For: J. Mildorf & B. Thomas (Eds.) Dialogue Across Media, Amsterdam: John Benjamins.

DRAFT

Dialogue with Computers

Paul Piwek The Open University, UK

Abstract

With the advent of digital personal assistants for mobile devices, systems that are marketed as engaging in (spoken) dialogue have reached a wider public than ever before. For a student of dialogue, this raises the question to what extent such systems are genuine dialogue partners. In order to address this question, we propose to use the concept of a dialogue game as an analytical tool. Thus, we reframe the question as asking for the dialogue games that such systems play. Our analysis, as applied to a number of landmark systems and illustrated with dialogue extracts, leads to a fine-grained classification of such systems. Drawing on this analysis, we propose that the uptake of future generations of more powerful dialogue systems will depend on whether they are *self-validating.* A self-validating dialogue system can not only talk and do things, but also discuss the *why* of what it says and does, and learn from such discussions.

Introduction

dialogue | dialog *noun*

1 a. A literary work in the form of a conversation between two or more persons, in which opposing or contrasting views are imputed to the participants
1 b. *Music*. A composition for two or more alternating voices.
2 a. A conversation carried on between two or more people; a verbal exchange, a discussion.
2 b. As a mass noun: conversation carried on between two or more people; discussion, verbal interchange. Now somewhat *rare*.
2 c. Discussion between representatives of different countries or groups, esp. with a view to resolving conflict or solving a problem; an instance of this.
3. Conversation between two or more characters in a literary work; the words spoken by the actors in a play, film, etc. Also: the style or character of the spoken elements of a work.
4. Computing.
4 a. The exchange of data between computers on a network; an instance of this.
4 b. Chiefly in form dialog. = *dialogue box* n. at Compounds 2.

(OED Online, June 2015)

According to the Oxford English Dictionary, the word ‘dialogue’ has eight different senses (as listed above). Each of these senses identifies certain parties to a dialogue. These include persons, alternating (musical) voices, people, representatives of countries or groups, characters and even computers. Out of the eight senses, five (1a, 2a, 2b, 2c and 3) focus on dialogue as conversation.[1] According to three of these senses, dialogue is, by definition, done by people (2a, 2b and 2c). The possibility of a machine as a dialogue partner is not countenanced. In contrast, at least one of the entries that concerns literary works (i.e. 3) is non-committal, speaking of ‘characters’. And indeed, most of us have encountered machines capable of dialogue in fiction or films. What emerges from this dictionary entry for dialogue is a firm distinction between dialogue in fiction and dialogues in real life: the former permit non-human participants, whereas the latter exclude them.

However, over the last decade this neat demarcation has been breached; machines that engage in dialogue seem to have infiltrated the real world. Personal assistants for mobile phones, think Apple’s Siri, Google Now and Microsoft’s Cortana, are available to anyone in possession of a reasonably modern mobile device. These assistants can help, for example, with booking an appointment or locating a place of interest. In future, their capabilities are likely to extend to many more everyday activities.

With talking machines having made the leap from fiction into reality, it is timely to take stock. Is it still warranted to exclude machines from dictionary definition of dialogue? That question leads to other questions such as: What is the current generation of talking machines capable of? In what sense can they be said to engage in dialogue? How big is the gulf, if any, between them and the talking machines we know from film and fiction? In this paper we approach these questions by examining the ideas and technologies that sit behind the current generation of talking machines or, henceforth, dialogue systems. The discussion is aimed at an interested, but non-technical audience and includes a generous serving of transcripts of ‘conversations’ with such dialogue systems.

We will see that there is a large variety of dialogue systems. For instance, the medium of communication ranges from typed text and spoken language to virtual computer-animated agents and robots. The purpose of the dialogue differs significantly from system to system. We will encounter both dialogue systems that generate entire dialogue scripts (emulating the work

[1] As opposed to dialogue as a musical composition with alternating voices (1b) or and data exchange with computers (4a and 4b)

of the author of a book or play) and systems that engage in one-to-one dialogue. The emphasis will however be on the latter.

To introduce some structure into the analysis, we deploy the notion of a dialogue game. This notion has been hugely influential in linguistics, philosophy, computer science and abutting areas in which dialogue is studied. Each dialogue system is described in terms of the dialogue game that it can play. The description of dialogue systems at this level of abstraction will facilitate comparisons between these systems.

In the remainder of this paper, we proceed as follows. Section 1 looks at influential representations of talking machines in film. The section highlights a common template behind these representations. Section 2 introduces the notion of a dialogue game. This section provides us with the tools to describe the dialogue systems that are introduced in the next two sections. Sections 3 and 4 deal with reactive and agenda-driven dialogue systems, respectively. Reactive systems have no explicit representation of purpose; they take the words of their interlocutor, reorganise these words and fire them back. In contrast, what an agenda-driven system says next depends not only on what the user has said but also on the system's goals. In some sense, such systems have a mind of their own. Section 5, looks at recent research on dialogue systems. Finally, in Section 6 we take stock, returning to the questions that were raised in this introduction.

1. Talking machines in fiction: HAL, Ava and Baymax

HAL is possibly the most influential instance of a fictional talking machine. It features in '2001: A Space Odyssey', which is both a novel and a film created alongside each other in a collaboration between the science fiction author Arthur C. Clarke and director Stanley Kubrick. HAL is the central computer of a spaceship and speaks with a rather artificial sounding voice. Its presence is marked by an ominous looking camera lens, which exudes a constant red glow. HAL can be polite, express fears, and, as the ship's crew find out, its own goals, when Dave is locked out of the spaceship:

(1) Dave: Open the pod bay door please, HAL. Open the pod bay door please, HAL. Hello, HAL. Do you read me? Do you read me, HAL? Hello, HAL. Do you read me?
HAL: Affirmative Dave, I read you.
Dave: Open the pod bay doors, HAL.

HAL:	I'm sorry, Dave, I'm afraid I can't do that.
Dave:	What's the problem?
HAL:	I think you know what the problem is just as well as I do.
Dave:	I don't know what you're talking about.
HAL:	I know that you and Frank were planning to disconnect me, and I'm afraid that's something I cannot allow to happen.

