Sign Systems Studies 48(1), 2020, 79–100

Chatting with chatbots: Sign making in text-based human-computer interaction

Dorthe Duncker[1]

Abstract. This paper investigates the kind of sign making that goes on in text-based human–computer interaction, between human users and chatbots, from the point of view of integrational linguistics. A chatbot serves as a "conversational" user interface, allowing users to control computer programs in "natural language". From the user's perspective, the interaction is a case of semiologically integrated activity, but even if the textual traces of a chat may look like a written conversation between two humans the correspondence is not one-to-one. It is argued that chatbots cannot engage in communication processes, although they may display communicative behaviour. They presuppose a (second-order) language model, they can only communicate at the level of sentences, not utterances, and they implement communicational sequels by selecting from an inventory of executable skills. Instead of seeing them as interlocutors *in silico*, chatbots should be seen as powerful devices for humans to make signs with.

Keywords: chatbots; human-computer interaction; communicating participant; natural language processing; artificial intelligence; integrational linguistics

Introduction

In this paper, I will consider communication in text-based human–computer interaction from the point of view of integrational linguistics (e.g., Harris 1981, 1996, 1998; Pablé, Hutton 2015) – in particular, communication between a human user and a chatbot. A chatbot is a chat robot, a conversational tool that allows users to operate computers "with simple human language that people can understand" (Winchurch 2019).[2]

[1] Department of Nordic Studies and Linguistics, University of Copenhagen, Emil Holms Kanal 2, DK-2300, Copenhagen S, Denmark; e-mail: duncker@hum.ku.dk.

[2] Winchurch, Emily 2019. What's happening in conversational AI. *IBM, Conversational Services*. Available at: https://www.ibm.com/blogs/watson/2019/02/whats-happening-in-conversational-ai/.

https://doi.org/10.12697/SSS.2020.48.1.05

The integrational focus is on human communication, on signs made by human beings. This is not to say that humans are claimed to be the only sign makers, only that integrational theory, according to Roy Harris (1996: 50), is "agnostic on such issues". Presumably, other living beings also make signs, but their integrational basis differs from that of humans. On the other hand, integrationism does not try to "specify what forms of communication *only* human beings can engage in" (Harris 1996: 50). The assumption is rather that "languaging [speaking, listening, writing, reading, etc.; Harris 1981: 36] is grounded in intraspecific – including intrapersonal – communication, and that communication (*sensu latissimo*) is fundamental to human existence" (Love 2017: 116).

In integrational theory, communication is treated as "including all processes in which human activities are contextually integrated by means of signs" (Harris 1996: 7). Signs do not exist prior to concrete episodes in which human beings communicate; they are the products of communication, "not its prerequisites" (Harris 1996: 7). There is no fixed inventory of pre-existing signs for the communicating participants to *use*. Signs only come into existence because they are *made* by the participants, and no signs escape the communicative episode in which they were made. "A sign is not a sign until it is contextualized" (Harris 2009: 72). This also means that "the signification of a sign has no other source than the contextualized complex of activities which it integrates" (Harris 1995: 88).

The idea that two or more activities become integrated suggests that without the integration they would have remained separate, thereby preventing a person from doing something. When they are integrated, they produce in combination "results which could not have been achieved by any of those single activities independently" (Harris 2009: 69). For example, if I want to eat, I need to get food into my mouth somehow, before I can start chewing it. If I hold the food in my hand, I will have to coordinate my hand-to-mouth movements so the food ends up in my mouth and, say, not on my shoulder or on the floor.

When activities are integrated *semiologically*, the person in question needs to work something out based on an interpretation of an observation of some sort. If I draw the curtains one autumn morning and see frost on the lawn, I may take this as indicating that the night has been cold, and I may further decide on that basis to turn off the outside tap for the winter. In that case, only one person is involved in the process, and therefore there is only one sign maker. The lawn is not communicating with me. I am the observer and the interpreter. I assign meaning to my observation, given the circumstances that apply to the time of my observing. If I have never seen this kind of white substance before, if I have no previous winter experience with outdoor plumbing, if this is not my house, etc., I may interpret the observation differently and integrate it into an entirely different programme of activity.

When two or more persons need to integrate their activities, the situation becomes much more complex, and this is when semiological integration becomes of essence (Harris 2009: 69).

> A communication process, for the integrationist, is defined by the activities it integrates, the particular constraints on integration involved, and the signs produced to implement the process. We communicate with others, i.e. engage in the process, either by taking an initiative to which others construct an integrated sequel, or else by constructing such a sequel ourselves in response to the initiative taken by someone else. (Harris 1996: 63)

In interpersonal communication, one person (*A*) does something and this something is treated by another person (*B*) as a communicational initiative. Whatever *A* did is treated as "sign material" by *B*. This involves *B* projecting a sequence of events into which *A*'s initiative fits and then producing an integrated sequel to complement the initiative, by contributing "to the sequence of events the first is interpreted as projecting" (Harris 1996: 70).

It may make no difference to my deciding to turn off the outside tap whether I noticed the frost on the lawn myself or someone else drew my attention to it, but it does make a difference to the integrational function of the sign. In the former case, I act based on an observation I made and interpreted myself, whereas in the latter case, I respond to a communicational initiative from another *person* (Harris 1996: 58).

Integrational theory only considers communication processes in which either one person's activities are integrated "with objects and events in the physical world" (Harris 2009: 72), or processes that involve "the integration of activities between individuals" (Harris 2009: 68). That is, a person can be semiologically engaged either in *environmental* integration or in *interpersonal* integration.

Although the integrational position is to remain "agnostic on such issues", the distinction between environmental and interpersonal integration does not account for situations in which humans engage in communication processes with non-humans. Pet owners have conversations with their pet animals, some people talk to their plants or their cars (Bade 2012: 370–371), and literature from Aesop to La Fontaine to Disney and beyond is populated with talking animals and objects. Humans readily engage in communicative processes with non-humans or even non-animate entities, whether fictional or real. The expressivity of objects invites animation. According to Bettina Perregaard, "[w]e perceive of objects as expressive, because our perception has a physiognomical character", and this is why we are able "to see likenesses and make analogies between circumstances and objects that have certain expressive features in common, although they are otherwise quite different" (Perregaard 2016: 15; my translation, D. D.).

