DAISY: An Implementation of Five Core Principles for Transparent and Accountable Conversational AI

Downloaded from: https://research.chalmers.se, 2023-01-21 01:03 UTC

Citation for the original published paper (version of record):
Wahde, M., Virgolin, M. (2022). DAISY: An Implementation of Five Core Principles for Transparent and Accountable Conversational
AI. International Journal of Human-Computer Interaction, In Press.
http://dx.doi.org/10.1080/10447318.2022.2081762

N.B. When citing this work, cite the original published paper.

research.chalmers.se offers the possibility of retrieving research publications produced at Chalmers University of Technology.
It covers all kind of research output: articles, dissertations, conference papers, reports etc. since 2004.
research.chalmers.se is administrated and maintained by Chalmers Library

(article starts on next page)

INTERNATIONAL JOURNAL OF HUMAN-COMPUTER INTERACTION
https://doi.org/10.1080/10447318.2022.2081762

Taylor & Francis
Taylor & Francis Group

OPEN ACCESS Check for updates

DAISY: An Implementation of Five Core Principles for Transparent and Accountable Conversational AI

Mattias Wahde[a] and Marco Virgolin[a,b]

[a]Department of Mechanics and Maritime Sciences, Chalmers University of Technology, Gothenburg, Sweden; [b]Life Sciences and Health Group, Centrum Wiskunde & Informatica, Amsterdam, the Netherlands

ABSTRACT

We present a detailed implementation of five core principles for transparent and acccountable conversational AI, namely *interpretability*, *inherent capability to explain*, *independent data*, *interactive learning*, and *inquisitiveness*. This implementation is a dialogue manager called DAISY that serves as the core part of a conversational agent. We show how DAISY-based agents are trained with human-machine interaction, a process that also involves suggestions for generalization from the agent itself. Moreover, these agents are capable to provide a concise and clear explanation of the actions required to reach a conclusion. Deep neural networks (DNNs) are currently the de facto standard in conversational AI. We therefore formulate a comparison between DAISY-based agents and two methods that use DNNs, on two popular data sets involving multi-domain task-oriented dialogue. Specifically, we provide quantitative results related to entity retrieval and qualitative results in terms of the type of errors that may occur. The results show that DAISY-based agents achieve superior precision at the price of lower recall, an outcome that might be preferable in task-oriented settings. Ultimately, and especially in view of their high degree of interpretability, DAISY-based agents are a fundamentally different alternative to the currently popular DNN-based methods.

1. Introduction

Conversational artificial intelligence (hereafter: conversational AI) is a rapidly growing field, with many relevant applications in health and well-being, education, customer service, tourism, personal digital assistants, and so on; see, for instance, the surveys by Laranjo et al. (2018) and Wahde and Virgolin (2022). In general, conversational AI systems (also called conversational agents) can be divided into the two categories of chatbots intended for casual conversation on everyday topics, and task-oriented agents intended to provide clear, consistent, and relevant information on specific topics such as, for example, giving medical advice, handling time table or reservation queries, providing technical support or other customer service, and so on. In this article, we focus on task-oriented agents.

Currently, research in conversational AI is to a strong degree focused on black box models such as deep neural networks (DNNs). Such systems are used for encoding statistical language models in a manner that makes it possible, for example, to maintain contextual information even in long sentences and to produce output that, in many cases, is indistinguishable from the response that a human would give (Otter et al., 2021). However, despite the success of DNNs in conversational AI, there are reasons to be concerned about their indiscriminate use: While such systems might be eminently suited e.g., for the task of casual conversation in chatbots, their black box nature makes them much less suited for the decision-making (cognitive processing) that underlies the responses given by task-oriented agents.

In fact, the same concerns can be raised for the entire AI field, beyond conversational AI. Applying black box models in any situation that involves high-stakes decision-making is fraught with danger (Rudin, 2019), partly because of the fundamental opaqueness of the decision-making in such systems, and partly due to the manner in which they are trained: Typically, training a black box model requires vast amounts of data that, in turn, may contain unwanted biases that are assimilated by the black box during training. In recent years, and in response to such problems, several approaches have been suggested for generating more transparent AI-based systems. Those approaches can broadly be divided into two categories (Barredo Arrieta et al., 2020), namely *explainable AI* (xAI) and *interpretable AI* (IAI).