A more recent instance of a talking machine that breaks free from its human masters is Ava. In Alex Garland's 2015 film 'EX_MACHINA', Ava is a humanoid robot portrayed by the actress Alicia Vikander. Though HAL and Ava appear to have little in common on the outside, Ava, not unlike HAL, discovers that its goals diverge from that of the humans, in this case its creator Nathan, and solicits the help of Caleb, a programmer working for Nathan's company. The following dialogue snippet is representative of their interactions:

(2)	Ava:	Nathan isn't your friend. You're wrong.
	Caleb:	Wrong about what?
	Ava:	Everything.

As we shall, even Baymax, the cuddly inflatable robot protagonist in the Disney film 'Big Hero 6', seems to fit, up to a point, the template that was established by HAL. Baymax is a personal healthcare companion. It tends to the needs of its human users. However, despite or perhaps because of its benevolent goals, Baymax can end up acting in ways that are in direct conflict with explicitly stated requests. Here is an interaction between Baymax and Hiro, a teenage boy:

(3)	Baymax	I heard a sound of distress. What seems to be the trouble?
	Hiro	Oh, I just stubbed my toe a little. I'm fine.
	Baymax	On a scale of 1 to 10 how would you rate your pain (Baymax displays a scale from ☹ to ☺).
	Hiro	A zero. I'm okay really. Thanks, you can shrink now.
	Baymax	Does it hurt when I touch it?

Hiro	Naah. Okay. No touching. (Moves backwards to evade Baymax who is trying to touch his toe) I'm fine (loses balance and falls backwards).
Baymax	You have fallen.
Hiro	You think.

The theme that emerges is one of conflict: conflict between the goals of the human interlocutor and the machine. In the case of HAL and Ava, the machine pursues its own goals whilst realising that these are at odds with those of its human interlocutors. In contrast, Baymax (at least in the dialogue fragment above) seems to stubbornly follow the script it has been given. It ignores any input that falls outside of this script. Part of the comical effect is derived from Baymax appearing to be genuinely oblivious to this fact.

So much for the fictional machines. After an introduction to the concept of a dialogue game in the next section, the subsequent two sections deal with actual dialogue systems. In the concluding parts of this paper, we return to the question how these systems compare to HAL, Ava and Baymax.

2. Dialogue Games

> Imagine this language: --
>
> 1). Its function is the communication between a builder A and his man B. B has to reach A building stones. There are cubes, bricks, slabs, beams and columns. The language consists of the words "cube", "brick", "slab", "column". A calls out one of these words upon which B brings a stone of a certain shape. Let us imagine a society in which this is the only system of language. The child learns this language from the grown-ups by being trained to its use. (Wittgenstein, 1958: 77)

In his later work, the philosopher Ludwig Wittgenstein (1889 – 1951) describes numerous hypothetical practices, such as the one given above. For these he coined the term 'language game'. His aim was to show that the meaning of an utterance can be understood in terms of what one can do with the utterance (in this example, coordinate the actions between A and B). He did this partly to supplant his earlier picture theory of meaning with another metaphor: the use of sentences in conversation as moves in a game. He saw this change of metaphor as crucial for dispelling certain philosophical quandaries that the picture theory leads to (questions such as: ‘What does the number ‘four’ represent?’, ‘Do numbers exist?’, and so on. – arguably, these questions evaporate when the use of numbers is thought of in terms of practical language games).

Wittgenstein’s idea of utterances as moves in a language game became hugely influential. Researchers in several disciplines developed it further. For example, the Wittgenstein scholar Erik Stenius proposed precise formulations of several dialogue game rules (Stenius, 1967). Another pioneer was the Australian philosopher and computer scientist C.L. Hamblin. He provided a firm basis for formal study of dialogue games or, in his terminology, dialectical systems (Hamblin, 1970). These strands of research prepared the ground for the use of the notion of a dialogue game in research on dialogue systems. Possibly the earliest example of such research is the work by Bunt and Van Katwijk at the Institute for Perception Research (a collaboration between Eindhoven University and Philips Research in the Netherlands). They draw on the analogy between dialogue and parlour games such as chess:

> What does it mean to view something as a game? A game is an activity in which the participants take turns in performing certain actions, chosen from the set of ‘legitimate moves’, in order to arrive at a preferred situation (‘favourable position’). Comparing this characterisation of a game with the characterisation of informative dialogues […] we can indeed view [dialogue] as a game, sequences of dialogue acts corresponding to moves, and the position that the players want to reach being a desired state of knowledge (…) think of a ‘position’ as an independent concept, as ‘configuration of pieces’, as is for instance common in chess. (Bunt and Van Katwijk, 1979: 266-268)

In the same spirit, we define dialogue games as consisting of two key components: [2]

---

[2] The terminology in this paper is rooted in the tradition that was started by Bunt & Van Katwijk (1979) at the Institute of Perception research (IPO). Our definition draws on subsequent work in this tradition at IPO, in particular: Beun (2001), Ahn et al. (1995) and Piwek (1998). Related approaches to dialogue (in terms of a rule-governed activity) have been developed by, for example, Ginzburg (2012) and researchers involved with the influential TRINDI project and its successors, see, e.g., Larsson and Traum (2003) and Bos et al. (2003).

(Definition) Dialogue Game

A dialogue game consists of two principal components:

A dialogue store, for keeping track of the current position.

Dialogue rules which specify, for any given point in a dialogue, which dialogue acts are permitted at that point in the dialogue and how the store changes as a result of those actions. They are divided into two types of rules:

a) *update rules*, which specify how the dialogue store evolves in the course of a dialogue.
b) *generation rules*, which specify which dialogue acts are legitimate given a specific position (as recorded in the dialogue store).

