predict the probability of the next word in a sequence. The probabilities are learned from a large corpus of data (sentences). N-grams are used in applications such as auto-completion in emails and spell checking. 45, 79, 128, 147

named entity A named entity is a real word object such as a person, location, organization, etc. that is referred to with a proper name, e.g., Boris Johnston, London, etc. Named entities are instances of entities. For example, in the sentence *Boris Johnston is an instance of Prime Minister of the UK*, *Boris Johnston* and *UK* are examples of named entities and *Prime Minister* is an example of an entity. Recognizing named entities in a text is called *Named Entity Recognition*. See also entity. 3, 68

ontology An ontology, as used in computer science and the Semantic Web, is a formal representation of the relationship between concepts or entities within a domain. For example, in medicine an ontology would represent knowledge about symptoms, diseases, and treatments. An ontology models general entities and their relationships and is typically curated by hand for a specific domain, whereas a knowledge graph models specific items and their relationships, often in multiple domains. However, often the terms are used interchangeably. 4, 49, 168, 177, 178

overfitting Overfitting in machine learning is when a model is too closely fit to the training data. In particular, the model might learn irrelevant details and noise in the training data that are included as concepts in the model. As a result these concepts will not apply when the model is applied to new data so that the performance of the model is adversely affected in terms of its ability to generalize. 161

parse Parsing is a process in which a string of text such as a sentence is analyzed according to the rules of a grammar. Parsing determines if the sentence is correct according to the grammar rules and, in the case of semantic parsing, returns the meaning of the sentence. The result of parsing can be shown in a parse tree, as in Figures 2.2 and 3.3 that show the relationships between phrases and words in the parsed sentence. 1, 6, 46, 73–75, 97

parser A parser is a computer program that creates a parse of a sentence. 47, 73, 75

perplexity Perplexity measures the uncertainty of a language model in terms of the number of choices available when selecting the next token in a sequence, where a lower perplexity indicates greater confidence. 106, 113, 121, 128, 141, 149

precision Precision is the ratio of correctly identified items over all the items detected, reflecting the accuracy of the classification. For example, if 60 items were identified and 40 of them were correct, precision would be 40/60, i.e., 2/3. Precision is usually measured along with recall. 6, 61, 98, 105, 113

Q-function The Q-function in Reinforcement Learning estimates the overall expected reward of taking a particular action in a given state and then following the optimal policy. 6, 83, 87, 88

Q-learning In Reinforcement Learning Q-learning is an algorithm that is used to estimate an optimal policy using the Q-function. The "Q" in Q-learning stands for quality, in other words, it represents how useful an action is in achieving a future reward. Q-learning is applied to and adjusts the Q-values in the system's state space. 87

Q-values Q-values in Reinforcement Learning are the values of each state-action pair. At the start of the learning process the Q-values are initialized to zero and are then updated after each learning episode. 6, 87

recall Recall is the ratio of the items that the system has correctly classified over all the items that should have been identified, reflecting the system's ability to identify as many correct items as possible. For example, if 60 items were identified out of a total of 100 relevant items, recall would be 60/100, i.e., 3/5. Recall is usually measured along with precision. 6, 98, 113

SVM A Support Vector Machine (SVM) is a supervised learning model used in machine learning tasks such as classification and regression tasks. In classification the objective is to separate two classes of data points using a hyperplane (i.e., decision boundary) that shows the maximum distance between the two classes of data points. An SVM model is given sets of training examples for each category and once trained can classify new text. 75

ultimate default category The Ultimate Default Category in Artificial Intelligence Markup Language (AIML) is matched when the dialogue system cannot match the user's utterance against one of its list of categories. The system's response is specified in the <template>, for example, *Sorry I didn't understand that, can you please repeat?* , or *tell me more*. The term **fallback intent** is used on most other platforms. 58–60

Wizard of Oz The Wizard of Oz (WoZ) technique is used in studies of human-computer interaction to collect data about how users interact with an automated system, such as a dialogue system. In the technique the wizard, who is concealed from the user in a laboratory setting, simulates the dialogue system and the user believes that they are interacting with the system. The technique is particularly useful for observing the use and effectiveness of a user interface. The wizard may perform the role of part of the system. For example, the wizard might take the role of the dialogue manager to test different strategies before they have been implemented in the system. 53, 70, 77, 139

zero-shot learning Zero-shot learning in natural language processing refers to the learning of tasks for examples that have never been seen during training. Zero-shot learning provides a way to understand the generalization power of the learned representations. 165, 166

Acronyms

AI Artificial Intelligence. 15, 16, 19, 29, 54, 65, 130, 146

AIML Artificial Intelligence Markup Language. 7, 21, 55, 58–60, 67, 69, 179

AMT Amazon Mechanical Turk. 95, 105, 113, *Glossary:* AMT

ASR Automatic Speech Recognition. 17, 18, 29, 45, 46, 48–50, 57, 68, 72, 76, 82, 83, 85, 89, 94, 98, 101, 102

BLEU Bilingual Evaluation Understudy. 104, 105, 149

CFG Context-Free Grammar. 46, 73, *Glossary:* CFG

CSLU Center for Spoken Language Understanding. 18

DAMSL Dialog Markup in Several Layers. 48

DBDC Dialogue Breakdown Detection Challenge. 110, 157

DM Dialogue Manager. 2, 49, 50, 60, 65, 66, 71, 72, 75–78, 82, 88, 98, 99, 101, 175

DNNs Deep Neural Networks. 45, 52, 71, 125

DR Dialogue Register. 77, 78

DSTC Dialogue State Tracking Challenge. 3, 86, 99, 152, 154

ECA Embodied Conversational Agent. 22

FIA Form Interpretation Algorithm. 57, 67, *Glossary:* FIA

GP Gaussian Process. 88, *Glossary:* Gaussian Process

GPT Generative Pre-Training. 21, 130, 139, 140, 144

GRU Gated Recurrent Unit. 135

HMIHY How May I Help You. 19

HMMs Hidden Markov Models. 45

IQ Interaction Quality. 113, 118–120, 122

LSTM Long Short-term Memory. 80, 134, 135, 137, 149, 174

MDP Markov Decision Process. 81–83, 86, 87, 140

ML Machine Learning. 29

MT Machine Translation. 104, 105, 126

NLG Natural Language Generation. 50, 51, 60, 66, 71, 78–81, 99, 100, 175

NLP Natural Language Processing. 3, 5, 29, 144, 145

NLU Natural Language Understanding. 3, 17, 25, 29, 45, 46, 48–51, 57, 60, 61, 66–68, 71–73, 75, 76, 82, 85, 86, 89, 94, 97, 98, 101, 102, 140, 175, 182

NP Noun Phrase. 73

NUC Next Utterance Classification. 105, 106

PARADISE PARAdigm for Dialogue System Evaluation. 82, 113, 114, 116, 122

PDA Personal Digital Assistant. 13, 26, 181, 182

POMDP Partially Observable Markov Decision Process. 83, 84, 86–88, 120

QoE Quality of Experience. 113, 116, 122

RL Reinforcement Learning. 2, 43, 71, 72, 77, 79, 81, 82, 84–86, 88, 99, 140

RNN Recurrent Neural Network. 69, 76, 80, 127, 132–134, 136, 137, 139, 162, 164, 173, 174

SASSI Subjective Assessment of Speech System Interfaces. 102, 103

Seq2Seq Sequence to Sequence. 71, 81, 100, 104, 106, 125–127, 136, 141, 147, 149

SMS Short Messaging Service. 24

SSA Sensibleness and Specificity Average. 106, 112, 121, 122, 141, 142

SSML Speech Synthesis Markup Language. 51, 52, 69

SVM Support Vector Machine. 75, 120, *Glossary:* SVM

TTS Text to Speech Synthesis. 18, 51, 52, 69, 100, 101, 126

US User Satisfaction. 114, 116, 118

UX User Experience. 103

VoiceXML Voice Extensible Markup Language. 20, 35, 47, 52, 54, 55, 57, 58, 63, 66, 67

VP Verb Phrase. 73

VUI Voice User Interfaces. 19, 20

W3C World Wide Web Consortium. 20

WER Word Error Rate. 29, 97, 112

CHAPTER 1

Introducing Dialogue Systems

A dialogue system is a computer program that supports spoken, text-based, or multimodal conversational interactions with humans. Generally, a distinction is made between task-oriented and non-task-oriented dialogue systems. In task-oriented dialogues the human user and the system engage in an interaction to accomplish some task. Example 1.1 is taken from a demo in which Paul Cutsinger (PC), Head of Amazon Alexa Voice Design Education, initiates a dialogue with Alexa to plan a trip and Alexa asks him a series of questions to obtain the information required to complete the task[1]:

Example 1.1

PC: Alexa, open plan my trip.
ALEXA: Where are you planning to go?
PC: I'm going to Portland.
ALEXA: What city are you leaving from?
PC: Seattle.
ALEXA: What date are you flying out?
PC: Next Thursday.
ALEXA: This will be fun. You go from Seattle to Portland on April 27th, 2017.

In non-task-oriented dialogues the human user and the system engage in general conversational interaction (also known as *chit-chat*). Example 1.2 is taken from a conversation with the Mitsuku chatbot.[2]

Example 1.2

USER: How are you today?
MITSUKU: Ah. Everything is going extremely well. How are you?
USER: Not too bad.
MITSUKU: Glad to hear it. What's new?
USER: The weather is not so great today.
MITSUKU: Ah. Where would you rather be?
USER: Somewhere warm and sunny.