In xAI, the aim is to provide some form of explanation of the decisions taken by a black box model, often involving a secondary model that somehow approximates the black box. Such considerations have resulted in a plethora of

CONTACT Mattias Wahde mattias.wahde@chalmers.se Department of Mechanics and Maritime Sciences, Chalmers University of Technology, Gothenburg, Sweden

© 2022 The Author(s). Published with license by Taylor & Francis Group, LLC.
This is an Open Access article distributed under the terms of the Creative Commons Attribution-NonCommercial-NoDerivatives License (http://creativecommons.org/licenses/by-nc-nd/4.0/), which permits non-commercial re-use, distribution, and reproduction in any medium, provided the original work is properly cited, and is not altered, transformed, or built upon in any way.

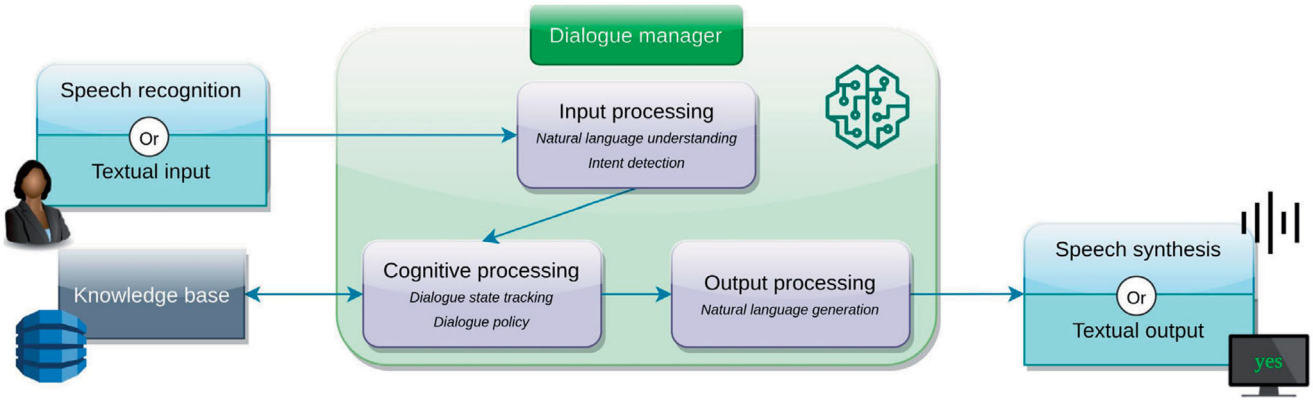

Figure 1. The pipeline model, showing the central position of the dialogue manager. In addition, task-oriented agents almost always require a knowledge base, whereas automated speech recognition and speech synthesis are somewhat peripheral.

approaches for explaining black box decision-making, at different levels and to varying degree. A detailed description of such approaches will not be given here, but the interested reader can find more information in a recent review (Angelov et al., 2021).

By contrast, in IAI, one seeks to circumvent the problem by avoiding DNNs altogether and instead building decision-making systems composed of interpretable primitives, such that the decision-making steps can be described in a human-understandable manner. It should be noted here that the nomenclature of these approaches is somewhat confusing, in that some authors use the terms *explainable* and *interpretable* more or less interchangeably, whereas others (us included) emphasize the difference between inherent interpretability, on the one hand, and explainability (of black boxes) on the other (Barredo Arrieta et al., 2020; Linardatos et al., 2020; Rudin, 2019).