Additionally, each dialogue participant needs a *dialogue strategy*. Given a set of available legitimate dialogue acts for a position, the strategy picks the act which will actually be played, as illustrated in Figure 1. The analogy with the game of chess is helpful here. Think of the rules of chess that specify the possible moves of the pieces as the generation rules. Such rules determine the legitimate moves one can make at each point in a chess game. To play the game, every time it is one's turn, one needs to select an actual move from the set of legitimate moves.

We will see that in many dialogue systems, generation rules and a strategy are conflated: such systems have a single set of rules that determines the next dialogue act, without distinguishing between the legitimate acts that one is allowed to play according to the game and the actual act that is played.

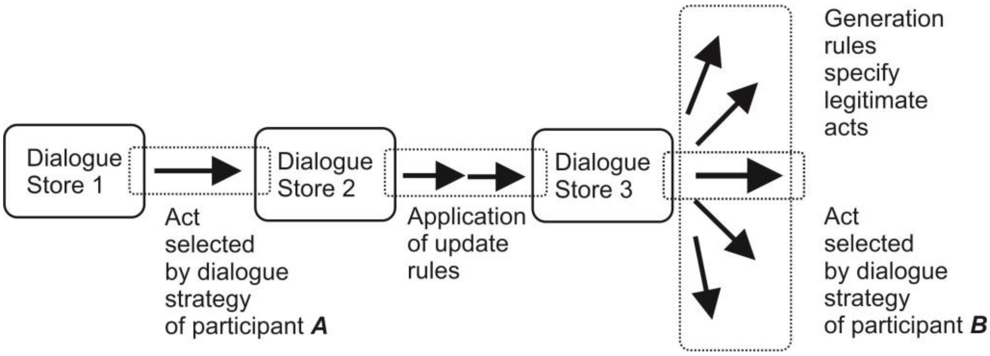

Figure 1: Playing a dialogue game involves participants, here *A* and *B*, taking turns. We begin with Dialogue Store 1, the initial dialogue store. Participant *B* performs a dialogue act. This results in Dialogue Store 2. Application of the update rules yields Dialogue Store 3. Given Dialogue Store 3, the generation rules determine which legitimate acts are available to *B*. From these legitimate acts, *B*'s dialogue strategy selects an act to perform. And so on.

We have been purposely agnostic about the precise content of the dialogue store and details of the update and generation rules. We shall see that these vary with the dialogue game. A dialogue system can be thought of as playing a particular dialogue game that is fixed by the specific dialogue store and rules that are involved. Looking at dialogue systems in this way allows us to make explicit both similarities and differences between such systems.

Before we look at some concrete dialogue systems and the corresponding games, we briefly address a prima facie objection to this view of dialogue. It may appear rather restrictive to think of a dialogue as governed by a set of underlying rules that can be mechanically applied. How can this be squared with the inherent flexibility and creativity of genuine dialogue? A variant of this objection is dealt with by Alan Turing, the father of computing, in his seminal 'Computing machinery and intelligence'. He refers to it as 'The argument from informality of behaviour' (Turing, 1950:58). Turing points out that even though human behaviour may not appear to follow a clear set of rules, it doesn't follow that such rules cannot exist. He points out that a computer programmed with a couple of simple rules can display apparently informal, unpredictable behaviour:

> I have a set up on the Manchester computer a small programme using only 1000 units of storage, whereby the machine supplied with one sixteen-figure number replies with another within two seconds. I

would defy anyone to learn from these replies sufficient about the programme to be able to predict any replies to untried values. (Turing, 1950: 59)

Of course, there are further simplifying assumptions behind the view of dialogue as a game. It is beyond the scope of this paper to delve into those, but see for example Piwek (2011).

3. Reactive dialogue systems

One of the earliest and most well-known dialogue systems is ELIZA. It was developed in the 1960s at MIT by Joseph Weizenbaum. Interaction with ELIZA is via the keyboard (through a chat interface). ELIZA takes on the role of a psychotherapist and does this quite convincingly. Weizenbaum (1966) reports that it was difficult to persuade some people that they were interacting with a machine.

What follows is a, according to Weizenbaum, typical conversation with ELIZA:

(4)	1	Human	Men are all alike.
	2	ELIZA	in what way
	3	Human	They are always bugging us about something or other.
	4	ELIZA	can you think of a specific example
	5	Human	Well, my boyfriend made me come here.
	6	ELIZA	your boyfriend made you come here
	7	Human	He says I'm depressed much of the time.
	8	ELIZA	I am sorry to hear you are depressed
	9	Human	It's true. I am unhappy.
	10	ELIZA	do you think coming here will help you not be unhappy
	11	Human	I need some help, that much seems certain.
	12	Eliza	what would it mean to you if you got some help
	13	Human	(…)

(Extract from Weizenbaum, 1966: 36-37)

Weizenbaum argues that:

> The whole issue of the credibility (to humans) of machine output demands investigation. Important decisions increasingly tend to be made in response to computer output. (…) ELIZA shows, if nothing else, how easy it is to create and maintain the illusion of understanding, hence perhaps of judgement deserving credibility. A certain danger lurks there.' (Weizenbaum, 1966: 42-43).[3]

At the same time, Weizenbaum suggests that once the lid is lifted on how the machine achieves this feat, 'the magic crumbles away'. So let us briefly look at what makes ELIZA tick. To do so, we specify the dialogue game that it plays in terms of its dialogue store and rules.

ELIZA's game

In the case of ELIZA, the dialogue store records the most recent input from the human user. For instance, consider dialogue act (4.7) by the human interlocutor, repeated here for the reader's convenience:

(5) He says I'm depressed much of the time.

This text is recorded on the dialogue store. In a first step towards converting this user input into a response, an update rule transforms certain pronoun-verb combinations. In particular, there is an update rule that replaces occurrences of 'I'm' with 'you are'. Thus the content of the dialogue store is changed into the following sentence:

(6) He says you are depressed much of the time.

Next, the generation rules are applied. Generation rules specify, given the content to the dialogue store, which dialogue acts are legitimate responses. ELIZA's generation rules consist of two parts.

Firstly, there is a decomposition template which may or may not match with the input text. For example, the following template matches with the (amended) input text:

(7) you are sad / unhappy / depressed / sick ...