However, apart from the talking animals and objects of our own imagination or in fiction, people do not expect their pets, plants, cars, etc., actually to respond verbally (maybe excepting some parrots). Talking machines represent an intriguing case in this respect. For centuries, people have been fascinated by the idea of such machines. It has become a well-known trope in science fiction, and descriptions of mechanical word processing devices can be traced back at least as far as the language machine of Lagado in Jonathan Swift's *Gulliver's Travels* from 1726 (Weiss 1985; Harris 1987: 10ff). Unlike your cat or the weeping fig in your living room, the whole fascination of the talking machine is that it *does* respond verbally when addressed by a human operator. Artificial intelligence (AI) and natural language processing (NLP) researchers have been experimenting with computer programs designed for "natural language" human-computer interaction since the 1960s. Joseph Weizenbaum's (1966) amazingly simple program ELIZA and Terry Winograd's (1971) SHRDLU still stand as landmark dialogue systems. However, the technology has not been widely available until chatbots came to be applied commercially a few years ago (business-to-consumer), and since then they have proliferated all across the Internet (see Fig. 1).

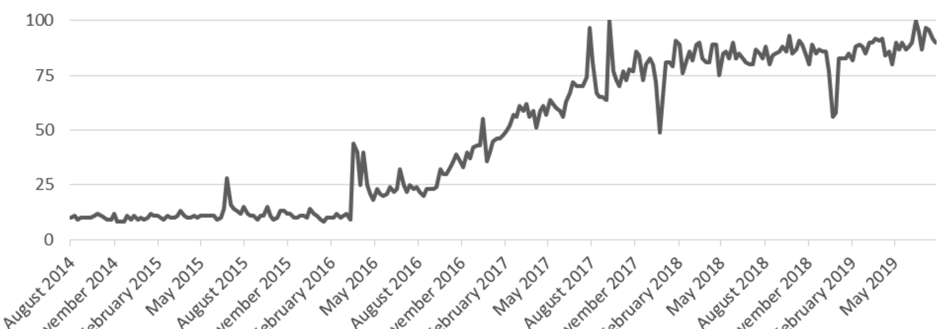

Figure 1. The relative popularity of the search term 'chatbot' worldwide over the past five years (data from Google Trends).

In order to appreciate what is special about communicating linguistically with a machine, consider the textual interaction in Fig. 2. From the look of it, this could derive from a chat or a text conversation between two persons. It shows a dialogic question–answer format known from computer-mediated interpersonal communication, and it ends in an exchange of politeness. However, only the gray bubbles are written by a human; the white ones are produced by a chatbot. This transcript seems to have all the characteristics of a written conversation between two humans. The user asks a series of weather-related questions, the bot answers, and, from what we can tell from reading the textual traces of the event, the answers are adequate and satisfactory.

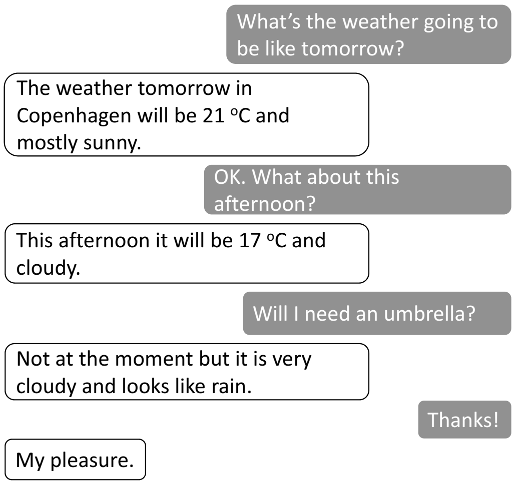

Figure 2. Text-based interaction between a human (gray bubbles) and a chatbot (white bubbles).[3]

The question is what status this kind of "conversation" has. Does the bot produce integrational sequels in response to the user's communicational initiatives? Does the user treat the bot-generated responses as contributions from an interlocutor? Can the bot read what the user writes? If, in fact, the bot is capable of making integrational sequels, this implies that it is also capable of sign making.

Three user interfaces to the universal machine

Machines do not necessarily have to perform an activity the same way as the mimicked agent. After all, airplanes do not fly like birds do. The point is to have the machine produce a functionally equivalent activity. Nevertheless, we often talk

3 The transcript is an edited version of a session between myself and the chatbot Mitsuku (https://www.pandorabots.com/mitsuku/). Mitsuku responds to weather-related questions by linking to an external weather service. The response format of the first two questions is taken from the weather bot Poncho (see Fig. 3).

about machines in terms of the activities and the agents they take from. Planes *fly*, they have *wings*, a *body*, and a *tail*. We *ride* cars although no horses are involved, and cars have *horsepowers*. We often talk about new inventions or unfamiliar phenomena in metaphorical pictures. Talking about automated activities in birdy or horsey terms may be quite harmless and inconsequential, but things get more complicated when the activity in question is not only something humans can engage in, but something *only* humans can engage in.

Unlike most other electronic appliances in our everyday life, computers are multifunctional devices. They are "universal machines" (Turing 1950: 441). The computer is built to execute programs, and each computer program typically has a functionality of its own. What this means is that, in order to carry out different computing processes, it is not necessary to build a new computer (hardware) for each process. You can simply write another program (software) and run it on the same physical machine. From the user's point of view, it may seem as if the user interacts with the hardware, but it is the functionality of the software that defines what kind of things the user can do with the computer, and a user interface is required. A user interface is "a collection of techniques and mechanisms that allow the user to interact with the computer program" by acting as an "intermediary" between the program and the user (Zhang, Gourley 2009: 89).