[1]The demo can be viewed in this video https://youtu.be/0V_rNKx-P1I.
[2]Conversation recorded at https://www.pandorabots.com/mitsuku/.

MITSUKU: Where? Not too hot I hope.
USER: Perhaps Spain.
MITSUKU: You seem uncertain. Tell me a little about your life in Spain.

In this example Mitsuku engages in a general conversation. Mitsuku reacts to the user's utterances and prompts for further input, simulating the way humans engage in everyday conversations.

This book is about dialogues such as these and the technologies that are used to design, develop, and evaluate them. The current chapter begins with a brief discussion of what a dialogue system is and why researchers and developers are fascinated with the dream of creating a conversational computer. Section 1.2 provides a brief historical review, describing some of the achievements as well as the limitations of earlier dialogue systems. Section 1.3 looks at examples of current systems and how they are deployed on different types of platform and device. Section 1.4 examines what is required for a system to be able to engage in dialogue with human users and to what extent current systems meet this requirement. Finally, Section 1.5 introduces three different approaches that will be described in greater detail in later chapters: the traditional approach in which systems are developed using handcrafted rules; an approach that emerged in the late 1990s in which aspects of dialogue behavior are learned from data rather than handcrafted; and an approach that has come to dominate current dialogue systems research from around 2014 in which dialogue systems are learned from data in an end-to-end fashion using deep neural networks.

1.1 WHAT IS A DIALOGUE SYSTEM?

Although dialogue systems have been around for a long time, it is only recently that they have become mainstream and a part of everyday life for billions of users. It is generally agreed that dialogue systems came of age in 2011 when Apple launched Siri, a personal assistant that supports spoken interactions with smartphone users. Since then dialogue systems have appeared in various forms, as chatbots on channels such as Facebook Messenger, as personal digital assistants on smartphones, for example, Apple's Siri,[3] Google Assistant,[4] Microsoft's Cortana,[5] and Samsung's Bixby;[6] on smart speakers such as Amazon Echo[7] and Google Nest;[8] and as social robots such as Pepper[9] and Furhat [Al Moubayed et al., 2012].

Various terms have been used to describe the dialogue systems that operate on these devices, including: *Personal Digital Assistant*, *Virtual Personal Assistant*, *Conversational Agent*, *Chat-*

[3] https://www.apple.com/siri/
[4] https://assistant.google.com/
[5] https://www.microsoft.com/en-us/cortana
[6] https://www.samsung.com/global/galaxy/what-is/bixby/
[7] https://www.amazon.com/smart-home-devices/b?ie=UTF8&node=9818047011
[8] https://store.google.com/magazine/compare_nest_speakers_displays
[9] https://softbankrobotics.com/

bot, and *Conversational User Interface*. Indeed, the website *chatbots.org* lists 161 synonyms for Conversational AI systems.[10] There is little consistency in the use of these various terms in the research literature and in the media. For example, Chen et al. [2017] use the term *dialogue system* to describe both task-oriented and non-task-oriented systems, while Jurafsky and Martin [2020] distinguish between *dialogue systems* that engage in conversations with users to help complete tasks, and *chatbots* that mimic the conversations characteristic of casual, social interactions between humans. Sarikaya [2017] prefers the term *Personal Digital Assistant (PDA)* to describe multi-purpose dialogue systems that can answer questions from any domain, help with a variety of goal-oriented tasks, and engage in casual conversation. Others, especially in the media, use the term *chatbot* to describe this sort of system.

Rather than attempting to tease out fine distinctions between all these different terms, it is more productive to focus on what all of the terms mentioned here have in common, i.e., that they provide a new type of interface—a *conversational user interface*—that replaces the traditional graphical user interface [McTear et al., 2016]. So now, instead of responding to text and images on a computer screen by clicking and selecting with a mouse, or on a mobile phone screen by using their fingers to tap, pinch, and scroll, users can interact with an interface that allows them to engage with applications in a conversational manner, i.e., by taking turns as in a dialogue.

1.1.1 WHY DEVELOP A DIALOGUE SYSTEM?

There are several reasons why researchers should wish to develop a dialogue system:

- To provide a low barrier entry for users, enabling them to interact in an intuitive way with services, resources, and data on the internet. With dialogue systems there is no need to learn an interface—in theory, at least. The user can say what they want and the assistant can act as a social companion, providing support and entertainment, or in commercial environments, providing customer self service and automated help.

- From a Cognitive Science point of view to address the challenge of how to model human conversational competence computationally as a means of understanding and studying human behavior and social interaction. The ability to converse in a natural manner, provide relevant responses, and understand the partner's emotional state is one of the high-level cognitive skills that enables social bonding and coordination of actions. Communication is based on the agent's cognitive capabilities such as memory, perception, and the ability to plan and learn. Modeling these capabilities computationally is a key challenge in Cognitive Science research.

- To simulate human conversational behavior so that the dialogue system might pass as a human, as in the Turing test and competitions such as the Loebner prize (see Section 1.2.3). Note, however, that being able to fool humans into believing they are talking to a human

[10] https://www.chatbots.org/synonyms/

is not necessarily a requirement for an effective dialogue system. Moreover, there are also ethical concerns with this approach, as people may feel uncomfortable with a dialogue system that is too human-like or that deceives them into thinking that they are interacting with a human. See, for example, initial reactions to Google's Duplex system that sounded so human-like that some users believed they were talking with another human [O'Leary, 2019]. In order to address these concerns, Duplex now starts each voice call by identifying itself as a virtual assistant from Google.

1.2 A BRIEF HISTORY OF DIALOGUE SYSTEMS

Currently, there is a lot of hype about dialogue systems and conversational user interfaces, but it is important to realize that the idea of creating a conversational computer has been around for a long time. For example, Pieraccini [2012] states that the dream of building a machine that could speak, understand speech, and display intelligent behavior can be traced back at least to the early 1700s, while Mayor [2018] describes how the Ancient Greeks imagined robots and other forms of artificial life, and even invented real automated machines.

Historically there have been five distinct traditions in dialogue systems research involving communities that have largely worked independently of one another. These are:

- Text-based and Spoken Dialogue Systems.
- Voice User Interfaces.
- Chatbots.
- Embodied Conversational Agents.
- Social Robots and Situated Agents.

It will be helpful to review the achievements as well as the limitations of these different traditions and to assess their relevance for dialogue systems research and development.

1.2.1 TEXT-BASED AND SPOKEN DIALOGUE SYSTEMS

The term *dialogue system* is generally used to refer to systems developed in research laboratories in universities and industry with the aim of automating text-based and voice-based interactions between machines and human users.

Dialogue systems that appeared in the 1960s and 1970s were text-based. BASEBALL [Green et al., 1961], SHRDLU [Winograd, 1972], and GUS [Bobrow et al., 1977] are some well-known examples. BASEBALL was a question-answering system that could answer questions about baseball games. The system was able to handle questions with a limited syntactic structure and simply rejected questions that it was not able to answer. SHRDLU was linguistically more advanced, incorporating a large grammar of English, semantic knowledge about objects in its domain (a blocks world), and a pragmatic component that processed non-linguistic information about the domain. GUS was a system for booking flights that was able to handle linguistic phenomena such as indirect speech acts and anaphoric reference. For example, the utterance *I want to go to San Diego on May 28* was interpreted as a request to make a flight reservation, and the utterance *the next flight* was interpreted with reference to a previously mentioned flight. GUS used frames to guide the dialogue, for example, with slots for the values of the travel date, destination and so on that the system had to elicit from the user—a technique that is still used widely in today's task-oriented dialogue systems. See McTear [1987] for an overview of these early text-based dialogue systems.

During the late 1970s and early 1980s, dialogue researchers turned their attention to more advanced aspects of dialogue, such as how to recognize the intentions behind a user's utterances, how to behave cooperatively, and how to deal with different types of miscommunication such as misconceptions and false assumptions [Reilly, 1987]. This work was inspired by philosophers of language such as Grice, Austin, and Searle, as well as research in Artificial Intelligence (AI) on plan recognition and plan generation.

Grice [1975] developed a theory of conversation in which he proposed that participants in conversation are expected to observe the Cooperative Principle (CP) which he formulated as follows:

> *Make your contribution such as is required, at the stage at which it occurs, by the accepted purpose or direction of the talk exchange in which you are engaged.*

Based on the CP Grice proposed four conversational maxims: Quantity, Quality, Relation, and Manner that cover, respectively: how much we should say in a conversation; the truth of what we say; how what we say should be relevant; and how we should communicate clearly. These maxims are still being used widely by dialogue designers as general recommendations for how to design conversations with automated systems, for example, by the conversation designers at the Actions on Google website.[11],[12]

Austin [1962] and Searle [1969] developed a theory of speech acts based on the observation that when people engage in conversation they do more than simply produce utterances—they perform actions. For example, they ask questions, make promises, pay compliments, and so on. One important insight from Speech Act Theory is that the performance of a speech act requires that certain conditions be fulfilled. For example, for an utterance to be intended as a command by a speaker and understood as such by an addressee, various pre-conditions are required, including the following [Searle, 1969]:

- The hearer is able to perform the requested act.

[11] https://designguidelines.withgoogle.com/conversation/conversation-design/learn-about-conversation.html

[12] Note that Grice's main intention was to use the maxims and the CP as a basis for his theory of *conversational implicature* in order to explain how speakers could flout the maxims in order to convey meanings beyond the literal meanings of their utterances (see discussion in Levinson [1983]).