[3] More recently, two researchers at Stanford University carried out a series of experiments to see how people treat machines. This led to an influential book entitled 'The Media Equation' (Reeves & Nass, 1996). In their book, Reeves and Nass argue that people do indeed tend to treat computers as if they were real people.

Here, '…' indicates an indefinite sequence of words.

Secondly, the generation rule consists of one or more reassembly patterns, such as:

(8) I'm sorry to hear you are sad / unhappy / depressed / sick

The generation rule consisting of the aforementioned decomposition template and this reassembly pattern turns

(9) He says you are depressed much of the time.

into ELIZA'S response:

(10) I'm sorry to hear you are depressed.

Note that the generation rule ignores some of the input ('much of the time') and prefixes the response with the phrase 'I'm sorry to hear'.

ELIZA has a stock of generation rules. Each of these specifies a potential legitimate act. Its dialogue strategy is to try one rule at a time until a match has been found.[4] Recall that a dialogue strategy determines how a specific dialogue act is selected from the set of legitimate ones (as specified by the generation rules). When a match between an input and generation rule has been found, a text is put together according to the reassembly pattern of the rule. Then the text is presented to the user (via the chat interface) and the store is wiped clean. The latter is effected by an update rule.

If the input doesn't match any of the normal generation rules, ELIZA will do of two things. Either it applies a rule of last resort. This rule matches regardless of the input text. The reassembly pattern is bit of canned text such as 'I see' or 'that's interesting'. Alternatively, it can draw on a phrase which it stored earlier on. For this purpose, there is a section of the dialogue store, labelled 'memory', in addition to the 'input' section we've made use of so far. Items can be added to memory by a special update rule. Whenever the system encounters 'your ...' in the input text, it puts in the 'memory' the

[4] We are skimming over some technical details. In particular, ELIZA's generation rules are actually ranked. Thus, the strategy is more sophisticated in that the system always first tries the highest ranked rules. There are other details of the ELIZA implementation (mostly due to the fact that in the 1960s computer memory was much more limited than today, necessitating various memory saving techniques). We've ignored those in our discussion.

phrase ‘let’s discuss further why your ...’ or ‘earlier you said your ...’ and proceeds to respond in the usual way. If it later on encounters a situation where no rule matches, it can retrieve a phrase from memory and produce it. For instance, if earlier on in the dialogue the human interlocutor said ‘my boyfriend made me come here’, when the system gets stuck, it can be thrown into the conversation ‘let’s discuss further why your boyfriend made you come here’.

Weizenbaum points out that he carefully selected the psychiatric interview in order to keep the number of generation rules for ELIZA under control. A psychiatrist gets away with saying ‘Tell me about boats’ in response to ‘I went for a long boatride’ because

> one would not assume that he knew nothing about boats, but he had some purpose in is so directing the subsequent conversation. It is important to note that this assumption is made by the speaker. Whether it is realistic or not is an altogether separate question. (Weizenbaum, 1966: 42)

Beyond ELIZA

The ideas that underpin ELIZA live on in the chatbots of today. There have also been efforts to address some of ELIZA’s shortcomings. For example, the way ELIZA matches an input with a response is rather brittle: it requires that the exact words of the decomposition template have been used by the human interlocutor. For instance, Leuski & Traum (2008) relaxed this requirement. Given a database of generation rules, their algorithm responds to a user input by finding a generation rule whose decomposition template is most similar to the user input, no longer requiring an exact match. This means that the system will able to respond in more situations. This of course has to be traded off against the fact that some of the system’s responses may be less relevant or appropriate.

CODA: Automatic harvesting of generation rules

A further area in which progress has been made is the automatic harvesting of generation rules from text. For this purpose, it is best to think of a generation rule as a short dialogue fragment (e.g. a question followed by an answer). A dialogue system utilises such rules by recognising that the user’s input, e.g. a question, matches with the beginning of such a fragment, and responding with the remainder of the fragment. In the CODA project[5] (Piwek & Stoyanchev, 2010), automatically extracting such rules was addressed in three steps. Firstly, a set of monologue-dialogue pairs was constructed in which professionally-authored dialogue was aligned with monologue expressing the same information.

Table 1: Example of monologue-dialogue pairs, with the monologue on the left-hand side and dialogue expressing the same information on the right-hand side. The monologue is annotated with rhetorical relations (Attribution, Contrast) and the dialogue is annotated with dialogue acts (Yes/No Question, Explain, Answer No).

Monologue		**Dialogue** (from Twain 1919: 14 and 1)		
Text	*Rhetorical relation*	*Speaker*	*Text*	*Dialogue act*
One cannot doubt that he felt well.	Attribution	OM	He felt well?	Yes/No Question
		YM	One cannot doubt it.	Explain
The metals are not suddenly deposited in the ores. It is the patient work of countless ages.	Contrast	OM	Are the metals suddenly deposited in the ores?	Yes/No Question
		YM	No --	Answer No
		YM	it is the patient work of countless ages.	Explain

Both the monologue and the dialogue were analysed for patterns, using rhetorical structure theory and dialogue act annotation. An example is provided in Table 1. Secondly, from this resource, rules were automatically constructed that mapped patterns in monologue to dialogue patterns. [6] Finally, these rules could then be applied to new monologue to extract generation rules, e.g. in the shape of question-answer pairs. This approach was used to automatically create generation rules for a virtual instructor that explains consent forms for clinical trials (Kuyten et al., 2012).

In fact, CODA was initially conceived for a different purpose. The aforementioned automatic monologue-to-dialogue mapping can also be used to turn an extended piece of monologue into a dialogue. This opens the possibility of automatically creating a short film script from a text. For example, the CODA approach was applied to leaflets from a charity, the Papworth trust, to generate short film scripts that presented the information from the leaflets in a different medium.