A user interface consists (minimally) of an input device and an output device. With a computer keyboard as the input device, the user can press a key on the keyboard and have the corresponding character displayed on the output device, the computer screen. In a "command line interface", the user enters a sequence of characters and sends it by pressing a particular key (the "enter key"). A computer program interprets the user's input as a command to perform the task specified in the input. Depending on the functionality of the program, it will display the result of the operation on screen, prompt the user for additional input, or simply confirm that the process has ended, but usually it will provide the user with some kind of textual feedback concerning its current state.

What this means is that the human user *writes* to the machine, and that feedback and results of the desired operation are displayed in a format the user can *read*.[4] Writing and reading are distinctive human skills and, before the advent of digital computers, they were exclusively exercized in human-to-human

[4] With speech recognition (and speech synthesis) software, computers can also accept acoustic input (and produce acoustic output). These technologies can be applied to voice user interfaces that allow humans to interact with computers acoustically. However, the function of these interfaces is to convert the speech signal to (glottic) text and vice versa (e.g. Rao, Vuppala 2014: 65), which means that, effectively, the interaction is still text-based and the voice user interface just adds an extra layer of information processing.

communication. However, since the computer can output in a format readable to humans, it seems an obvious conclusion to regard the output as the computer's written *response* to the user, suggesting that the computer is proficient in reading and writing and that it has the capacity to engage in *linguistic* communication. (The user issues a written *command* and a computer program *interprets* and executes it.) Considering this apparent comparability, perhaps it is no big surprise that it has become common practice to attribute human characteristics to computing technologies. The computer is "friendly", "smart", "it catches and transmits viruses", etc. (Marakas, Johnson, Palmer 2000: 720). It seems that we understand what a computer is through the metaphoric picture we have come up with: the picture of the human being. In particular, we see the computer in the picture of the human as "homo communicator" (Harris 1996: 13), and we talk about computer matters in terms of "language-based activities" (Harris 1987: 21).

A command line interface is not a very user-friendly interface. It excludes users who are not familiar with the *command language syntax* or its defined *vocabulary*. Since the 1980–1990s, another generation of user interfaces, "graphical user interfaces", have been developed with the goal of making it easier for users to interact with computer programs through graphical elements displayed on the screen. With graphical user interfaces also a new input device was introduced, the computer mouse. In addition to typing commands on the keyboard, the user can interact with the program by pointing and clicking on various textualized or iconic graphical objects on the display. Although graphical user interfaces are relatively self-explanatory and constitute an improvement over command line interfaces, users are still required to possess a high degree of computer literacy. They need to familiarize themselves with the functionality of the program, find out where and under which label particular functions are located, which functions are associated with which graphical objects, and find out what happens when this or that object is clicked.

Recently, a third generation of user interfaces has become popular, "conversational user interfaces". Chatbots are conversational user interfaces. Here the idea is that users are not required to possess a level of technical expertise that goes beyond that of texting on their mobile phone; they are not even expected to concern themselves with how the program works. Instead of "communicating with a computer on its own inhuman terms – by clicking on icons and entering syntax-specific commands – you interact with it on yours, by just telling it what to do" (Brownlee 2016).[5] In this way, chatbots "make software accessible

[5] Brownlee, John 2016. Conversational interfaces, explained. *Fast Company*. Available at: https://www.fastcompany.com/3058546/conversational-interfaces-explained/.

to everyone who understands human language" (Winchurch 2019). With the increasing deployment of conversational user interfaces (cf. Fig. 1), the anthropomorphization of computers seems to have come full circle. These modern dialogue systems can "participate in conversation", "they are able to share personal opinions, relay experience of family dramas, be relevant, but also be vague, and mislead just as humans do" (Shah *et al.* 2016: 278). However, there is a giant leap from, on the one hand, perceiving of a computer as an electronic device with the capacity to process and execute fixed-code "syntax-specific commands", to, on the other hand, perceiving of it as an (artificially intelligent) entity "who" possesses what it takes to refer to itself as "I".

The human computer

Before 1940, a computer was the name of an occupation held by people who earned their living by calculating. Computers were hired by scientists to make the calculations they required, and the new electronic computers were built with the goal of carrying out "any operations which could be done by a human computer" (Turing 1950: 436). At first, the electronic computers were just that, machines to calculate with, but soon it was realized that they could be put to other uses besides calculating. By 1950, machines had been constructed that could "in fact mimic the actions of a human computer very closely" (Turing 1950: 438), and Alan Turing pushed the question of their development potential further by proposing a test, an "imitation game", in which a computer program should fool a human into believing that the program was another person (Turing 1950: 442). If a human judge cannot tell the difference between a human correspondent and a computer program, then the computer has passed the test. Since 1991, an annual competition, the Loebner Prize, has awarded prizes to the chatbots that came the closest to passing the Turing test.[6] So far, no chatbot has managed to fool the judges and only bronze medals have been awarded.

Turing's approach aims at convincing the reader that digital computers are just as reliable as human computers and that their computational architecture is a functional match to the human (biomechanical) layout (Turing 1950: 436–438). Human computers are "supposed to be following fixed rules", a "book of rules". Digital computers follow "an instruction table", i.e. a computer program. Human computers make calculations with pen and paper, or in their head, and similarly computers "store" calculations in their (read/write) "memory", together with their

6 The competitions are organized by the Society for the Study of Artificial Intelligence and Simulation of Behaviour (see https://www.aisb.org.uk/events/loebner-prize).

program. The rules change with each new job, and "[i]f one wants to make a machine mimic the behaviour of the human computer in some complex operation one has to ask him how it is done, and then translate the answer into the form of an instruction table" (Turing 1950: 438).

In this way, the digital computer is *moulded* in the image of the human computer, and digital computing is to be understood in human terms. However, this is not to understand the computer metaphorically through the human, but rather to apply the human as a *model* for the construction of the computer (Harris 1987: 20). A metaphor only works as long as the two domains of comparison differ from each other. Once they overlap, the metaphor becomes redundant. Computation is not explained as a mechanical enterprise but as a human skill set a machine can be instructed to mimic. This overlap also helps to explain why the machines succeeded so quickly in replacing their human predecessors,

> because they copied the sequential architecture of human computers. In so doing, they inherited all the sequential ways of expressing and formulating science that had developed over twenty-five hundred years, a period in which computers shaped science far more than science shaped computers. In effect, the architects of the 1940s packaged their wonderfully speedy electronic circuits in anthropomorphic forms to meet an existing market. (Bailey 1992: 67)

The digital computer became a success because human computation was already being conceived of as a mechanical sequential operation with the implication that, eventually, "many activities traditionally regarded as characteristically human, and hence mechanically inexplicable, will turn out to be at bottom mechanical after all" (Harris 1987: 21). This includes language-based activities.