- The speaker believes that the hearer is able to perform the requested act.
- The speaker wants the hearer to perform the act.
- It is not obvious to the speaker and hearer that the hearer would have done the act in the normal course of events.

In the plan-based model of dialogue that became prominent in the 1980s, speech acts such as requests were formalized as *action schemas* similar to those used in AI models of planning [Allen and Perrault, 1980], [Cohen and Perrault, 1979]. This early work on planning in dialogue was the basis for subsequent theoretical work in dialogue technology such as the BDI (Belief, Desire, Intention) model [Allen, 1995], Information State Update Theory [Traum and Larsson, 2003], and the Constructive Dialogue Modeling theory [Jokinen, 2009]. However, one of the problems with the plan-based approach was that it was computationally complex and in the worst case intractable.

Around the late 1980s and early 1990s, with the emergence of more powerful and more accurate speech recognition engines, Spoken Dialogue Systems (SDSs) began to appear, such as: ATIS (Air Travel Information Service) in the U.S. [Hemphill et al., 1990], and SUNDIAL, a major project funded by the European community [McGlashan et al., 1992]. Later systems include MIT's Mercury [Seneff and Polifroni, 2000], Ravenclaw [Bohus and Rudnicky, 2009], and TRIPS [Allen et al., 2001]. These systems were usually restricted to a single domain, such as flight enquiries or train timetable information, where the system had to elicit several items of information from the user such as departure place, date, and time, destination, and so on, in order to construct a query to retrieve a suitable flight or train from a database. The DARPA Communicator systems were an exception as they investigated multi-domain dialogues involving flight information, hotels, and car rentals [Walker et al., 2001]. These systems often suffered from speech recognition errors and so a major focus was on avoiding miscommunication, for example, by employing various strategies for error detection and correction, including different forms of confirmation. See utterances S2, S3, and S5 in Example 1.3, taken from an interaction with the MIT Mercury system that provides telephone access to an online flights database [Seneff and Polifroni, 2000].

Example 1.3 (U=User, S=System)

S1: How may I help you?
U1: I want to travel to Washington Dulles on November fourth.
S2: From Boston to Washington Dulles on November 4. Can you provide the approximate departure time or airline preference?
U2: I need to arrive at around nine a.m.
S3: Okay. Flights from Boston to Washington Dulles arriving at 9:00 am on November 4.
One moment please.

The best matching flight is a US Airways flight arriving at 8:59 a.m. I also have a United flight arriving at 8:36 a.m.
U3: What time does the United flight leave?
S4: United flight 1819 departs at 7:00 am from Boston. Shall I book this flight for you?
U4: Yes please.
S5: United flight 1819 has been added to your itinerary. You depart Boston at 7:00 am and arrive in Washington Dulles at 8:36 am. Where would you like to go next?
U5: I would like to return to Boston.
S6: What date will you be returning on?
U6: The same day.

The dialogue begins with an open-ended question and continues with questions to elicit further information from the user. Generally, the system controls the dialogue by asking questions but the user is also allowed to take the initiative as in U3. This type of interaction is known as *mixed-initiative dialogue* as both the system and the user can ask questions or add additional information. However, although mixed-initiative dialogue offers a more natural style of interaction, this comes at the cost of more complex processing for the Automatic Speech Recognition (ASR) and Natural Language Understanding (NLU) components, increasing the potential for errors.

Example 1.4 shows an interaction that goes wrong, due mainly to speech recognition errors [Bohus, 2007]. The system's interpretation of the user's input is in square brackets and uppercase).

Example 1.4

S1: What city are you leaving from?
U1: Urbana Champaign [OKAY IN THAT SAME PAY].
S2: Sorry I'm not sure I understood what you said. What city are you leaving from?
U2: Urbana Champaign [FOR MINUTE SINCE HEY].
S3: I'm still having trouble understanding you. To go on, I need you to answer the following question. What city are you leaving from?
U3: Chicago [CHICAGO].
S4: Traveling from Chicago. Where would you like to go?
U4: Huntsville [SEOUL].
S5: Traveling to Seoul. What day did you need to travel?
U5: No no I'm traveling to Birmingham [THE TRAVELING TO BERLIN PM].
S6: Traveling in the afternoon. Okay what day would you be departing Chicago?
U6: Uh the tenth of august [AT THE TENTH OF AUGUST].
S7: A flight tomorrow. I have a Korean Airlines flight departing Chicago at 1:40 p.m., arrives Seoul at 5 p.m. the next day.

In this example the system is unable to correctly recognize the user's spoken input for the departure and arrival cities and also makes errors with the time and date. While speech recognition has improved considerably since this dialogue was recorded, it is still the case that the performance of deployed dialogue systems degrades when dealing with unusual accents, interference on the channel, or background noise [Sahu et al., 2018].

In addition to developments in dialogue theory, one of the major contributions of this earlier research was the production of toolkits to support developers of spoken dialogue systems, including: the Center for Spoken Language Understanding (CSLU) Toolkit [Sutton and Cole, 1997], Trindikit [Larsson and Traum, 2000], and DIPPER [Bos et al., 2003].

The CSLU toolkit was developed as an integration of core speech technologies (Automatic Speech Recognition (ASR) and Text to Speech Synthesis (TTS)) along with facial animation and RAD (Rapid Application Development)—a graphically based authoring environment for designing and implementing spoken dialogue systems using a simple drag-and-drop interface. Thus, researchers with little technical knowledge of speech technology could develop simple spoken dialogue systems quickly and with little effort. The toolkit was used widely in academia to support the teaching of spoken dialogue technology [Cole, 1999], [McTear, 1999]. See also Heeman's course on spoken dialogue systems at CSLU.[13] However, over time the toolkit was superseded by other technologies. RAD supported the development of finite state dialogues but could not be easily extended to include dialogues requiring more complex representations. Moreover, the underlying programming language was Tcl/Tk, which is less familiar to developers, while more recent toolkits are based on languages such as Java, Python, and Node.js.

The aim of Trindikit was to support developers wishing to implement dialogue systems involving Information State Update Theory. DIPPER borrowed many of the core ideas from TrindiKit but also simplified the technology in various ways, for example, by using a revised update language and enabling greater flexibility by integrating the system more tightly with the Object Oriented Architecture (OAA).[14] However, this work on representing the dialogue state has been largely superseded by more recent developments in Dialogue state tracking (see Section 3.3.3).

There are also practical reasons why these early toolkits have not been more widely adopted. In some cases the researchers moved to new positions and became engaged in other projects, or funding dried up so that the project could no longer be maintained. Another factor was a major change in research direction in dialogue technology from symbolic to statistical and machine learning-based approaches, so that more recent toolkits are based on the new paradigm, for example, OpenDial [Lison and Kennington, 2016] and PyDial [Ultes et al., 2017]. Finally, as a consequence of the increased interest of major software companies in this area, new toolkits and development platforms have been created that, while incorporating some of the features

[13] https://cslu.ohsu.edu/~heeman/cs550/
[14] http://www.ai.sri.com/~oaa/main.html

of these earlier examples, are generally easier to use, are more readily available, are not tied to proprietary platforms, and are more robust in performance (see Chapter 2, Section 2.3).

Around 2000 the emphasis in spoken dialogue systems research moved from handcrafted systems using techniques from symbolic and logic-based AI to statistical, data-driven systems using machine learning (see further Chapters 3 and 5). For comprehensive overviews of dialogue systems up to around 2010, see McTear [2004], Jurafsky and Martin [2009], and Jokinen and McTear [2009]. For developments since then, see Rieser and Lemon [2011b], McTear et al. [2016], Celikyilmaz et al. [2018], Gao et al. [2019], and Jurafsky and Martin [2020, Chapter 26].

1.2.2 VOICE USER INTERFACES

Alongside the text-based and spoken dialogue systems from academic and industrial research laboratories, various companies and enterprises were developing systems for commercial deployment to support automated telephone-based customer self-service tasks such as directory assistance, information enquiries, and other routine transactions. These systems became known as *Voice User Interfaces (VUI)* and are still being used widely to provide automated customer support over the telephone.

AT&T's How May I Help You (HMIHY) system is an early example [Gorin et al., 1997]. HMIHY supported call routing by classifying customer calls and routing them to the correct destination, thus reducing the costs of human agents and allowing them to handle more complex interactions. For example, by the end of 2001 HMIHY was handling more than 2 million calls per month and showing significant improvements in customer satisfaction over alternative solutions. HMIHY greets users with an open-ended prompt that encourages fairly free-form spoken language input. The system determines the class of the call from a set of 14 possible classes and one additional class for tasks not covered by the 14 classes and that require transfer to a human operator. The system extracts data relevant to the query either from the initial utterance or from a sub-dialogue. For example: given the user query *I have a query about my bill for October* the system would classify the query as the type **Ask(Bill)** and extract the item **Date** with the value *Ask(October)*. In many cases the dialogues are short: for example, 23% of the dialogues consisted of only two exchanges in a corpus of 4692 dialogues collected in several experimental trials of the system in use with real customers [Walker et al., 2002]. Example 1.5 shows a successful interaction with the HMIHY system [Walker et al., 2002]:

Example 1.5

S1: AT&T: How may I help you?
U1: I need to [uh] put a call on my calling card please.
S2: May I have your card number please?
U2: 7 6 5 4 3 2 1 0 9 8 7 6 5 4
S3: What number would you like to call?