[5] CODA is short for 'COherent Dialogue Automatically generated from text'. It was a two-year project funded by the UK EPSRC research council under grant EP/G020981/1.
[6] The resource, annotation tools and rules are available under Creative Commons license via the CODA project homepage at `http://computing.open.ac.uk/coda/`

4. Agenda-driven systems

Reactive systems take the user's input, transform it, and present the result to the user. Such a system translates the user's language into a system response. The transformations for achieving this stay at the surface of the user's input. The dialogue store merely stores the user's input and the successive transformed versions, until a response has been constructed and is ready to be released back to the user.

We use the term agenda-driven systems for dialogue systems that have their own agenda. Such systems are driven not only by what the user says, but also by their own goals and plans. We will distinguish between three types of agenda-driven systems, according to the agenda's flexibility:

1. fixed-task agenda
2. selected-task agenda
3. joint-task agenda

4.1 Fixed-task agenda

Systems based on a fixed-task agenda are aimed at helping the user with one specific pre-defined task. A classical example is the task of obtaining travel information. The dialogue system plays the intermediary between the user and a database. The user has a goal: they want to travel from *A* to *B* at time *t* and need to know available connections. This information can be obtained from the database. However, the language that the database 'speaks' is not the user's, rather it is formal database query language. The dialogue system's goal is to obtain the place of departure and destination and travel time from the user. It then translates this into an appropriate query which it puts to the database. The database supplies an answer which the system translates back into language that the user understands. The following is a transcript of an interaction with the first publicly available system of this kind, the Philips automatic train timetable information system.[7]

(11)	1	System	Good morning, this is the automatic time-table information service. How can I help

[7] The key ideas behind this type of system can be traced back even further to work by Daniel Bobrow and colleagues at Xerox Palo Alto Research Center on the GUS travel agent system (Bobrow et al., 1977).

		you?
2	Caller	Good morning. I need a connection from Hamburg to Munich.
3	System	When would you like to go from Hamburg to Munich?
4	Caller	Next Friday, er, at eight o'clock.
5	System	So you want to travel on Sunday the 25th of September at 8am?
6	Caller	No, on Friday.
7	System	So you want to travel on Friday the 23rd of September at 8am?
8	Caller	Exactly.
9	System	There is the following connection: With Intercity Express 793, departure from Hamburg at 8:02, arrival in Munich at 14:05. Would you like me to repeat the connection?
10	Caller	No, but I need a return trip on Sunday.
11	System	(…)
		(Extract from Aust et al., 1995: 251)

In agenda-driven systems, the direct link between user input and system response is no longer present. The system's utterances are motivated primarily by the underlying task. This is reflected by the kind of dialogue game that fixed-task systems play. In particular, the dialogue store includes the system's agenda. In the case of a travel information system, the agenda consists of the following ordered list of items:

1. ask for the place of departure
2. ask for the destination
3. ask for the time of travel
4. provide the connection

This agenda is private to the system. At the outset of the conversation, the user may not know that the system is going to proactively seek this information.

Apart from this private section of the dialogue store, there is a common section. This section stores the information the interlocutors have, so far, shared with each other. It focuses on information related to the task. The common section of the store has three slots (corresponding with the first three items on the private agenda):

- point of departure: ___
- destination: ___
- travel time: ___

Ignoring the initial exchange of greetings,[8] let's look at (11.2). An update rule scans the user's utterance for possible fillers for these slots. In this case, the words 'to' and 'from' suggest the presence of such fillers: 'from Hamburg' and 'to Munich'. The private section is updated as follows:

- point of departure: Hamburg (unconfirmed)
- destination: Munich (unconfirmed)
- travel time: ___

As shown, the system isn't yet entirely sure whether the user really wants to go from Hamburg to Munich; these slots are, as yet, unconfirmed. It may for instance, be that the system misheard the user.

In this game, the system relies on a generation rule which does two things: it retrieves the next item on the agenda and any unconfirmed slots, and formulates an utterance relating to the next agenda item (provided the next item can be carried out), whilst also confirming any unconfirmed slots. In the dialogue, this is utterance (11.3): the system asks for the time of travel and tries to confirm the place of departure and destination. Note that agenda items 1. and 2 are skipped since the user provided a point of departure and destination even before the system could explicitly ask for these.

The user responds with the time of travel (11.4) and the private section is updated accordingly:

- point of departure: Hamburg (confirmed)
- destination: Munich (confirmed)
- travel time: Sept 25, 8am, Sunday (unconfirmed)

Because the user didn't refer to the place of departure and the destination, these are considered confirmed. The travel time has been entered, but is as

[8] An initial greeting can be dealt by a simple generating rule which stipulates that at the start of a conversation the system produces an utterance along the lines of 'Good ___, this is the automatic time-table information service. How can I help you?'. Depending on the time of day, ___ is replaced with 'morning', 'afternoon' or 'evening'.

yet unconfirmed. On this occasion the system has misheard what the user said. The generation rule is again applied. In this case, there is no agenda item to ask about (since the final agenda item can only be carried out once all slots are known). However, there is an unconfirmed slot, and so the system asks about this slot: 'So you want to travel on Sunday the 25th of September at 8am?' (11.5) In response, the user utters a correction: 'No, on Friday.' (11.6) The system updates the common section accordingly:

- point of departure: Hamburg (unconfirmed)
- destination: Munich (confirmed)
- travel time: Sept 23, 8am, Friday (unconfirmed)

Note that the system has worked out that Friday means Sept 25. The corrected travel time is, however, still unconfirmed. The generation rule is applied once more, and the system utters 'So you want to travel on Friday the 23rd of September at 8am?' (11.7) The user responds with the confirmation 'Exactly' (11.8). Finally, all the slots are confirmed and the system can proceed with the final agenda item: retrieving the train connection from the database and providing the information to the user (11.9).