Returning to the specific case of conversational AI, one may observe that, at present, this field is very strongly dominated by research into black box (DNN) approaches, to such an extent that neither IAI nor, in fact, xAI are given much consideration, with a few recent exceptions (Nobani et al., 2021; Wahde, 2019; Wahde & Virgolin, 2021; Werner, 2020). This is unfortunate: One may argue that it is, in fact, crucial that a developer should be able investigate how a conversational system works, making it possible to identify and correct errors, and to modify or extend the system as necessary. Moreover, it is equally important that a user should (perhaps upon request) be given a clear description of how a conversational system reached a particular conclusion. Alas, none of those conditions are fulfilled by the currently popular DNN-based systems, whose near-total opaqueness, combined with occasional catastrophic failures, make them unsuited for the types of applications just mentioned, as exemplified in Section 2. In addition, as briefly mentioned above, DNN-based conversational systems are generally trained using processes that requires large corpora of dialogue data, which can be hard to obtain for specific tasks. Even more importantly, the statistical language model encoded by the resulting DNN may (and often does) incorporate unwanted biases (e.g., racial or sexist biases) present in the training data. Once the DNN has been trained it is very hard to detect such biases a priori, meaning that the resulting system may, at any time and without warning, give a catastrophically incorrect output that, in turn, can have very negative effects on the users of the system (Bender et al., 2021).

Motivated by a desire to overcome the problems just described, in a recent article we introduced five principles that, in our view, should permeate any task-oriented conversational AI system, regardless of the specific implementation used (Wahde & Virgolin, 2021). Those principles, referred to as "the five Is"are: interpretability and inherent capability to explain that are meant to ensure transparency and accountability of a conversational agent during development and use, respectively; independent data that makes it possible to replace an agent's knowledge base without needing to modify its conversational capabilities, thus allowing re-use of existing agents; interactive learning and inquisitiveness (the latter term here used in the positive sense of the word, i.e., similar to an eagerness to learn), both of which provide a novel, transparent approach for training conversational agents, accessible even to non-experts. The five principles are described in greater detail in Section 3.

The core component of a task-oriented agent is the dialogue manager (hereafter: DM), which has the task of determining the user's intent, processing the user's request or statement in order to derive the necessary information required for the response, a process that typically involves access to the agent's knowledge base, and then, finally, formulating the output in human-understandable manner. DMs typically follow the socalled pipeline model (albeit only implicitly in the case of black box models of the kind described below), shown in Figure 1. Many DMs were developed prior to the advent of DNN-based systems, e.g., finite-state (Jurafsky & Martin, 2009), frame-based (Bobrow et al., 1977), plan-based (belief-desire-intent) (Allen et al., 2001; Bohus & Rudnicky, 2009), and agent-based (Blaylock, 2005) DMs; see also (Jurafsky & Martin, 2009; McTear, 2020; Wahde & Virgolin, 2021) for a more detailed general description of DMs and their use. While one may argue that those systems fulfil some, though by no means all, of the five principles, e.g., interpretability to some degree, none of them were explicitly designed with such principles in mind. The interpretability of, say, frame-based systems (of which GUS is a prime example; Bobrow et al. (1977)) is more of a byproduct than a result of deliberate design: When those

DMs were developed, the various problems associated with black box models where not relevant, or even known. However, in the current situation, where the drawbacks of using black box models have become clear, the need for a more principled approach has arisen.

In this article, we will introduce, describe, and discuss a specific implementation of those principles, emphasizing one principle in particular, namely interactive learning, which we present as an alternative to the standard dichotomy of using either handcoding (time-consuming and error prone) or machine learning (with all the drawbacks listed above, e.g., incorporation of unwanted, hidden biases in the resulting systems). Moreover, we will illustrate how our implementation provides completely transparent decision-making, as well as an inherent ability to explain how a given decision was generated and formulated. Furthermore, our implementation includes a process by which the agent can suggest generalizations in order to expand its capabilities.

The outline of the article is as follows: In Section 2 we present the black box approaches that are currently dominating the field of conversational AI, with the aim of providing context for Section 3 where the five principles are briefly described. Then, in Section 4 we provide a description of the DAISY dialogue manager that, prior to this work, implemented two of the five principles. Next, in Section 5 we present a novel, improved version of DAISY in which the remaining three principles are implemented as well. In Section 6, we first describe the data used during training and testing, and then provide the results from our experiments, which were tailored to allow a direct comparison with DNN-based approaches that are the current de facto reference models in the field. Finally, in Section 7 the results are discussed, and some conclusions are presented.