U3: 8 1 4 7 7 7 6 6 6 6 (misunderstood)
S4: May I have that number again?
U4: 8 1 4 7 7 7 6 6 6 6
S5: Thank you.

This example illustrates a more extended dialogue in which, following the classification of the call, the system enters a sub-dialogue to collect values for a set number of attributes relevant to this type of call. The example also illustrates how the system can recover from a misunderstood utterance (U3-S4).

The following are some of the achievements of the Voice User Interface community:

- the development of design and evaluation guidelines for commercially deployed VUI [Cohen et al., 2004], [Lewis, 2016a], [Pearl, 2016];
- the development of standards, such as Voice Extensible Markup Language (VoiceXML)[15,16] for scripting spoken dialogues, and EMMA (Extensible Multimodal Annotation Markup Language)[17] for annotating multi-modal user input (see further, Chapter 2). For a recent book on World Wide Web Consortium (W3C) standards for VUI, see Dahl [2017];
- toolkits such as Voxeo Evolution[18] for developing VoiceXML-based applications;
- speech analytics: the process of mining recorded conversations between a company's service agents and customers to obtain information about the quality of the interaction, agent performance, customer engagement, and other factors that determine customer satisfaction and loyalty;[19] and
- usability testing: the application of effective metrics and methods for testing the usability of VUI [Hura, 2017].

1.2.3 CHATBOTS

Chatbots, also known as chatterbots, were created originally as attempts to simulate human conversations. ELIZA is generally viewed as the first chatbot [Weizenbaum, 1966]. ELIZA simulates a Rogerian psychotherapist, often in a convincing way, and has inspired many generations of chatbot authors for whom a major motivation is to develop a system that can pass Turing's Imitation Game [Turing, 1950]. The aim of the Imitation Game is to see if a machine can display intelligent behavior by fooling observers of a conversation between a human and a

[15] https://www.w3.org/TR/voicexml20/
[16] https://www.w3.org/TR/voicexml21/
[17] https://www.w3.org/TR/emma/
[18] https://evolution.voxeo.com/
[19] https://www.aspect.com/globalassets/10-best-practices-for-speech-analytics-wp.pdf

chatbot into thinking that the utterances from the chatbot were actually from another human participant. The Loebner Prize Competition,[20] launched in 1991 by Dr. Hugh Loebner, is an implementation of the Turing test. In this competition human users (known as the "judges") take part in text-based conversations on computer terminals with two different unseen conversational partners, one of which is another human (known as the "confederate") and the other a chatbot. After 25 minutes of questioning the judge must decide which conversational partner is the human and which is the chatbot. If a system can fool half the judges that it is human under these conditions, a solid Silver Medal is awarded to the creator of that chatbot, otherwise prizes are awarded to the creators of the chatbots according to the ranked scores of the judges.

The chatbot Mitsuku, which was introduced at the beginning of this chapter, has won the Loebner prize five times. Mitsuku was developed by Steve Worswick using the dialogue scripting language Artificial Intelligence Markup Language (AIML) (see Chapter 2, Section 2.3.2). Examples of chat logs with Mitsuku and additional information can be found at the Mitsuku chatbot website.[21]

Chatbots are being used increasingly in areas such as education, information retrieval, business, and e-commerce, where they act as automated online assistants to complement or even replace the services provided by humans in call centers.

Traditionally, chatbots like Mitsuku as well as the business chatbots developed for e-commerce have been handcrafted using scripting languages such as AIML[22] and ChatScript.[23] A new development that has become a hot topic in Conversational AI is to train open-domain chatbots such as Google's Meena [Adiwardana et al., 2020], Facebook's BlenderBot [Roller et al., 2020], and Open AI's Generative Pre-Training (GPT) models[24] from very large datasets of conversations using neural dialogue technologies (see further Section 1.4.3, and Chapter 5). The achievements of the chatbot community include the following:

- the development of scripting languages such as AIML and ChatScript;
- toolkits and platforms, for example, Pandorabots[25] and PullString;[26]
- advances in technology, such as the use of knowledge repositories to provide some degree of world knowledge as well as discourse mechanisms to provide limited support for and topic tracking;
- the incorporation of mobile functions to enable the deployment of chatbots on smartphones and other smart devices;

[20] https://en.wikipedia.org/wiki/Loebner_Prize
[21] http://www.square-bear.co.uk/mitsuku/home.htm
[22] http://www.aiml.foundation/
[23] https://sourceforge.net/projects/chatscript/
[24] https://openai.com/blog/openai-api/
[25] https://home.pandorabots.com/home.html
[26] https://www.facebook.com/pullstringinc/

- machine learning of conversational patterns from corpora of conversational data [Shawar and Atwell, 2005]; and
- within the past few years the use of neural dialogue technologies to train open-domain chatbots from large datasets of dialogues (see Chapter 5).

1.2.4 EMBODIED CONVERSATIONAL AGENTS

An Embodied Conversational Agent (ECA) is a computer-generated animated character that combines facial expression, body stance, hand gestures, and speech to provide a more human-like and more engaging interaction [André and Pelachaud, 2010],[Cassell et al., 2000]. An ECA takes the form of virtual agents and screen-based characters. Examples are:

- Smartakus, an animated character used in the SmartKom project to present information [Wahlster, 2006];
- REA, a real-time, multi-modal, life-sized ECA that plays the role of a real estate agent [Bickmore and Cassell, 2005]; and
- GRETA, a real-time three dimensional ECA that talks and displays facial expressions, gestures, gaze, and head movements [Niewiadomski et al., 2009].

The achievements of the ECA community include the following:

- advances in technology, such as: how to handle multi-modal input and output, the development of avatars and talking heads, and the production and interpretation of gestures and emotions;
- the development of standards and annotation schemes, such as SAIBA (Situation, Agent, Intention, Behavior, Animation), BML (Behavior Markup Language), FML (Functional Markup Language), MURML (Multi-modal Utterance Representation Language), and EML (Emotion Markup Language). See Dahl [2017] and [Jokinen and Pelachaud, 2013] for descriptions of many of these standards; and
- toolkits, for example, the Virtual Human Toolkit [Gratch et al., 2013] and ACE (Articulated Communicator Engine) [Salem et al., 2010].

For more detailed descriptions of ECAs, see McTear et al. [2016], especially Chapters 13–16.

1.2.5 ROBOTS AND SITUATED AGENTS

Social robots are becoming increasingly popular as companions for the elderly, as educational and entertainment toys for children, as self-service aids in public places, and more. Social robots allow users to perform tasks similar to those provided by a virtual personal assistant on a smartphone or smart speaker. Additionally, because of their physical embodiment, they are expected to possess social qualities such as the ability to recognize and display emotions, and other human-like social cues [Graaf et al., 2015].

Pepper is an example of a social robot that can recognize emotions based on characteristics of the user's voice, facial expression, and body movements.[27] Pepper can adapt its behavior to suit the situation and display empathy. For example, Pepper is used to greet guests in a hotel lobby to perform check-in either through dialogue or by getting the guest to use a touchscreen on its chest, which is useful if Pepper is unable to understand the guest's spoken input. Pepper can also engage in basic conversation. For example, in the hotel lobby scenario it can handle questions about room confirmation or enquire whether the guest requires help with their bags.

Mummer (MultiModal Mall Entertainment Robot)[28] is a four-year European Union (EU)-funded project with the overall goal of developing a social robot based on Softbank's Pepper platform [Foster et al., 2019]. The robot is designed to interact with users in a public shopping mall in a natural and engaging manner, combining spoken interaction with non-verbal communication and human-aware navigation. The technologies being investigated include: audio-visual sensing, social signal processing, conversational interaction, perspective taking, geometric reasoning, and motion planning.

SARA (Socially Aware Robot Assistant), developed in Carnegie Mellon University's ArticuLab, also recognizes and displays emotions [Matsuyama et al., 2016]. SARA studies the words a person says during a conversation as well as the tone of their voice, and feeds these cues into a program that determines an appropriate response designed to build a feeling of rapport with a person and to improve task performance.

Other examples of social robots are Professor Einstein, a physics tutor,[29] Leka, a robot that provides help and companionship for children with autism,[30] and Furhat, a robotic head based on a projection system that renders facial expressions, with motors to move the neck and head [Al Moubayed et al., 2012].

1.2.6 LIMITATIONS OF EARLY DIALOGUE SYSTEMS

While there is much to be learned from the achievements of these early dialogue systems, in many cases they suffered from one or more of the following limitations:

- they were often extremely brittle and would fall over or crash if there was the slightest deviation from the expected input;
- the systems worked well for the purposes for which they were designed but did not scale up or transfer easily to other domains;
- dialogue decisions were handcrafted and could not be guaranteed to be optimal;

[27] https://www.wired.com/story/pepper-the-humanoid-robot/
[28] http://mummer-project.eu/
[29] https://www.hansonrobotics.com/professor-einstein/
[30] https://www.leka.io/

- the systems were often developed using proprietary toolkits and languages that were not always openly available and that were often based on particular theories of dialogue;
- they were deployed on specialized platforms and could not be easily ported to other domains or deployed on other platforms; and
- they focused only on spoken or written language and didn't consider other modalities that are important in natural communication.

Many of these issues have been addressed in the current generation of dialogue systems.