Turing silently assumes that the players in his imitation game are communicationally proficient in writing, although the ability to answer (written) questions and provide (written) responses was probably not listed explicitly in the job descriptions of the human computers. Thus, it seems that he takes it for granted that since human computers have linguistic capacities, besides their capacities for computation, these extra capacities are automatically included when their computational capacities are "translated" to the computational architecture. The human computers' language is just another "book of rules", just another "instruction table" for the digital computer to store in its "memory" in order enable it to carry out language operations.[7] The post-Renaissance view of language and

7 Linguists and information scientists were not slow to pick up this baton. In the early days of the Cold War, some of the first efforts within the rising discipline of "computational linguistics" (originally "algebraic linguistics") were invested in attempts to automate translation from Russian to English (Bar-Hillel 1951).

linguistic communication made this assumption quite straightforward. A virtual "language machine" had already been up and running for centuries, envisaging "language as the product of mysterious inner machinery, run by programs over which human beings have no control" (Harris 1987: 171). In particular, Ferdinand de Saussure's (1983[1922]: [27–29]) "speech circuit" proved Turing right in this assumption. Saussure's detailed description of the "life of the language" can be seen as a blueprint predating the actual construction of digital language machines. However, Turing failed to include one particularly important aspect in the goal for digital computers. While humans *can* be made to follow fixed rules with algorithmic precision, they are only required to do so under very specific circumstances – for example when doing calculations – whereas computers *cannot not* follow rules. A computer is incapable of deviating from its programming.

Chatting in writing

Besides presupposing participants, communication processes for the integrationist presuppose "some biomechanical capacity or capacities on the part of the participants" (Harris 1996: 44) relating to "the physical and mental capacities of the human being" (Harris 1996: 28). Since computers are not biological entities, *bio*mechanical factors do apply to them, and this suggests that the only communicating participant in human–computer interaction is the human. The reason why the question of the participant status of the computer even comes up as an issue is that communication processes that involve the use of machines differ from other cases of single-participant processes. When humans operate machines, it seems as if the user integrates his or her activities with activities of another entity, another *agent*. The machine *does* something.

Probably only a few people would consider their coffee machine a communicating participant. The coffee machine is made for making coffee. It is a tool, just like a knife or a hammer are tools, the difference being that electrical tools *do* things for you, hand tools do not. The coffee machine does not brew coffee because you ask it to, but because you press the start button.[8] A conversational user interface raises the question of participation status in a different way, because the interface requires the user to engage in *linguistic* communication, in what appears to be inter*personal* communication with an entity that is not another person.

In integrational theory, writing as a form of communication comprises three phases called "forming", "processing", and "interpretation" (Harris 1995: 63–68).

[8] Although you may control your "smart" coffee machine with vocal commands, with a voice user interface such as Apple's Siri, Amazon's Alexa, or Google Assistant.

The forming phase results in a new spatial organization of an originally blank input field. This happens when the user hits the keys on the computer keyboard and the corresponding characters show up on the display in the order of entry. Biomechanically this phase requires the user to integrate different forms of activities, especially hand–eye co-ordination in order to press the right keys in the right order. When the characters are displayed, the writer can *see* them, and this leads to the second phase, the processing, in which the writer engages in optical scanning.

The display allows the writer to *read* what has been written so far (indicated by the position of the cursor) and to check whether the displayed characters are the ones the writer wanted to write and that the text says what the writer wants it to say. If not, the writer can resume forming and edit the text by reorganizing the characters. This requires the writer to navigate the text spatially, by relocating the cursor, and to delete, replace, or move some of the characters. Thus, forming includes any activity involved in producing a written form and processing includes any activity that allows the reader to recognize the units of writing, i.e. the characters on the display, and to recognize "the patterns into which these units are organized for purposes of articulating the message" (Harris 1995: 65). In practice, the writer may switch between forming and processing any number of times.

When two participants, *A* and *B*, are chatting or texting with each other, *A*'s text will not show up on *B*'s display until *A* has pressed the send key. At this point, *B* may begin to read what *A* has written, in which case *B* will engage in optical scanning of the text as a preparatory step in the construction of an integrational sequel in response to *A*'s initiative. After the processing phase, *B* will take the reading to the next stage, the interpretation phase. Here, *B* must do whatever it takes to assign meaning to *A*'s words. The scanning of *A*'s text is part of reading, but in order to complete the sequel, *B* also needs to find out what to *do* with *A*'s message.

Interpretation presupposes processing, but does not automatically follow from it. The integrational concept of interpretation extends beyond the everyday notion of text "comprehension" by including whatever is involved in *doing* something about the message, and "[d]oing things with words involves integrating them into a communication process" (Harris 1998: 91). If I text you to buy a bottle of red wine for tonight, and you buy the wine and bring it, your buying and bringing the wine will not be considered something extra to your interpretation of my request. In order to fulfil an integrational sequel, something more than grasping what the text says may or may not be required, depending on the circumstances (Harris 1996: 72). In this case, when I write to you and ask you to buy the wine, your buying it is the course of action I envisage in my message. "Communication, for the integrationist, essentially involves the *anticipated* integration of certain forms of activity" (Harris 1998: 118; my italics, D. D.).

This roughly describes the course of action in which two communicating participants engage in written computer-mediated communication, but what happens when *A* is human and *B* is a chatbot? In the example in Fig. 1, *A* writes: "What's the weather going to be like tomorrow?", and *B* writes back: "The weather tomorrow in Copenhagen will be 21 °C and mostly sunny". From this response, nothing suggests to *A* that *B* is not a communicating participant. It appears that *B* complies with *A*'s request and treats it as anticipated by *A*. *B* answers the question and returns the kind of information *A* is asking for. If *B* had been human, this course of action would have required *B* to read and interpret *A*'s message, and to write the response.