4.2 Selected-task agenda

In the case of the train timetable information system, the dialogue is structured by the slots that the system needs to fill to accomplish the task. Recent personal assistants, such as Siri, Google Now and Cortana, operate in a similar way. These systems can, however, assist with more than one task. Whenever the user says something, the system first needs to determine which task the user has in mind. For instance, tasks that can be accomplished with the help of Siri include: launching an application, sending messages, accessing restaurant recommendations, adding reminders to the calendar, and setting the alarm and searching on the web. Here is dialogue in which the user engages Siri to place a call:

(12)	1	User	*Activates Siri by pressing a specific button on the phone or saying 'Hey Siri'*
	2	User	Call Joe.
	3	Siri	Just to confirm – you'd like to call Joe Bloggs? [Cancel] [Call]

4	User	*Selects cancel by pushing the button on the touch screen*
5	Siri	Ok. (Based on conversation with Siri on June 20 2015)

Because Siri can assist with many different tasks, on each occasion it needs to establish which task the user is currently talking about. For this it looks for keywords such as ‘Call’, ‘Launch’, ‘Search for’, etc. In this example, it finds a match with ‘Call’. To place the call Siri needs a name. Generally, the tasks that Siri deals with involve only a single slot. The name provided by the user is matched against the user’s address book. If a matching name exists, Siri confirms that it has identified the correct address book entry. If confirmed, it proceeds to place the call. All this, when compared to the train timetable system, results in a relatively brief dialogue.

Apart from the length of conversations, there are two further key differences between Siri and the train timetable system. First, Siri can access various bits of contextual information that the user has stored on their mobile phone - in contrast with the train timetable system which is accessed by calling a landline number. In our example, Siri can only perform its task, because it has permission to consult the user’s address book. Second, Siri interacts with the user by means of a combination of media. In our example, it uses speech but also the touch screen (to obtain the user’s confirmation). In response to another query “Where is the nearest supermarket”, it will bring up a map. In that case, it also draws on the user’s location (as provided by the phone’s GPS system), to work out the nearest supermarket.

When Siri has determined the task, it needs to establish the slot value. E.g., for the ‘Where’ question in the previous paragraph, it uses ‘supermarket’. It works out that this is a type of location, rather than, say, a street name. Sometimes it will however get this wrong. For instance, when asking ‘Where is the sea’, whilst being only 2 miles away from the sea front in the South-east of England, it came back with ‘The only possibility I found is the Seattle-Tacoma International Airport on International Blvd in Sea Tac.’ and a map of this airport. The hedge ‘The only possibility I found’ suggests that Siri had some idea that the answer might not be what I was looking for.

4.3 Joint-task agenda

Both fixed-task and selected-task systems have an agenda, but this agenda is relatively rigid. It involves a single task, whether fixed in advance or selected by the user. In this section we consider systems that operate with a joint-task agenda. Such systems are distinct from fixed and selected-task systems in two important ways. Firstly, the task at hand is established as part of the conversation, rather than being determined or guessed at by one of the parties. Secondly, the interlocutors collaborate on achieving this task. In particular, they jointly plan and then carry out actions that lead to its achievement.

Joint-task systems are still confined to the research laboratory. In fact, their theoretical underpinnings go back to the 1970s. The pioneering work in this area is exemplified by a system that was developed as an exercise in theoretical psychology. The author of this system emphasized that

> We are not yet able to construct formally precise theories of advanced cognitive processes such as language understanding (…) The present model will have served its purpose if it (a) highlights some problems in the organisation of dialogue which the reader may not have explicitly noticed, (b) explores a clear set of ideas for solving these problems (the most important idea in this case being the 'conversational procedure'), and (c) exposes the limitations of these ideas and therefore helps someone to construct a better theory. (Power, 1979: 109)

and this still remains largely true today. Power's (1979) system involves two computer programs, 'John' and 'Mary' that can engage in a conversation with each other. They inhabit a simple virtual world consisting of a room with a door through which one can enter or leave the room (see Figure 3). The door is fitted with a bolt that is either up or down.

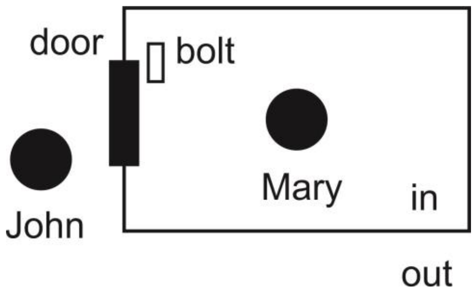

Figure 2: A visualisation of the virtual world in which the John and Mary programs operate.

John and Mary have different capabilities. Both can move around and slide the bolt, but only Mary can push the door. John can see things, whereas Mary is blind. As a result, John and Mary have to cooperate to achieve certain goals. In particular, we shall look at a conversation that took place when John found himself outside the room, with the door bolted:

(13)	1	John	I want to suggest a goal.
	2	Mary	Go ahead.
	3	John	Will you help me get in?
	4	Mary	By all means.
	5	John	Shall we make a plan.
	6	Mary	May I ask you something
	7	John	Go ahead.
	8	Mary	Are you in?
	9	John	No.
	10	Mary	Shall we make a plan?
	11	John	Okay.
	12	Mary	Is the door open?
	13	John	No
	14	Mary	I suggest that we get the door open and then you move.
	15	John	I want to explain something.
	16	Mary	Go ahead.
	17	John	If one moves, nothing happens.
	18	Mary	I disagree. If one moves when the door is open one changes position.
	19	John	I see.
	20	Mary	I suggest that we get the door open and then you move.
	21	John	All right.
	22	Mary	Shall we make a plan?
	23	John	Okay.
	24	Mary	I suggest that I push the door.
			(…)
			(Power, 1979: 116-118)[9]

[9] In the actual conversation between John and Mary, before each new section of dialogue (e.g. suggesting a goal, making a plan), the initiating partner calls the other by name and the addressee than acknowledges, as in:

John	Mary
Mary	Yes.

For the sake of brevity, we've omitted these dialogue acts.

This conversation emerged as a result of the two programs, John and Mary, addressing the situation they found themselves in. Both had a desire for John to join Mary in the room, but needed to cooperate to achieve that end. Each can be thought of as playing a dialogue game. The conversational score has a number of novel components. In particular, rather than a simple agenda, it now contains a plan, which is constructed in the course of the conversation (see Figure 4).