1.3 PRESENT-DAY DIALOGUE SYSTEMS

Whereas dialogue systems previously existed either on specially dedicated servers in academic and industrial research laboratories or as telephone-based voice user interfaces, now they can be encountered on a wide variety of platforms and devices that are available to the general public. Dialogue systems can take the form of messaging apps on smartphones, PCs, and tablets; they can act as personal digital assistants on smartphones; and more recently they are to be found as voice-based assistants on smart speakers. We can also have dialogues with social robots and with smart devices in the home, in the car, and elsewhere.

1.3.1 DIALOGUE SYSTEMS ON MESSAGING PLATFORMS

In his keynote address at Microsoft Build 2016, Microsoft CEO Satya Nadella announced that "chatbots are the new app". Also in 2016, Facebook launched their Messenger-based chatbot platform. Suddenly the term *chatbot* re-emerged as the label for a new type of user interface that allows users to interact with services and brands using a conversational interface on their favorite messaging apps [Shevat, 2017].

One of the advantages of chatbots is that they can run on messaging applications such as Facebook Messenger, Telegram, Slack, Skype, Line, and WhatsApp that are widely used by millions of people to interact with friends, colleagues, and the services of companies. This means that it is not necessary to download and install a different app for each new service. Furthermore, since chatbots live within messaging applications, there is no need to worry about platform issues, as each chatbot can be available on all operating systems that are supported by the messaging app. In contrast, native mobile apps have to be adapted or rewritten for each mobile operating system and they need to be frequently updated to keep up with upgrades to the host system and its features. Since chatbots are implemented server-side, any updates can be propagated almost immediately to all users. The chatbot interface is similar to text messaging (SMS), except that the interaction takes place synchronously in real time and the other participant in the conversation is a chatbot and not a human. Generally, chatbot dialogues on messaging platforms are system-led and the user's responses are often limited to clicking on buttons containing pre-defined words and phrases (known as *Quick Replies* or *Suggestion Chips*). In

Figure 1.1: A chatbot with Quick Replies. Used with permission.

some cases the user can also type in a few words from a fixed set of possible inputs. Figure 1.1 shows an example from the iHelpr chatbot that provides guided self-assessment and advice in areas of mental health [Cameron et al., 2018]. Natural Language Understanding (NLU) can be used in more advanced systems to interpret the user's free text inputs, giving the user the opportunity to "say what they want and how they want", without being restricted to a fixed set of commands or queries. For example, in iHelpr Microsoft's Language Understanding Intelligent Service (LUIS)[31] was used to extract intents from the free-form utterances of the users.

In many simple chatbots the output takes the form of text, although nowadays many chatbot platforms support rich messages for delivering audio, video, images, maps, charts, buttons, emojis, and persistent menus.

The dialogue flow can vary across applications. In the simplest cases the entire dialogue flow is pre-determined and designed using a graphical flow chart (see Chapter 2). In other cases the dialogue flow may be more open-ended and evolve dynamically according to the context.

[31] https://www.luis.ai/

Chatbot conversation on messaging platforms is thread-centric. Messages are grouped automatically according to the sender, so that the thread of a conversation can be maintained and users can easily locate all the messages from a particular sender. In this way the dialogues maintain some permanence in contrast to voice-only dialogues that are transient.

Most chatbots are designed to connect to a specific service, such as news, weather, hotel bookings, or flight reservations. A notable exception is WeChat, a Chinese multi-purpose chatbot developed by Tencent.[32] With WeChat it is possible to accomplish many tasks, such as ordering flowers, making dinner reservations, ordering pizza, and making payments, all within the same interface.

1.3.2 DIALOGUE SYSTEMS ON SMARTPHONES

Dialogue systems on smartphones are often known as Personal Digital Assistants (PDAs) [Sarikaya, 2017] or Voicebots [Batish, 2018]. Examples include Apple's Siri, Microsoft's Cortana, Google Assistant, Samsung's Bixby, and others. A PDA on a smartphone supports a range of modes of interaction, including text input and output, speech-to-text, text-to-speech, as well as direct manipulation, for example, by allowing the user to make selections from a set of options by tapping. A PDA can also display images, and play audio or video clips.

PDAs on smartphones can also make use of information about the user, for example, user preferences, location, and information from sensors that has been collected on the phone. This enables the PDA to provide more intelligent and more personalized assistance, both proactively (for example, to provide reminders) and reactively (for example, to respond to the user's queries). Ideally, according to Sarikaya [2017], PDAs should be able to answer questions from any domain by accessing a range of different knowledge sources. They should also support the execution of goal-oriented tasks and they should have the ability to engage in conversations involving chit-chat. Current PDAs only support these functionalities to a limited extent.

Dialogues with PDAs can take a variety of different forms. When answering the user's questions, the dialogue takes the form of a *one-shot exchange* involving the user's question and the system's response. More recently, follow-up questions have been supported (see Section 1.4.1 and Chapter 2, Section 2.3.2). For task-oriented dialogues, once the user has activated a particular service, the system takes over and asks a series of questions to elicit required information from the user—for example, with a hotel booking application the dates of the booking, how many guests, the type of room, etc. This is known as *slot-filling*, since the information required to complete the transaction is gathered into a data structure containing a number of slots to be filled. Finally, in chit-chat type interactions the dialogue may continue for as long as the user wishes, resulting in a *multi-turn dialogue*.

[32] https://web.wechat.com/

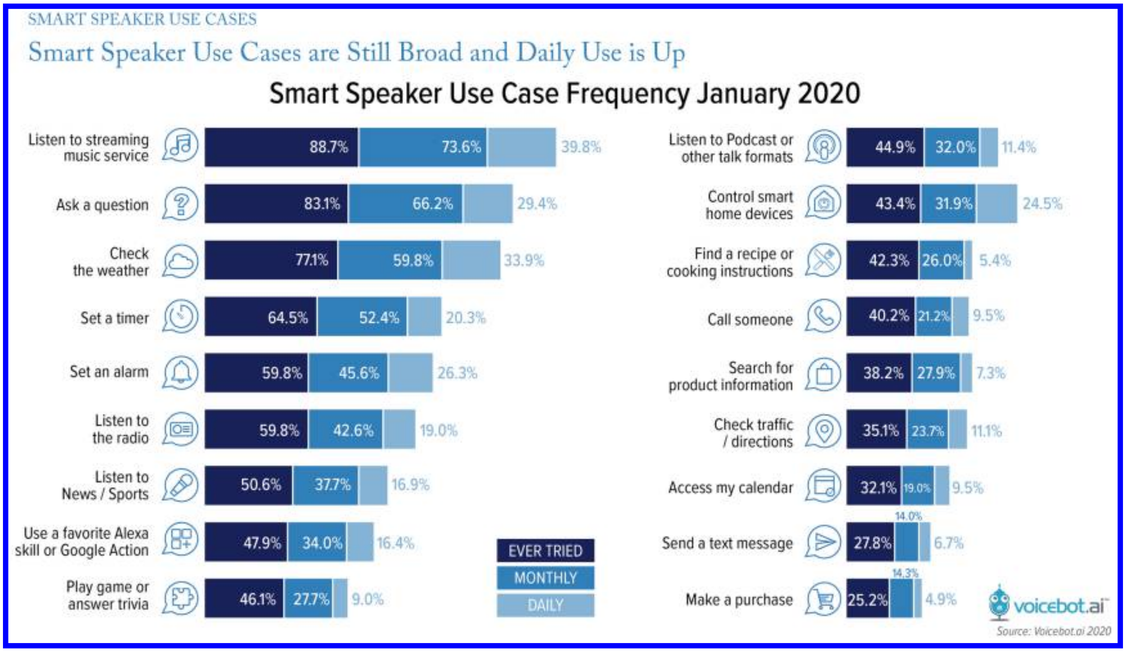

Figure 1.2: Smart Speaker Use Case Frequency, January 2020. Source: Voicebot Smart Speaker Consumer Adoption Report Executive Summary p. 9, April 2020. Used with permission.

1.3.3 DIALOGUE SYSTEMS ON SMART SPEAKERS AND OTHER DEVICES

Dialogues on smart speakers with assistants such as Amazon Alexa or Google Assistant are similar to those provided by PDAs on smartphones except that on some of the devices the interaction mode is voice-only. Naturally this places some restrictions on the interactions as everything has to be communicated by voice, including dealing with recognition issues and other problems that previously also affected telephone-based voice user interfaces. Recently, both Amazon and Google have released smart speakers with displays, such as Amazon Echo Show and Google Nest Hub in which voice interaction is integrated with a visual display on touch-sensitive screens. Smart speakers have become extremely popular. In a recent survey it was reported that in 2019 nearly 1 in 3 U.S. adults, i.e., 88.7 million adults, have access to a smart speaker.[33] Smart speakers are used for a wide variety of tasks, as shown in Figure 1.2. The most frequent tasks are similar to those provided by assistants on smartphones, such as requesting music, asking questions, and checking the weather. In the future it is predicted that voice assistants will be used increasingly to control smart appliances in the home such as thermostats, cookers, and dishwashers. For example, at CES2019, the world's largest exhibition for new consumer technologies, various

[33] https://research.voicebot.ai/download-smart-speaker-consumer-adoption-2020-executive-summary/

voice-activated devices were displayed, such as an Alexa-activated toilet flush, voice-controlled pianos, heart rate monitors, lawnmowers, and motorcycle helmets, and for the kitchen, a smart speaker with display providing visual and audio walkthroughs of cooking recipes.