Obviously, the biomechanical factors that condition *A*'s participation in the communication process do not apply to *B*. Being a computer program, the chatbot has no eyes, no hands, no body. This means that the bot cannot engage in optical scanning, but must engage in some other form of processing. This processing is *digital*. The user's input is sent via the keyboard and received by the program as a sequence of characters, a "string", to be processed. What happens next depends on how the chatbot is programmed.

Constructing a language model

Two main types of dialogue systems, "service chatbots" and "social chatbots", may be distinguished, based on their functionality and their software architecture. Service bots are task-completion systems that allow users to access the functionality of a backend program, e.g. an online customer service, a banking service, a weather service, etc. They can be thought of as very user-friendly command line interfaces, because commands are entered in a "natural language". The functionality of social chatbots is different, although these bots may also complete tasks. "The primary goal of social chatbots is to be AI companions to humans with an emotional connection" (Shum *et al.* 2018: 14).

The conversational "skills" of chatbots can be implemented by scripting, training, or a combination. Scripted chatbots follow a conversational path defined by a set of heuristic rules consisting of question-answer pairs.[9] The conversational

[9] Weizenbaum's ELIZA was hand-scripted, and scripted chatbots generally perform well in the Loebner competitions. The chatbot A.L.I.C.E., a three-time winner of the Loebner Prize, may be seen as an extension of ELIZA (Wallace 2008: 182). The chatbot Mitsuku (cf. Footnote 3) is an A.L.I.C.E. type bot; it won the Loebner Prize in 2013, 2016–18.

rules may be either hand-written or extracted automatically from sets of human-to-human dialogue data and subsequently edited. For the bot author ("botmaster") writing such "responses is more like writing literature, perhaps drama, than writing computer programs" (Wallace 2008: 204). The quality of the conversational performance of the bot is monitored by the botmaster who "creates new AIML [:the markup language for the rules] content to make the responses more appropriate, accurate, believable, 'human', or whatever the botmaster intends" (Wallace 2008: 182).

Trained chatbots apply machine learning and "big" conversational data harvested from social network sites, public forums, news comments, etc., on the Internet. From the collected question–answer pairs two kinds of resources can be constructed: a database from which types of questions [based on patterns of similarity across question tokens (strings)] can be recognized and their corresponding answers retrieved; and a response generator that is "*trained* using the paired dataset to learn to simulate human conversation, and is able to generate responses for any topics including those that are unseen in human conversational data" (Zhou *et al.* 2018: 10). Once these databases are constructed, the result is a model of *a language* created automatically out of (traces of written) *language* (cf. Love 1990: 100).

This model has remarkably much in common with the language system as envisaged by Saussure. The model collects the "social product", "a fund accumulated by the members of the community through the practice of speech, a grammatical system existing potentially in every brain, or more exactly in the brains of a group of individuals" (Saussure 1983[1922]: [30]). As a "collective phenomenon" the language is "rather like a dictionary of which each individual has an identical copy" (Saussure 1983[1922]: [38]). This "dictionary" can be accessed by every instance of the chatbot, i.e. by each digitally identical copy of the bot software.

It might be objected that the collected question–answer pairs are *parole* data, that they are "no more than an aggregate of particular cases" and that therefore there is "nothing collective" about them (Saussure 1983[1922]: [38]). However, in order to build a language model from the collected data, the model building software needs to make sure that the data are in fact "naturally" linguistic, and that they only manifest one language. Otherwise, the chatbot may output an answer in one language to a question in another, it may mix languages, or it may output an answer that belongs to no language. Unless the program can control the linguistic status of the data on which its training rests, it cannot "learn" the language and, hence, it cannot construct a model for it.

This means that before the data set can be used as training material, the individual question and answer strings must be segmented into words ("tokenized") and *verified* according to a particular (institutionalized) codification of the language, e.g. a "machine-friendly" dictionary. But the program cannot look up a tokenized unit in the dictionary unless it has determined its entry form (i.e., the unit must be "lemmatized"), and it can only look up a word if there is an exact match between the source string and the target string. Therefore, the verification process may require the tokens to be "normalized" in order to get rid of any orthographical anomalies and other kinds of "noise". (Words that have no dictionary entries may still be recognized as valid, e.g. through morphological analysis, and their part of speech may be determined on the basis of their surroundings.) After the tokens have been identified as word forms of the language in question, the program also has to control the grammaticality of the sentences. This requires a grammar for the language that can be used by a parser to assign a constituent structure to the sentences in the data. Other kinds of filtering may be applied as well, e.g. to exclude inappropriate content or personally identifiable information (Zhou *et al.* 2018: 10).

In addition to extracting training pairs from human-to-human conversational data, the bot may also use its own interactions with users as training data. This means that the bot "learns" from the users it chats with. This feature is often mentioned when chatbots are promoted, e.g. the banking bot Kai:

> Kai helps you manage money, track expenses, and make payments, […] Kai is not your average bot because it understands and produces conversational language naturally, and knows banking inside and out. You just type to Kai as you would to a friend, no prompts required. And, as you chat, Kai gets smarter, learning from your conversations eagerly. […] Kai responds in raw text form. We chose this format to showcase the power of pure conversation. […] [I]nteracting with Kai feels just like texting a friend. (Gorelov 2016)[10]

Since Kai already "knows banking inside out", the bot, presumably, is not meant to get smarter banking-wise from the conversations with users (bank customers). Rather, the idea seems to be that it gets *conversationally* smarter because it is designed to incorporate the user's input and thereby enrich and calibrate its existing conversational language model, e.g. by increasing its inventory of words and word combinations.