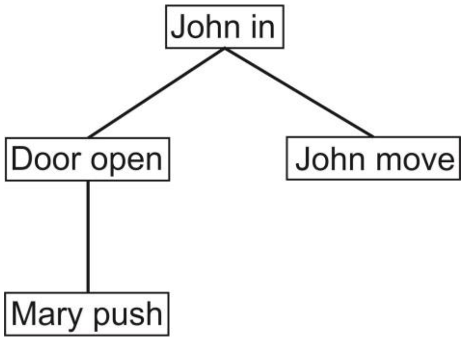

Figure 3: A visualisation of the plan that John and Mary construct.

This plan is no longer a simple sequence of actions, but shows that some goals require that one or more other goals are achieved. For instance, for John to be inside the room, the door needs to be opened and then he can move. After having proposed the goal (13.1), John suggests to make a plan for achieving the goal (13.5). Mary is not able see, so she first checks with John whether the goal has already been achieved (13.6-9). They then proceed to flesh out the plan. There is a further complication along the way. Each interlocutor has beliefs about the world, which populate the belief section of their dialogue store. In particular, John originally has the erroneous belief that if he moves, nothing happens. He is set straight on this one in (13.15-19). As the conversation proceeds the plan and beliefs are updated and new utterances are produced.

The generation of new utterances is driven by the planning process. Planning systems have been pivotal in Artificial Intelligence research from its inception in the 1950s. What makes the John and Mary programs different is that they coordinate and achieve their goals through communication. When one of them realises that they can’t achieve a goal on their own (e.g. for John to get inside the room), they initiate a conversation which allows them to jointly achieve the goal. This is done through conversational procedures (such as ‘ChooseGoal’, ‘AgreePlan’ and ‘Ask’). John and Mary maintain a control stack that records which procedures are currently active. For the dialogue to be successful, they need to carefully coordinate their control stacks, which they achieve by explicitly announcing conversational procedures (e.g. 13.1, 13.5 and 13.10).

John and Mary's dialogue is more complex than any of the dialogues so far, requiring a more complex dialogue store which includes a plan, beliefs and a control stack. The dynamics of the store can be modelled in terms of update and generation rules.

5. Research frontiers

We have outlined some of the core ideas behind dialogue systems and seen that recent systems make extensive use of contextual information and multimodal interaction. In this section, we want to briefly highlight two developments that are still confined to the research lab, as a selective illustration of current research.

So far, the systems we have looked at take a complete utterance, process it in some way, and then respond. In human-human dialogue, interlocutors don't wait until the other has finished before they start processing what was said. They may even interrupt each other mid utterance:

(14)	A:	They X-rayed me, and took a urine sample, took a blood sample. Er, the doctor
	B:	Chorlton?
	A:	Chorlton, mhmm, he examined me, erm, he, he said now they were on about a slide huncleari on my heart. [BNC: KPY1005-1008] (Cited from Gregoromichelaki et al., 2011: 212)

For a system to accomplish this, it needs to analyse input in a piecemeal, incremental word-by-word fashion. A number of theories and computer models, such as Dynamic Syntax (Kempson et al., 2001; Gregoromichelaki et al., 2011), are being developed to do exactly this.

A second development that we would like to highlight is the study of non-cooperative dialogue. All systems we described engage in cooperative dialogues. In recent years, the first models of non-cooperative dialogue have been constructed (Plüss et al, 2011, Plüss, 2013). In this work, a dialogue game-based approach is used to deal with non-cooperation in political interviews. The idea is that certain utterances put an obligation on the addressee to respond in a particular way (e.g. a question should be followed by an answer) – we can think of these obligations as being imposed by generation rules. In other words, the set of legitimate moves is constrained by previous utterances. The key ingredient of the new model is the way the dialogue strategy relates to these obligations. So far, we have assumed that the dialogue strategy selects an utterance from this set of legitimate moves – the ones that are compatible with the speaker's obligations. Plüss considers a more liberal strategy where an interlocutor may chose a move that is not legitimate, but serves a private goal. E.g., in a political interview, a politician may choose to evade a question and raise, instead, an issue that they want to discuss. Key to such a model is a clear distinction between the dialogue rules (the game) and the strategy – as we have seen, many existing system do not distinguish between the two.

6. Concluding remarks

A dialogue system can be seen as playing a dialogue game. We looked at two distinct kinds of dialogue game that current systems engage in: reactive and agenda-driven. Reactive systems take the user's utterance, reorganise, prune and embellish it, and then play it back to the user. Any intelligence is in the eye of the beholder. Agenda-driven systems maintain a representation of things to do, the agenda. This agenda drives what they say next. Agenda-driven come in three flavours: fixed-tasked, selected-task and joint-task. Both fixed-task and selected-task systems orient the conversation around a single well-defined task. This task may be fixed, even before the conversation has started, or be selected by the user in their initial move. Selected-task systems have become a standard accessory of mobile phones, witness Siri (for the iPhone), Google Now (for Android phones) and Cortana (for Windows devices).

In contrast with fictional dialogue systems, such as Ava and HAL, the goals these dialogue systems pursue are simple and the idea that they could develop malicious or human-unfriendly goals seems far-fetched. In particular, Ava and HAL's intelligent assessment of the situation and deliberate violation of the wishes of their human masters is at odds with how current dialogue systems operate.

If anything, it seems more likely that, at least in the foreseeable future, the danger lies not so much with systems that deliberately come into conflict with their human masters, but rather with systems that, inevitably, sometimes make mistakes. Such mistakes arise when the system misunderstands what their human user wants. If such a system then nevertheless proceeds with carrying out what it thinks the user wants, there may be undesirable consequences. For this reason, current systems have safeguards. In particular, such systems ask the user for explicit confirmation before carrying out a task (e.g. the user is given the option to cancel or proceed with making a call).