Dialogue is also available on a number of wearable devices. For example, smart watches provide many of the functions that are also available on smartphones, such as notifications, messaging, navigation, and search. Users can speak to their smart watches as well as tapping on items on the screen and swiping displayed cards. Compared with smartphones the display on a smart watch is much smaller. This has a bearing on how much information can be displayed visually. See McTear et al. [2016], Chapter 13 for more discussion of dialogues with wearable devices.

1.3.4 DIALOGUE SYSTEMS IN CARS

Voice-based systems have become a standard feature in many new vehicles, motivated primarily by a need to keep people safe by allowing drivers to communicate with devices in the car without taking their eyes off the road or their hands off the steering wheel. Drivers can obtain directions, send emails, make phone calls, and play music using voice commands.

There are several voice-based systems for cars, including Apple CarPlay,[34] which contains a fully integrated version of Siri; Google Android Auto;[35] Nuance Dragon Drive,[36] and several brand-specific devices.

Dialogue in cars is more or less limited at present to voice commands that activate and control some of the car's features, such as environmental controls. Recently, Amazon Alexa and Google Assistant have been integrated into certain cars. In Fiat Chrysler cars Alexa can be used to start the car remotely, lock and unlock the doors, find the nearest gas station, say how much gas is left in the car, and provide information about the car's type pressures.[37] Similar commands are available in Ford cars. Google Assistant is integrated into the Android Auto smartphone projection technology and is available in a number of cars, including Acura, Alfa Romeo, Audi, Hyundai, Jeep, Kia, and Mercedes-Benz.[38] Google Assistant can be used to carry out several tasks, including navigation, finding gas stations, consulting the driver's calendar, reading incoming text messages, issuing reminders from the calendar, playing music and accessing radio stations, providing opening hours of businesses and restaurants and providing information on topics such as music and sports. It is also possible with Alexa and Google Assistant to control devices in the home such as thermostats, lights, and intruder alarms remotely from the car.

[34] https://www.apple.com/uk/ios/carplay/
[35] https://www.android.com/auto/
[36] https://www.nuance.com/about-us/newsroom/press-releases/nuance-announces-new-ai-powered-dragon-drive-features.html
[37] https://www.amazon.com/FCA-Chrysler/dp/B07DD2NSSW
[38] https://assistant.google.com/platforms/cars/

1.3.5 HOW CURRENT DIALOGUE SYSTEMS ARE DIFFERENT

Current dialogue systems have addressed many of the limitations of earlier systems:

- they can be developed and deployed on messaging apps such as Facebook Messenger, Slack, or Skype that users are already familiar with;
- they work seamlessly across multiple devices and platforms;
- the user does not need to download and install separate apps for each application;
- in many cases the systems have access to contextual information about users, such as their location, health, and other data that may have been acquired through sensors. This allows them to provide a more personalized experience for each user;
- the systems can often learn from experience in contrast with earlier systems that were static and did not alter or improve their behavior over time;
- many systems, especially robot agents, have multi-modal interaction capability and they can effectively analyze gaze signals, gesturing, nodding, and body posture. Generating appropriate multi-modal behavior has been extensively studied with ECAs and social robots; and
- a number of toolkits have become available that incorporate the latest developments in Artificial Intelligence (AI), Machine Learning (ML), and Natural Language Processing (NLP), and provide an intuitive and easy-to-learn resource for developers (see Chapter 2, Section 2.3.3).

There are also various technological drivers that have facilitated the development and deployment of this new generation of dialogue systems:

- advances in ASR, driven by the application of deep learning and resulting in dramatic reductions in Word Error Rate (WER), making spoken dialogue systems really possible;
- advances in NLU, also as a result of the application of deep neural networks;
- greater computing processing power to support the massive parallel computations required to run deep neural networks;
- the availability of vast amounts of data that enable AI systems to learn and become increasingly more intelligent;
- increased connectivity, allowing users to connect their smart devices to vast cloud-based resources;
- advances in computer vision, eye-tracking, and video processing; and

- the interest of the major technology companies in chatbots and conversational interfaces, enabling them to more accurately profile their users and thus gain a competitive advantage in the promotion of their e-commerce services. For example, the global research and advisory firm Gartner has predicted that 25% of all customer services operations will use virtual customer assistants by 2020.

1.4 MODELING CONVERSATION IN DIALOGUE SYSTEMS

Conversational interactions in current systems fall into three distinct types in terms of the types of interaction they support and which participant initiates and controls the dialogue.

- *User-initiated dialogues*: Interactions initiated by the user are typical of the way users interact with smart speakers and virtual assistants. The interaction is usually brief, consisting of a two-turn exchange in which the user asks a question or issues a command and the system responds.
- *System-directed dialogues*: In these interactions the system controls the dialogue. There are several types of system-directed dialogue:
    1. dialogues in which the system initiates the interaction proactively, for example, to deliver a reminder to a care recipient to take their medication;
    2. dialogues initiated by a user seeking instructions, for example, in an online recipe application, where the system provides a set of instructions with little input from the user, except to ask for the next instruction or for an instruction to be repeated; and
    3. dialogues initiated by a user requesting a service, for example, to make a hotel reservation. The system takes control of the interaction and asks a series of questions to determine the user's requirements and help complete the task. This is known as a *slot-filling dialogue*.
- *Multi-turn open-domain dialogues*: These are extended interactions in which both the system and the user take turns as in natural conversations between humans, where the conversation extends over potentially many turns and can involve a range of topics. This type of dialogue is generally not supported in currently deployed systems[39] but is the focus of much research in Conversational AI.

1.4.1 USER-INITIATED DIALOGUES

Examples 1.6 and 1.7 show one-shot exchanges in which the user asks a question or issues a command and the system responds.

[39] A notable exception is XiaoIce [Zhou et al., 2020], a social chatbot developed in China by Microsoft that can engage in extended dialogues with users.

Example 1.6 Asking for information

U: What's the weather forecast for tomorrow in Frankfurt?
S: In Frankfurt tomorrow it will be mostly sunny with a high of 23 and a low of 9

Example 1.7 Issuing a command

U: Set the alarm for 7 tomorrow morning.
S: Alarm set for 7 a.m. on Friday August 28th.

There are certain requirements for one-shot exchanges to be successful.

- The user's input has to be in a form that the system can interpret.
- The user has to include all the necessary information in a single utterance for the system to be able to interpret it successfully and answer the question or fulfil the command.
- The user needs to be able to deliver the utterance without any disfluencies.[40]
- The system needs to be able to answer the query or be able to carry out the command.

Recently, some dialogue platforms such as Amazon Alexa and Google Assistant have extended one-shot exchanges by allowing the user to make a follow-up query or command. These take two different forms:

- slot replacement and
- anaphora resolution

Slot Replacement

Here the follow-up utterance is the same as the original utterance but with a different slot value. Examples 1.8 and 1.9 are from interactions with Google Assistant:

Example 1.8

U1: What's the weather in Berlin?
S1: Currently in Berlin there is rain and a high of 10 degrees.
U2: What about Frankfurt?

[40]Note that some systems such as Google Assistant can successfully handle a certain amount of disfluency in the user's utterance.

In Example 1.8 the utterance about the weather in Berlin is replaced by an utterance about the weather in Frankfurt, i.e., the value of the location slot in the original utterance has been replaced. Example 1.9 is a longer interaction involving a series of follow-up utterances with slot replacements:

Example 1.9

U1: What's the weather forecast for Frankfurt for tomorrow?
S1: In Frankfurt tomorrow it will be mostly sunny with a high of 23 and a low of 9.
U2: What about Belfast?
S2: In Belfast tomorrow there will be scattered showers with a high of 17 and a low of 12.
U3: I meant London.
S3: Tomorrow in London it will be cloudy with a high of 19 and a low of 14.
U4: What about Sunday.
S4: In London on Sunday it will be partly cloudy with a high of 22 and a low of 12.

Anaphora Resolution

Anaphoric reference is similar to slot replacement except that in the follow-up utterance an entity in a slot is referred to using an anaphoric reference, such as a pronoun or a word such as *there*, as shown in Example 1.10, taken from a dialogue with Amazon Alexa:

Example 1.10

U1: What's the weather in London?
S1: In London it's 8 degrees with mostly cloudy skies.
U2: What's the population there?
S2: The population of London is about eight million seven hundred and ninety thousand.

Anaphora resolution is a very hard problem, especially in long multi-turn conversations, as it requires contextual inference to find expressions that refer to the same entity in current or past utterances [Khatri et al., 2018a]. In the Alexa Prize 2018 most teams used StanfordCoreNLP's Coreference Resolution System for anaphora resolution [Manning et al., 2014].

One-shot exchanges that complete successfully in this way follow a so-called *happy path* in which the user behaves as expected. But what happens when the utterances do not follow the happy path? The following are some use cases.

- The system is unable to interpret the user's utterance and says something like *Sorry I do not understand* and asks the user to repeat.

- The system does not know the answer to the user's question and says something like *Sorry I do not know the answer to that*.
- The system is not able to or does not know how to carry out the user's command and says something like *Sorry I can't do that* or *Sorry I do not know how to do that*.
- There is something missing or ambiguous in the user's utterance and the system inserts a request for clarification, as in Example 1.11:

  Example 1.11

  U: I am looking for a restaurant nearby.

  S: What kind of food would you like?