However, the machine cannot *learn*, in the sense of getting smarter, if it cannot recognize the units of writing in the user's input (the "user-generated" content),

---

[10] Gorelov, Zor 2016. Say hello to Kai. *Kasisto*. Available at: https://kasisto.com/blog/say-hello-to-kai/.

because then it cannot decide what to *do* with the input. Therefore, when users produce "noisy" input, it may be necessary to "clean up" an input string for the sake of the language model before the input is being processed, i.e. to pre-process it. Effectively, this corresponds to an automatic proofreading and reorganizing of the user's text, which means that the chatbot re-engages in the forming phase on behalf, but independently, of the original writer. Human readers may also engage in proofreading, of their own or other authors' writings, but not as a prerequisite for reading the text themselves. Proofreading is a kind of linguistic "toilettage". It serves the purpose of making a text compliant with a particular codification and, therefore, it "aims at a different kind of interpretation" (Harris 1995: 65).

All chatbots, whether scripted or not, follow a "dialogue policy", i.e. "the optimal action that the system should take at each possible dialogue state" (Misu *et al.* 2012: 84). This involves that the bot has to determine whether the user's input contains something it can respond to, i.e. a string that triggers any of the bot's built-in skills, including its conversational skills. The implemented skills also set the bar for the interpretation phase on the part of the chatbot with respect to what the bot *can* do with the user's words. Thus, interpretation for the bot consists in matching up user input with the relevant skill. In case a match is found, the program must check whether all required information to perform the tasks is available, or whether additional user input is required.

For example, assume that a weather bot's backend service has a skill called *WeatherForecast* and that this procedure requires two parameters, place and time, called *WeatherWhere* and *WeatherWhen* respectively. Equipped with this information, the program can search its weather database and retrieve information about the weather quality, *WeatherQuality*, and air temperature, *AirTemperature*, at a particular place and time. When the information is found and retrieved, the program can send it to the bot, so the bot can form it and present it to the user on the display. This means that forming for the bot is a slot-filling task that consists in generating a sentence-formatted response string that includes the result of the database query, e.g. by inserting it into a pre-formed response template "The weather *WeatherWhen* in *WeatherWhere* will be *AirTemperature* and *WeatherQuality*". If the backend program has access to information about the location of the user's computer (e.g. GPS data) and the system time of the user's computer, both parameters are satisfied. Otherwise, the bot will have to prompt the user to supply the missing information, e.g. by displaying a pre-formed question string for the user to respond to.

In the case that the bot cannot establish a match between the user input and the built-in skills, the bot programmer has to decide what kind of fallback responses to prepare in advance. Several scenarios might be anticipated: the user could be demanding a skill the program cannot perform because it is not implemented (e.g. "Show me the statistics for lightning strikes over the past five years in Sweden"); because it cannot be executed by a weather service or any other computer program (e.g. "Feed the cat"); or because the program cannot determine whether or not the input matches an implemented skill. In the first two cases, a response may be prepared in advance, but the last case is trickier, because the program is unable to determine its own current state.

If the user can easily read the message and the program cannot, the reading proficiency of the program is immediately suspect in the eyes of the user. Another problem may be that the program recognizes a demand for an implemented function in the input, but that the recognized function is not the one the user asked for. With a human correspondent, one might be inclined to say that the former would be a failure to "understand" the message, and the latter a case of "misunderstanding".

Figure 3. Text-based interaction between a human user (gray bubbles) and a weather service bot (white bubbles).[11]

[11] A screenshot of this session with the weather service Poncho can be found in several places online (the original entry, from 2016, seems to be https://gizmodo.com/facebook-messenger-chatbots-are-more-frustrating-than-h-1770732045). Poncho was launched in 2013 and shut down in 2018 when it was sold to a beverage brand (see https://www.fastcompany.com/40578222/exclusive-weather-app-poncho-has-been-acquired-by-a-boutique-beverage-company).

The transcript in Fig. 3 shows a session with a weather bot in which the bot is unable to comply with the user's request. Apparently, the bot mistakes 'weekend' in the user's first input for a place name, i.e. the value of the *WeatherWhere* variable, and the bot prompts the user for additional information, because it "was not able to find any results for that location". The user responds by providing the extra information by specifying that the relevant location is "Brooklyn". Then it turns out that the bot has assigned the current system time as a default value to the *WeatherWhen* parameter, and that it cannot extract the relevant value "weekend" from the input string. When the user asks explicitly for it to be included in the forecast response ("This weekend?"), the bot responds in a way that suggests a request for a repetition ("Excusez-moi?"). The user repeats ("WEEKEND"), and the bot requests a second repetition ("Sorry, dozed off for a second. What were you saying?").

The bot cannot respond to the user's initial request as anticipated because it does not recognize the unit 'weekend' and it therefore fails to establish the relevant linguistic facts, given the circumstances. It cannot integrate the unit into the ongoing communication process. This means that it cannot move on to the interpretation phase because it is stuck in processing. The bot's response repertoire includes (pre-formed) means to enquire about "what was said" and "what was meant", but merely implementing the ability to output such strings does not, in and by itself, equip the bot with the metalinguistic resources required to establish the relevant facts (see Harris 1998: 145–146).

In this example, the problem might have been solved if the bot had used a lexical database such as Wordnet[12] to look up 'weekend'. In that case, the bot could have retrieved information about 'weekend' and its semantic relations to other words. It could have made use of the information that 'weekend' is a kind of 'time period' (direct hypernym), that it is part of 'week' (part holonym), and that it covers 'Saturday' and 'Sunday' (part meronyms). However, before this information could be retrieved, the bot would have had to identify the relevant 'weekend' entry. The database lists two entries spelled 'weekend', a noun and a verb (with the direct hypernyms 'spend' and 'pass', and the sister terms 'soldier', 'slum', 'vacation', 'holiday', and others). In other words, it would have had to disambiguate the unit 'weekend' before it could parse the input string.

---

12 Princeton University 2010. About WordNet. Wordnet [https://wordnet.princeton.edu/]. Princeton University.

Chatting across two orders of language

Unless a chatbot has access to the *relevant* linguistic facts *prior to* the actual chatting, it is prevented from *doing* anything with the user's words and, hence, also prevented from integrating them into a communication process. In that case, all or parts of the user's input consist of unrecognizable strings. This means that for the chatbot communication *presupposes* a language.