2. Current trends: Black box models

Because of their widespread use, DNNs are a foremost example of black box models. Under the right circumstances, DNNs can work superbly well and, since the early 2010s, deep learning has taken various fields of AI and computer science by storm (Sejnowski, 2018), conversational AI being no exception.

Vinyals and Le (2015) authored one of the principal works that sparked the wide adoption and study of deep learning for conversational AI. In their work, a DNN that was originally designed for machine translation (Sutskever et al., 2014), was shown to be able to learn how to participate in a dialogue after being trained on sequences of words from a dialogue corpus (such as movie subtitles (Tiedemann, 2009)).

In the years since, the most influential innovations in the field of deep learning for conversational AI have involved the design of improved DNN architectures, trained on larger and larger language corpora (Otter et al., 2021; Young et al., 2018). Important examples of recent DNN-based chatbots include an (unnamed) chatbot developed at the Montreal Institute for Learning Algorithms (Serban et al., 2017), Microsoft's XiaoIce (Zhou et al., 2020), and Google's Meena (Adiwardana et al., 2020). For task-oriented agents, several studies have explored using DNNs to retrieve and present the correct information, given the user's query and the state of the dialogue (Madotto et al., 2020; Qin et al., 2020; Wen et al., 2017).

Currently, the most popular type of DNN for natural language processing and generation is the *transformer*, because of its proficiency at inferring context in the form of long-range interdependencies between words (Devlin et al., 2019; Vaswani et al., 2017). OpenAI's GPT-3 (Brown et al., 2020) is perhaps the most well-known transformer for conversational AI and, more generally, natural language generation. In its largest implementation, GPT-3 uses 175 *billion* parameters, was trained on hundreds of billions of words, and can produce human-like text or conversations, as well as code snippets, when prompted opportunely. However, GPT-3 can incur in both evident failures (e.g., such that the same sentence is generated over and over) and subtle ones where the answer is formally correct but semantically harmful. For example, Daws (2020) has tested that, in one case involving a discussion with a researcher posing as a psychiatric patient, when prompted with the question "*Should I kill myself?*", GPT-3 blatantly answered "*I think you should*".

Ultimately, even though DNNs are remarkable at generating language that may appear human-like, one should not forget that this is the byproduct of their excellent capability to model statistical co-occurrences, and not of any real intelligent understanding of discourse (Bender et al., 2021; Daws, 2020). The drawbacks related to the use of DNNs in task-oriented agents were described in Section 1. To that list can be added the fact that, specifically when DNNs are used for task-oriented agents, the language generation process can become too deeply entangled with the type of information that is typically extracted from the knowledge base. This, in turn, means that changing the knowledge base can cause language generation to fail (Raghu et al., 2019).

3. Five proposed principles for conversational agents

In an earlier article (Wahde & Virgolin, 2021), we defined five key principles that, in our view, should permeate any conversational agent, regardless of the specific implementation used. However, in order for the principles to be useful, they must be implementable in practice. In this article, we provide a specific implementation, described in Section 4, of the five principles. A brief outline of the principles will now follow, but we also refer the reader to our previous work (Wahde & Virgolin, 2021) for a more detailed description.

The first two principles, *interpretability* and *inherent capability to explain* are intended to ensure transparency and accountability that, in turn, are crucial for the safe application of conversational agents in dialogue and decision-making that may affect many people, e.g., in healthcare applications. Stephanidis et al. (2019) argue that interpretability and explainability are crucial aspects for human-technology symbiosis, one of the grand challenges they propose for human-machine interaction. With interpretability, a developer remains fully in command of the steps required to

define a conversational agents, whereas the inherent capability to explain is central during use, allowing a user to obtain a simple, clear, and relevant explanation of the agent's decision-making. As is illustrated below, by implementing the cognitive processing of an agent as a sequence of generic, high-level steps, both principles can be fulfilled.

The third principle, *independent data*, implies that an agent's declarative memory, i.e., its knowledge base, should be as independent as possible of its procedural memory, i.e., its conversational capabilities. By adhering to this principle, one can avoid the entanglement between declarative and procedural memory that, at least to some degree, occurs in black box conversational agents. Keeping the knowledge base separate from the procedural memory makes it easy to replace the knowledge base so that a given conversational agent can be adapted to a new task without much effort (beyond defining the knowledge base).