Most systems are able to handle use cases such as these in a fairly simple way. A more advanced and more helpful approach would be to guide the user, for example, by saying something like *I can answer questions about sport and entertainment*. Responses such as these and other responses such as clarification requests are usually handcrafted to meet the requirements of a specific use case but are difficult to apply more generally.

1.4.2 SYSTEM-DIRECTED DIALOGUE

Pro-Active Dialogues

Pro-active dialogues are an extensive of push notifications on smart phones where a message is sent to the user to remind them of an upcoming meeting, or in the case of a care receiver, a reminder to take medication. In contrast to push notifications pro-active dialogues engage the user in a dialogue, thus ensuring that they have received the reminder. As an example, LifePod[41] provides a proactive voice-based service that contacts care recipients such as elderly adults, chronically ill or special needs users in their homes to check-in, issue reminders, and engage in care plan dialogues. Figure 1.3 shows an example of a template for a simple proactive dialogue in which the system checks whether the user has taken water to prevent dehydration. This dialogue only requires a *yes* or *no* response from the user. In the future more extended dialogues could involve discussion of routines, medication plans, etc.

Instructional Dialogues

Car navigation is a common example of an instructional dialogue in which the system gives step-by-step directions to a specified destination. In currently deployed systems no user input is possible except in some cases to initiate the directions by stating the destination. Another example is cooking instructions on smart speakers in which the system helps users follow the steps

[41] https://lifepod.com/

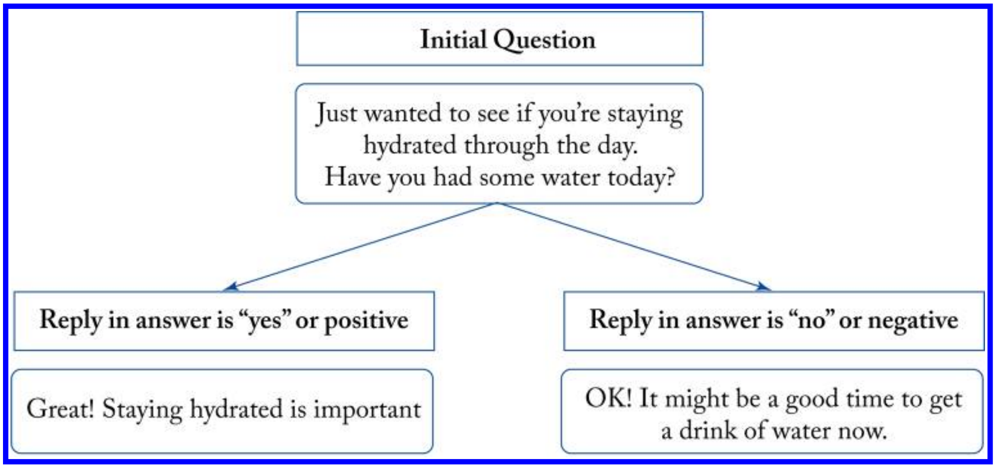

Figure 1.3: Template for a hydration reminder. Based on description in Patterson [2020, p. 125].

in a recipe hands-free by speaking step-by-step instructions, accompanied by video instructions in the case of smart speakers with displays. Two types of user input are typically supported[42]:

1. commands to navigate the recipe, e.g., *next step*, *previous step*, *what is step 5?*, and commands and queries about ingredients, e.g., *next ingredient*, *how much butter?*, etc.; and

2. general questions about cooking, e.g., *can I replace soda with baking powder?*, *how much is that in grams?*, etc.

Amazon Alexa supports similar skills, see, for example, the BBC Food Alexa Skill,[43] and the Allrecipes skill.[44]

Slot-Filling Dialogues

Slot-filling dialogues are similar to form-filling on the web. The system asks a series of questions to collect slot values for **destination**, **departure_date**, **departure_time**, and so on that are required in order to complete a task such as booking a flight, as in Example 1.12:

Example 1.12

U1: I want to book a flight.
S1: where do you want to fly to?
U2: Boston.
S2: what date do you wish to depart?

[42]Example taken from the Google Assistant app: https://support.google.com/googlenest/answer/7309433.
[43]https://www.bbcgoodfood.com/howto/guide/how-use-bbc-good-food-alexa-skill
[44]https://www.allrecipes.com/article/introducing-allrecipes-on-amazon-alexa/

S3-N: (System continues to ask questions until all the slots are filled).

One advantage of the slot-filling approach is that it provides some flexibility in the dialogue. The user can simply state their requirements without specifying any required values, as in the previous example, or they can specify several of the slots in a single utterance (this is known as *over-answering*), as in Example 1.13:

Example 1.13

U: I want to fly from London to Boston on Friday.

In this case the slots for **departure_city**, **destination**, and **departure_date** have been filled, so the system only needs to ask about any remaining unfilled slots, such as **departure_time**. This type of interaction is supported in VoiceXML and other tools that are reviewed in Chapter 2, Section 2.3.

A possible deviation from the happy path of a slot-filling dialogue is where the user requests repetition or clarification, as in 1.14:

Example 1.14

S: What kind of food would you like?
U: What are my choices?

Interpreting and responding to clarification requests from the user are usually handcrafted into slot-filling dialogues in current systems as special cases that have been predicted at design time.

1.4.3 MULTI-TURN OPEN-DOMAIN DIALOGUE

Multi-turn open-domain dialogue is more like conversations between humans in which the user and the system can engage in an extended interaction, where both participants can take the initiative and the dialogue is not restricted to a particular domain. Open-domain conversational systems have become a hot topic in Conversational AI. As Guo et al. [2018] write:

> *Achieving sustained, coherent and engaging dialog is the next frontier for Conversational AI, …*

Similarly, Adiwardana et al. [2020] state that

> *The ability to converse freely in natural language is one of the hallmarks of human intelligence, and is likely a requirement for true artificial intelligence.*

The following are some recent approaches to multi-turn open-domain dialogue.

Table 1.1: Open Request Pattern and Example. Source: Moore and Arar [2019, p. 93].

Open Request Pattern	Open Request Example
U1: FULL REQUEST	U1: Can you recommend a nearby Mexican restaurant?
S1: GRANT	S1: Mario's is located at Beach and Main.
U2: SEQUENCE CLOSER	U2: Thanks.
S2: RECEIPT	S2: You're welcome!

The Natural Conversation Framework

The Natural Conversation Framework (NCF) [Moore and Arar, 2019] is a design framework for dialogue systems based on patterns of conversational interaction that have been identified and documented in the Conversation Analysis (CA) literature [Schegloff, 2007]. One hundred patterns are documented that cover different types of conversational interactions, such as Inquiry (User), Open Request, and Extended Telling. There are also patterns for sequence-level management, such as different types of repair, and patterns for conversation-level management, such as openings and closings.

Table 1.1 shows the basic Open Request pattern along with an example. Note that in contrast to the one-shot exchanges typical of user-initiated interactions with smart speakers the pattern specifies two additional turns beyond the initial exchange in which the user closes the sequence by saying "thanks" and the system acknowledges. This expansion of the basic one-shot exchange enables the user to demonstrate that they have understood the system's response and are satisfied with it and it also allows the system to accept the user's acknowledgement and close the exchange. In this way mutual understanding between the conversational partners is achieved and acknowledged. Achieving mutual understanding is an important objective in conversational interaction and motivates conversational turns that are not part of the informational and transactional elements of a conversation. This principle has been adopted recently by some current dialogue systems that respond appropriately to user follow-ups such as *thanks*, *goodbye*, or *see you later* by saying something like *don't mention it* or *it's my pleasure*.

Sequences can be expanded if the system or the user requests more detail or asks for repetitions or paraphrases. Table 1.2 shows how a variation on the pattern in Table 1.1 is expanded by the system asking for more detail.

Example 1.15 shows how a basic exchange can be expanded in many ways with requests for further detail and paraphrases.

Example 1.15

U1: Do you know about restaurants?
S1: I know a few places.
U2: A few what?

Table 1.2: Open Request Detail Request Pattern and Example. Source: Moore and Arar [2019, p. 96].

Open Request Agent Detail Request Pattern	Open Request Agent Detail Request Example
U1: PARTIAL REQUEST	U1: Can you recommend a Mexican restaurant?
S1: DETAIL REQUEST	S1: Walking distance or car ride?
U2: DETAIL	U2: Walking distance.
S2: GRANT	S2: Mario's is located at Beach and Main.
U3: SEQUENCE CLOSER	U3: Thanks.
S3: RECEIPT	S3: You're welcome!

S2: Places.
U3: Can you recommend one close by?
S3: What kind of food do you want?
U4: Well, what are my choices?
S4: Around here Mexican, Japanese or Indian.
U5: Mexican.
S5: Okay. Mario's is on Beach and Main.
U6: What do you mean?.
S6: Mario's is a Mexican restaurant just a couple of blocks west on Chester Street.
U7: Oh ok, thanks
S7: You're welcome! Anything else?

U1 is a preliminary expansion that checks on the conditions required for the agent to be able to answer the upcoming query in U3. There are several inserted expansions: U2 is a user-initiated request for clarification or additional information, as is U4, while S3 is a system-initiated request for additional information, similar to a slot-filling question. Note that in addition to bringing the dialogue to a potential close in S7, the system asks if the user wants anything else, a question typically asked by agents at the end of service encounters that opens up the possibility of further interaction.