In integrational theory, *a* language (countable) can be codified, but language (uncountable) cannot. A language is a second-order construct abstracted from utterances, i.e. from first-order language: "Such a construct may be institutionalized and treated as *the* language of a community. But it remains a construct based on an idea: at no point does it become a first-order reality for individuals" (Love 1990: 101). However, a chatbot cannot engage in linguistic communication on these terms. It is a prerequisite for the bot that it has access to a language model that allows it to treat the units of user-generated text as tokens of the model's linguistic types. For the chatbot, only one linguistic reality exists and that is the implemented codification of the second-order construct. This also means that the program can only engage with the user's utterances indirectly, in terms of the underlying abstractions their inscriptions manifest. Hence, the bot can only engage in communication at the level of *sentences*, not utterances.

The gap conversational user interfaces attempt to bridge spans across the orders of language, with the ultimate goal of reconciling the second order with the first. This is not, however, like trying to square the circle because this task is complex beyond compute. The difference between the communicational proficiency of humans and computers is a difference in kind, not complexity. Human sign making is not based on a second-order construct or limited by its pre-established pairings of form and meaning. "Each of us contextualizes in our own way, taking into account whatever factors seem to us to be relevant" (Harris 2009: 71). Such semiological luxury does not belong to digital computers.

When a chatbot complies with a written request, this may very well satisfy the user's communicational purpose and leave the impression that the bot completed an integrational sequel thereby fulfilling the interpretation phase as anticipated by the user. However, the bot does not interpret the user's input as an initiative that makes it project a series of events to which the generated output contributes as a complementation. It merely searches its inventory of implemented skills and determines whether a match can be found so that the corresponding skill can be performed.

The result of the interaction does not depend on the bot having engaged in the activities in which a human reader would have engaged in order to read the user's written message. The bot's dialogue manager treats the user's input as a digital object, as a string. It passes the object through a series of separate stages in which certain operations are performed on the object so that each next stage presupposes the operations of the previous stage (e.g., the input string must be tokenized before the individual linguistic units can be recognized). On a superficial view, it may appear as if the bot has indeed *read* the message, that it has engaged in processing and interpretation as a human reader would have done. However, the integrational conception of the activities that are being integrated through writing does explicitly *not* involve the idea of an *object* being processed. The three phases that are being distinguished as forming, processing, and interpretation are

> not phases which are to be envisaged as separate stages through which "the message" has to be passed, as if it were an object being sent along a conveyor belt in a factory and having something different done to it at various points along the way. (Harris 1998: 117)

All the same, this is exactly how text is treated by AI and NLP technologies.

The user, on the other hand, may read the bot's output as a response from a communicating participant, thereby assigning authorship to the program. The responsibility of this assignment, however, solely rests with the human reader. The bot can generate sentences, whether from scratch or from scripted templates. It can produce written forms and send them to the computer's display, but it cannot produce utterances. The semiological expressivity of the bot is limited to what David Bade calls the production of "unsigned" signs (Bade 2012: 372). The bot does not "know what it is saying". It is lost in "autoglottic" space (Harris 2000: 236).

There is no *one* at the other end for the human to communicate with. Human–computer interaction is not inter*personal* communication. There is only one communicating participant. The programmer is not sitting somewhere inside the executable source code. The user is not responding to the programmer's communicational initiative any more than someone who looks at their watch is "responding to the communicational initiative of the clock-maker when [they] take the position of the hands on the dial as indicating that it is ten past four" (Harris 1996: 58). If I leave home, taking my umbrella because I read in a weather bot's auto-generated response that it looks like it is going to rain; or because I listened to the weather forecast on the radio; or because I looked out of the window and saw "the clouds looming up from the west", then the three cases "present different sets of activities integrated", but they also have a factor in common: that I take my umbrella (Harris 1996: 63).

Planes fly differently from birds, and chatbots chat differently from humans. Chatbots exhibit conversational behaviour without engaging in communicative processes. They implement communication as the swapping of messages in a fixed code. For humans, however, there is no fixed code. "And if there is no fixed code, then *a fortiori* communication is not a matter of swapping messages in it" (Love 1990: 95). A computer is an extremely powerful semiological *tool*. It is a machine to make signs *with*, a "contextualization device" (Harris 2000: 242), not a machine itself capable of sign making. As pointed out by Trevor Pateman (1981: 10),

> it is to miss completely the possibilities opened up by AI to write programs as if they were pseudo-persons and talk about them as if they were real persons. No one has any problem in seeing that an adding machine is a thing to add with, and not a machine which adds.

If we think of text-based human–computer interaction in terms of a conversation – that is as inter*personal* communication – then we come to think of computers as something that they are not, and we come to think of humans as something that we are not.

References

Bade, David 2012. Signs unsigned and meanings not meant: Linguistic theory and hypothetical, simulated, imitation and meaningless language. *Language Sciences* 34: 361–375. https://doi.org/10.1016/j.langsci.2012.02.003

Bailey, James 1992. First we reshape our computers, then our computers reshape us: The broader intellectual impact of parallelism. *Daedalus* 121(1): 67–86.

Bar-Hillel, Yehoshua 1951. The present state of research on mechanical translation. *American Documentation* 2(4): 229–237. https://doi.org/10.1002/asi.5090020408

Harris, Roy 1981. *The Language Myth*. London: Duckworth.

Harris, Roy 1987. *The Language Machine*. London: Duckworth.

Harris, Roy 1995. *Signs of Writing*. London: Routledge.

Harris, Roy 1996. *Signs, Language and Communication*. London: Routledge.

Harris, Roy 1998. *Introduction to Integrational Linguistics*. Oxford: Pergamon.

Harris, Roy 2000. *Rethinking Writing*. London: Continuum.

Harris, Roy 2009. The integrational conception of the sign. In: Harris, Roy, *Integrationist Notes and Papers 2006–2008*. Gamlingay: Bright Pen, 61–82.

Love, Nigel 1990. The locus of languages in a redefined linguistics. In: Davis, Haley G.; Taylor, Talbot J. (eds.), *Redefining Linguistics*. London: Routledge, 53–117.