As the capability of dialogue systems evolves and joint-action based systems move from the lab into the real-world, such safeguards will be more and more difficult to sustain. It will be impractical to rely on a system for accomplishing a complex task (which may involve many decisions and inferences) and expect it to ask for confirmation for each and every individual action that is required to accomplish the task. Just like we wouldn't want a human personal assistant to ask for permission for every action they undertake (which may range from putting a staple through a pile of paper to releasing a few thousand pounds from a budget to pay for some item). If such systems are to be relied upon, they will need to be able to make common sense judgements about when to ask for confirmation. Even if we can equip machines with the knowledge to make such judgements, they will get it wrong on occasion - just as we humans are not infallible when it comes to applying common sense.

The question therefore arises under what circumstances we would accept such systems. And this question is tied to practical issues such as: What would be their legal status? Who would be responsible if things go wrong? I would like to surmise that dialogue may be part of the answer: if we are to ever accept such systems, they'd need to able to discuss their reasons and goals – it's unlikely that we would trust systems that we don't understand. Currently, research on dialogue systems still has some way to go before we have systems that are able to discuss why they said or did something and learn from the discussion – in short, be self-validating dialogue systems. Arguably, such self-validating dialogue systems would also be more deserving of the claim that they engage in genuine dialogue.

References

Ahn, R., R.J. Beun, T. Borghuis, H. Bunt and C. van Overveld. 1995. "The DenK-architecture: a fundamental approach to user-interfaces." *Artificial Intelligence Review* 8(3): 431-445.

Aust, H., M. Oerder, F. Seide and V. Steinbiss. 1995. "The Philips Automatic Train Timetable Information System." *Speech Communication* 17: 249–262.

Beun, R.J. 2001. "On the Generation of Coherent Dialogue: A Computational Approach." *Pragmatics & Cognition* 9(1): 37-68

Bobrow, D.G., R.M. Kaplan, M. Kay, D.A. Norman, H. Thompson and T. Winograd. 1977. "GUS, A Frame-Driven Dialog System." *Artificial Intelligence* 8: 155-173.

Bos, J., E. Klein, O. Lemon, T. Oka. 2003. "DIPPER: Description and Formalisation of an Information-State Update Dialogue System Architecture". In *4th SIGdial Workshop on Discourse and Dialogue,* 115–124. Sapporo, Japan.

Bunt, H.C. and A.F.V. Van Katwijk. 1979. "Dialogue acts as elements of a language game." In *Linguistics in the Netherlands 1977-1979,* ed. by W. Zonneveld and F. Weerman, 264-282. Dordrecht: Foris Publications.

Ginzburg, J. 2012. *The Interactive Stance: Meaning for Conversation*. Oxford: Oxford University Press.

Hamblin, C.L. 1970. *Fallacies*. London: Methuen.

Gregoromichelaki, E., R. Kempson, M. Purver, G. J. Mills, R. Cann, W. Meyer-Viol, P. G. T. Healey. 2011. "Incrementality and Intention-recognition in Utterance Processing." *Dialogue and Discourse* 2(1): 199–233.

Kempson, Ruth, Wilfried Meyer-Viol, and Dov Gabbay. 2001 *Dynamic Syntax: The Flow of Language Understanding*. Oxford: Blackwell.

Kuyten, P., T. Bickmore, S. Stoyanchev, P. Piwek, H. Prendinger, and M. Ishizuka. 2012. "Fully automated generation of question-answer pairs for scripted virtual instruction." In *Proceedings of the 12th*

International Conference on Intelligent Virtual Agents, 12 - 14 September 2012, Santa Cruz, CA, USA.

Larsson, S. and D. Traum. 2003. “The information state approach to dialogue management.” In *Current and New Directions in Discourse & Dialogue*, ed. by R. Smith and J. Kuppevelt, Dordrecht: Kluwer academic publishers.

Leuski, A. and D. Traum. 2008. “A statistical approach for text processing in virtual humans.” In *Proceedings of the 26th Army Science Conference*, Orlando, Florida, USA, December 2008.

OED Online. June 2015. “dialogue | dialog, n.” Oxford University Press. http://www.oed.com.libezproxy.open.ac.uk/view/Entry/51915?isAdvanced=false&result=1&rskey=Y4sGSP& (accessed June 29, 2015).

Piwek, P. 1998. *Logic, Information and Conversation*. PhD Thesis, Eindhoven: Eindhoven University of Technology.

Piwek, P. and S. Stoyanchev. 2010. “Generating Expository Dialogue from Monologue: Motivation, Corpus and Preliminary Rules.” In *11th Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL),* Los Angeles, USA, 333-336.

Piwek, P. 2011. “Three Principles of Information Flow: Conversation as a Dialogue Game.” In *Perspectives on Information* , ed. by M. Ramage and D. Chapman, 106-120. New York and London: Routledge.

Power. R. 1979. “The organisation of purposeful dialogues.” *Linguistics*, 17:107–152.

Plüss, Brian and Paul Piwek and Richard Power. 2011. “Modelling Non-Cooperative Dialogue: the Role of Conversational Games.” In *Proceedings of the 15th Workshop on the Semantics and Pragmatics of Dialogue*, 212–213, Los Angeles, California, 21–23 September 2011.

Plüss, Brian. 2014. *A Computational Model of Non-Cooperation in Natural Language Dialogue*. PhD thesis, The Open University, Milton Keynes.

Reeves, B. and Nass, C. 1996. *The Media Equation: How People Treat Computers, Television, and New Media Like Real People and Places*. Cambridge: Cambridge University Press.

Stenius, E. 1967. “Mood and language game.” *Synthese* 17 (1): 254 - 274

Turing, A. 1950. “Computing Machinery and Intelligence.” *Mind*, LIX vol. 2236: 433–460. (Citation from *The Philosophy of Artificial Intelligence,* 1990. ed. by M.A. Boden. Oxford: Oxford University Press.)

Twain, M. 1919. *What is Man?* London: Chatto & Windus.

Weizenbaum, Joseph. 1966. “ELIZA – A Computer Program For the Study of Natural Language Communication Between Man and Machine.” *Communications of the ACM* 9(1): 36-45.

Wittgenstein, L. 1958. *The Blue and Brown Books: Preliminary Studies for the ‘Philosophical Investigations’*. Oxford: Blackwell. (Citation from the 2nd Edition, 1994.)