Sequence expansions enable conversational systems to adapt to the particular user on a local, turn-by-turn basis. One user may be able to complete the sequence without any sequence expansions, while another user may require many. In this way sequences cannot be pre-determined but evolve on a turn-by-turn basis as a result of the interactional work by the participants in the conversation as they aim to achieve their goals and demonstrate mutual understanding. The NCF provides patterns that enable the flexibility required in multi-turn open-domain dialogues. The many examples presented in Moore and Arar [2019] are taken from a conversational agent Alma that was implemented using NCF on the IBM Watson Assistant service and could in principle be implemented on other platforms.

The Amazon Alexa Prize

The Alexa Prize was set up by Amazon in 2016 as a challenge to student teams from universities to create a *socialbot* that could converse with members of the general public on a wide range of topics. The first challenge was held in 2017 [Khatri et al., 2018b] and the second in 2018 [Khatri et al., 2018a]. The third challenge was launched in 2019 and the finals were held in 2020.[45] The technologies developed by the teams in the Alexa Prize 2020 are reviewed in Chapter 5, Section 5.6.

The idea behind socialbots is that, like chatbots, they should be able to handle chit-chat but in addition they should also be able to talk with users about recent news and other topics of interest. Like traditional chatbots they should be reactive to the user's questions, but they should also be also proactive, for example, by introducing new topics into the conversation. In other words, they should be able to engage in mixed-initiative, multi-turn, open-domain dialogue. The challenge for the teams is to create a socialbot that can converse coherently and engagingly with human judges for 20 minutes and obtain a rating of 4 or higher out of 5 from the human judges interacting with it. Example 1.16 is an extract from an interaction with the socialbot *Alana*, developed by a team from The Interaction Lab at Heriot-Watt University, Edinburgh, UK that achieved third place in the 2017 and 2018 challenges [Cercas Curry et al., 2018]. Alana has since been launched as a spin-out company in 2020[46]:

Example 1.16
Turn 5

ALANA: Welcome back then `username`. Since I remember that you like movies, I am looking forward to the new movie Jurassic world that was just released. What movie are you looking forward to watching?
USER: Vanilla Sky.

Turn 6

ALANA: Oh I heard something interesting about Vanilla Sky. In the cell, Carl Jung's book, Memories, Dreams, and Reflections can be seen on the table between David and McCabe. The book …
USER: Interesting.

[45] https://developer.amazon.com/alexaprize/challenges/past-challenges/challenge3
[46] https://alanaai.com/

Turn 7

ALANA: I just think the world is full of strange and unexpected things like that. So, talking about movies, what famous actor or actress would you like to meet? I would love to meet **Will Smith**. **He**'s just so funny!
USER: Who is **he**?

Turn 8

ALANA: I guess you know that **Will Smith** is an American actor, film producer and rapper…So `username`, on the topic of movies, what's a movie that you like? …
USER: I want to talk about music.

Turn 9

ALANA: So, talking about music. Who is a singer that you like?

Note the use of anaphoric reference across turns 7 and 8. Alana mentions *Will Smith* and uses *he* anaphorically within the same turn. The user continues the reference and then in turn 8 Alana uses the full reference *Will Smith*, which is a way of checking that they are both referring to the same person. Note also that Alana is trying in turn 8 to find out which movies the user likes, but is able to switch the topic to music on the user's request. In other turns Alana retrieves content from various knowledge sources, such as news, Wikipedia, and ontologies. Chapter 2 provides more detail on the technologies used by Alana and other socialbots in the Alexa prize, while Chapter 4 reviews the methods of evaluation that have been used to select the winners in the Alexa Prize and in other challenges and competitions.

Systems Using Neural Dialogue Technologies

Within the past few years neural dialogue technologies have been used to develop multi-turn open-domain dialogue systems. Example 1.17 shows a dialogue with Meena, an open-domain, end-to-end neural dialogue system developed by the Google Research Brain Team [Adiwardana et al., 2020].

Example 1.17

HUMAN: Hi!
MEENA: Hello! How are you?
HUMAN: I'm doing okay. How about you?
MEENA: I'm doing pretty good, thanks for asking!
HUMAN: What's your name?
MEENA: It's Kimberly! What's your name?
HUMAN: That's a nice name. My name is Paul.

MEENA: Nice to meet you Paul!
HUMAN: Nice to meet you as well Kimberly.
MEENA: So, what do you like to do?
HUMAN: I like going to the movies.
MEENA: Cool! What's your favorite movie?
HUMAN: Pulp Fiction.
MEENA: Nice! I love Reservoir Dogs.
HUMAN: Yes, that movie is good as well.
MEENA: Who's your favorite character in Reservoir Dogs?

Not all of the conversations with Meena were successful, however, and about one third of them degenerated into cross-turn repetitions in which turns are repeated, as shown in Example 1.18:

Example 1.18

HUMAN: Have you even been to India?
MEENA: No, but I'd love to go.
HUMAN: How about Japan?
MEENA: I'd love to go to Japan too.
HUMAN: How about Paris?
MEENA: I'd love to go to Paris too.
HUMAN: Are you lying to me?
MEENA: I'd love to go to Japan too.
HUMAN: i think you're lying to me, that makes me sad.
MEENA: I'd love to go to Japan, too.
HUMAN: You're blocking me out! You have to talk to me.
MEENA: I'd love to go to Japan.
HUMAN: Where else?
MEENA: I'd love to go to Japan, too!

The technologies behind Meena and other advanced neural systems will be reviewed in Chapter 5, Section 5.6.

1.5 DESIGNING AND DEVELOPING DIALOGUE SYSTEMS

There are three main approaches to the design and development of dialogue systems: *rule-based*, *statistical data-driven*, and *end-to-end neural*. In rule-based systems conversation flow and other aspects of the interface are handcrafted using best practice guidelines that have been developed over the past decades by voice user interface designers [Pearl, 2016], [Batish, 2018]. These include guidelines on elements of conversations, such as:

- how to design effective prompts;
- how to sound natural;
- how to act in a cooperative manner;
- how to offer help at any time;
- how to prevent errors; and
- how to recover from errors when they occur.

There are also higher-level guidelines, for example:

- how to promote engagement and retention;
- how to make the customer experience more personal and more pleasant; and
- the use of personas and branding.

Some of these guidelines address linguistic aspects of conversational interaction, such as maintaining the context in multi-turn conversations, asking follow-up questions, maintaining and changing topics, and error recovery. Others are more concerned with social competence, such as promoting engagement, displaying personality, and expressing and interpreting emotion. Finally, there are psychological aspects such as being able to recognize the beliefs and intentions of the other conversational participant, i.e., what is known as *theory of mind*. All of these aspects are important for a conversational agent to be effective as well as engaging for the user.

In the second and third approaches, dialogue strategies are learned from data. Statistical data-driven dialogue systems emerged in the late 1990s and end-to-end neural dialogue systems using deep learning began to appear around 2014. Rule-based systems are reviewed in Chapter 2, statistical data-driven systems in Chapter 3, and end-to-end neural dialogue systems in Chapter 5. Chapter 6 discusses recent developments in hybrid systems that combine rule-based with statistical and/or neural approaches.

SUMMARY

This chapter has introduced dialogue systems, looking first at what motivates developers to develop systems that can engage in conversations with human users and then reviewing the history of dialogue systems. Five different traditions were identified: text-based and spoken dialogue systems that were developed in academic and industrial research laboratories; voice user interfaces that were developed by companies and deployed in commercial environments; chatbots that aimed to simulate human conversation; embodied conversational agents that focused on multi-modal aspects of conversational interaction; and social robots and situated agents. Following this present-day dialogue systems were reviewed, looking in particular at the types of conversational interactions that can be supported on different platforms, and the situations and purposes for which the systems can be deployed. The final section identified three different approaches to the design and development of dialogue systems: a rule-based approach involving handcrafting, a statistical data-driven approach using machine learning, and an end-to-end neural approach using deep neural networks. The next chapter provides an overview of rule-based dialogue systems.

CHAPTER 2

Rule-Based Dialogue Systems: Architecture, Methods, and Tools

Until around 2000 dialogue systems developed in academic and industrial research laboratories were based on rules that determined the system's behavior. Consider Example 2.1 in which the system has to choose between three different possible responses to the user's utterance:

Example 2.1

U1: I want to book a flight to Boston.

S1.1: Sorry, please repeat that. (System cannot interpret U1).

S1.2: Did you say Boston? (System asks for confirmation).

S1.3: Ok, a flight to Boston. What date? (System confirms implicitly and asks for the value of the next slot).

In this example the system's choice of its next action could be determined by how confident it was in its interpretation of the user's utterance, based, for example, on the confidence score returned by the speech recognition component. In a rule-based system this decision would be anticipated by the system designer and included as a pre-scripted rule (see further discussion in Section 2.1.3). In an alternative approach, to be discussed in Chapter 3, decisions such as these are learned from data using technologies such as Reinforcement Learning (RL) (see Chapter 3, Section 3.3).

This chapter reviews the rule-based approach. The chapter is structured as follows. Section 2.1 presents a typical dialogue systems architecture and describes the workings of the different components of the architecture. Section 2.2 describes the development lifecycle for handcrafted dialogue systems, outlining the various stages in the lifecycle. Section 2.3 reviews some tools that have become available for developing dialogue systems while Section 2.3.3 shows how these tools can be used to implement the different types of dialogue introduced in Chapter 1. Finally, Section 2.4 reviews rule-based techniques that were used in some of the systems that have competed for the Alexa Prize in recent years.