Love, Nigel 2017. On languaging and languages. *Language Sciences* 61: 113–147. https://doi.org/10.1016/j.langsci.2017.04.001

Marakas, George M.; Johnson, Richard D.; Palmer, Jonathan W. 2000. A theoretical model of differential social attributions toward computing technology: When the metaphor becomes the model. *International Journal of Human-Computer Studies* 52: 719–750. https://doi.org/10.1006/ijhc.1999.0348

Misu, Teruhisa; Georgila, Kallirroi; Leuski, Anton; Traum, David 2012. Reinforcement learning of question-answering dialogue policies for virtual museum guides. *Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL). Seoul, South Korea, 5–6 July 2012*. Association for Computational Linguistics, 84–93.

Pablé, Adrian; Hutton, Christopher 2015. *Signs, Meaning and Experience: Integrational Approaches to Linguistics and Semiotics*. (Semiotics, Communication and Cognition 15.) Boston: Walter de Gruyter.

Pateman, Trevor 1981. Communicating with computer programs. *Language & Communication* 1: 3–12. https://doi.org/10.1016/0271-5309(81)90003-3

Perregaard, Bettina 2016. *Vær i verden: Intentionelle og interaktionelle mønstre i barnets hverdagsliv* [*Being in the World: Intentional and Interactional Patterns in Children's Everyday Lives*]. Copenhagen: U Press.

Rao, K. Sreenivasa; Vuppala, Anil Kumar 2014. *Speech Processing in Mobile Environments*. Cham: Springer.

Saussure, Ferdinand de 1983[1922]. *Course in General Linguistics*. (Harris, Roy, trans.) London: Duckworth.

Shah, Huma; Warwick, Kevin; Vallverdú, Jordi; Wu, Defeng 2016. Can machines talk? Comparison of Eliza with modern dialogue systems. *Computers in Human Behavior* 58: 278–295. https://doi.org/10.1016/j.chb.2016.01.004

Shum, Heung-Yeung; He, Xiao-Dong; Li, Di 2018. From Eliza to XiaoIce: Challenges and opportunities with social chatbots. *Frontiers of Information Technology & Electronic Engineering* 19(1): 10–26. https://doi.org/10.1631/FITEE.1700826

Turing, Alan M. 1950. Computing machinery and intelligence. *Mind* 59: 433–460. https://doi.org/10.1093/mind/LIX.236.433

Wallace, Richard S. 2008. The anatomy of A.L.I.C.E. In: Epstein, Robert; Roberts, Gary; Beber, Grace (eds.), *Parsing the Turing Test: Philosophical and Methodological Issues in the Quest for the Thinking Computer*. Dordrecht: Springer, 181–210.

Weiss, Eric A. 1985. Jonathan Swift's computing invention. *Annals of the History of Computing* 7(2): 164–165. https://doi.org/10.1109/MAHC.1985.10017

Weizenbaum, Joseph 1966. ELIZA – A computer programme for the study of natural language communication between men and machines. *Communications of the ACM* 9: 36–45. https://doi.org/10.1145/365153.365168

Winograd, Terry 1971. *Procedures as a Representation for Data in a Computer Program for Understanding Natural Language*. Cambridge: MIT.

Zhang, Allison B.; Gourley, Don 2009. *Creating Digital Collections: A Practical Guide*. Oxford: Chandos Publishing.

Zhou, Li; Gao, Jianfeng; Li, Di; Shum, Heung-Yeung 2018. The design and implementation of XiaoIce, an empathetic social chatbot. *arXiv:1812.08989v1 [cs.HC]*: 1–26.

Общение с чат-ботами: Производство знаков в базирующемся на тексте взаимодействии человека и компьютера

В статье исследуется знакопроизводство при текстовом взаимодействии человека и компьютера (между человеком и чат-ботом) с точки зрения интеграционной лингвистики. Чат-бот служит «разговорным» пользовательским интерфейсом, позволяя пользователям управлять компьютерными программами на «естественном языке». С точки зрения пользователя, взаимодействие – это случай семиологически интегрированной деятельности, но даже если текстовые следы чата могут выглядеть как записанный разговор между двумя людьми, это впечатление обманчиво. Утверждается, что чат-боты не могут участвовать в коммуникационных процессах, хотя могут демонстрировать коммуникативное поведение. Они предполагают языковую модель (второго порядка), они могут общаться только на уровне предложений, а не высказываний, и они реализуют коммуникативные последовательности, выбирая из перечня доступных для исполнения навыков. Вместо того, чтобы рассматривать их как собеседников в виртуальном пространстве (*in silico*), чат-ботов следует рассматривать как мощные устройства, с помощью которых человек может производитьь знаки.

Vestlusrobotiga lobisemine: Märgiloome tekstipõhises inimese–arvuti koostoimimises

Artiklis uuritakse integratsioonilise lingvistika vaatevinklist märgiloomet, mis toimub tekstipõhises inimese–arvuti koostoimimises inimkasutajate ja vestlusrobotite vahel. Vestlusrobot toimib „konversatsioonilise“ kasutajaliidesena, võimaldades kasutajatel kontrollida arvutiprogramme „loomulikus keeles“. Kasutaja vaatevinklist on see koostoime semioloogiliselt lõimitud tegevuse juhtum, ent ehkki vestluse tekstuaalsed jäljed võivad välja näha nagu kahe inimese vaheline kõnelus, ei ole tegu üks-ühese vastavusega. Väidetakse, et vestlusrobotid ei suuda osaleda kommunikatsiooniprotsessides, kuigi nad võivad väliselt esitada kommunikatiivset suhtlemist. Nad eeldavad (teise astme) keelemudelit; nad suudavad kommunikatsioonis osaleda üksnes lause, mitte lausungi tasandil; ning nad teostavad jätkukommunikatsiooni, tehes valikuid sooritatavate oskuste nimistust. Selle asemel, et neid pidada vestluskaaslasteks *in silico*, peaks vestlusroboteid pidama võimsateks vahenditeks, mille abil inimesed märke loovad.