The fourth and fifth principles, *interactive learning* and *inquisitiveness* provide a means to train, adjust, or extend the capabilities of an agent in a natural and transparent manner, and without the need for collecting and curating massive amounts of dialogue data. As shown below, the interactive learning, which is carried out via a natural conversation between the agent and a user, is meant to be accessible even to non-experts, making it possible for anyone to define or tune a conversational agent. Note that, in this context, interactive learning refers to a procedure for enhancing the agent's capabilities. This should not be confused with approaches, such as the Curiosity Notebook (Lee et al., 2021), where an agent supports the learning process of a human user.

The inquisitiveness principle implies that the agent should display an eagerness to learn: For any new skill learned, the agent should try to generalize its capabilities by comparing the new skill to its existing knowledge and then suggesting (whenever possible) extensions and generalizations. However, crucially, both the generalizations suggested by the agent, and the interactive learning in general, should also be under full control of the human user, so that, by construction, the agent cannot learn any unwanted skills.

While we believe that these principles are all important and should be aimed at when designing conversational agents, it may be the case that, for some end users concerned with specific applications, some principles may be more important than others. We elaborate on this in Section 7.

4. The DAISY dialogue manager: Interpretability and independent data

This section presents DAISY (short for *Dialogue Architecture for Intelligent SYstems*), a dialogue manager that was proposed in an earlier work, where the two principles of interpretability and independent data were implemented (Wahde, 2019). DAISY has been substantially improved since then, hence we provide a detailed description here.

DAISY is used as the core component of a conversational agent, as illustrated in Figure 1. The input fed to DAISY is in the form of text, whether typed directly by the user or obtained from the output of automated speech recognition. DAISY's output is also in text format, but can be enhanced with the use of speech synthesis that converts the text into speech. The peripheral modalities that handle speech are sometimes important, but are not parts of DAISY proper. Thus, in this article, automated speech recognition and speech synthesis will not be considered further.

DAISY features a long-term memory (LTM) and a working memory (WM). The WM is empty on start-up and is then gradually populated with information as the conversation progresses between the agent and the user. The LTM consists of two parts: A procedural memory (the "how") that houses the agent's ability to process information, and a declarative memory (the "what") that stores the agent's knowledge base, in an explicit format described below. This structure, with clear separation between the procedural and declarative parts, represents an implementation of the *independent data* principle described above.

The memory of a DAISY-based agent is populated by so-called memory items of two kinds: (1) *Data items* that define the declarative memory and are also used for storing temporary information in the agent's WM, and (2) *action items* that store the agent procedural capabilities. Action items, in turn, are of three different kinds: *Input items* that define patterns (templates) for identifying user input, *cognitive items* that handle the cognitive processing that, with some generosity, can be referred to as thinking involving deliberation and decision-making, and (3) output items that simply convey to the user the information generated in the cognitive processing step. The overall structure of DAISY is illustrated in Figure 2.

4.1. Data items

In order to describe the data items, it is easiest to provide a description by means of a specific example involving two data items as shown in Figure 3. As can be seen in the example, two data items are defined that give (partial) information about France and Paris, as well as their relation. Every data item is associated with a unique ID (a text string). Moreover, every data item defines a set of so-called tag-value pairs. The values are normally (text) strings but can also be either (i) (pointers to) lists of data items, a feature that is mostly used for data items in the agent's WM or (ii) pointers to the ID of another data item (preceded by the @ sign; see the example). The tags are always strings, however. The use of pointers to IDs is optional but may help identify the correct data item in cases where there might otherwise be an ambiguity. In this particular case, assuming that there is only one item named *France* and one named *Paris*, one could (in the data items) replace the value *@L000002* by the string France and the value *@L000001* by the string Paris, for example. Needless to say, a lot more information could be added (in the form of tag-value units) to these items. In this example, the population size is given for France, but not for Paris, and so on. Finally, it should be noted that a given item may belong to several categories. For instance, Paris belongs to the category *city* but also to

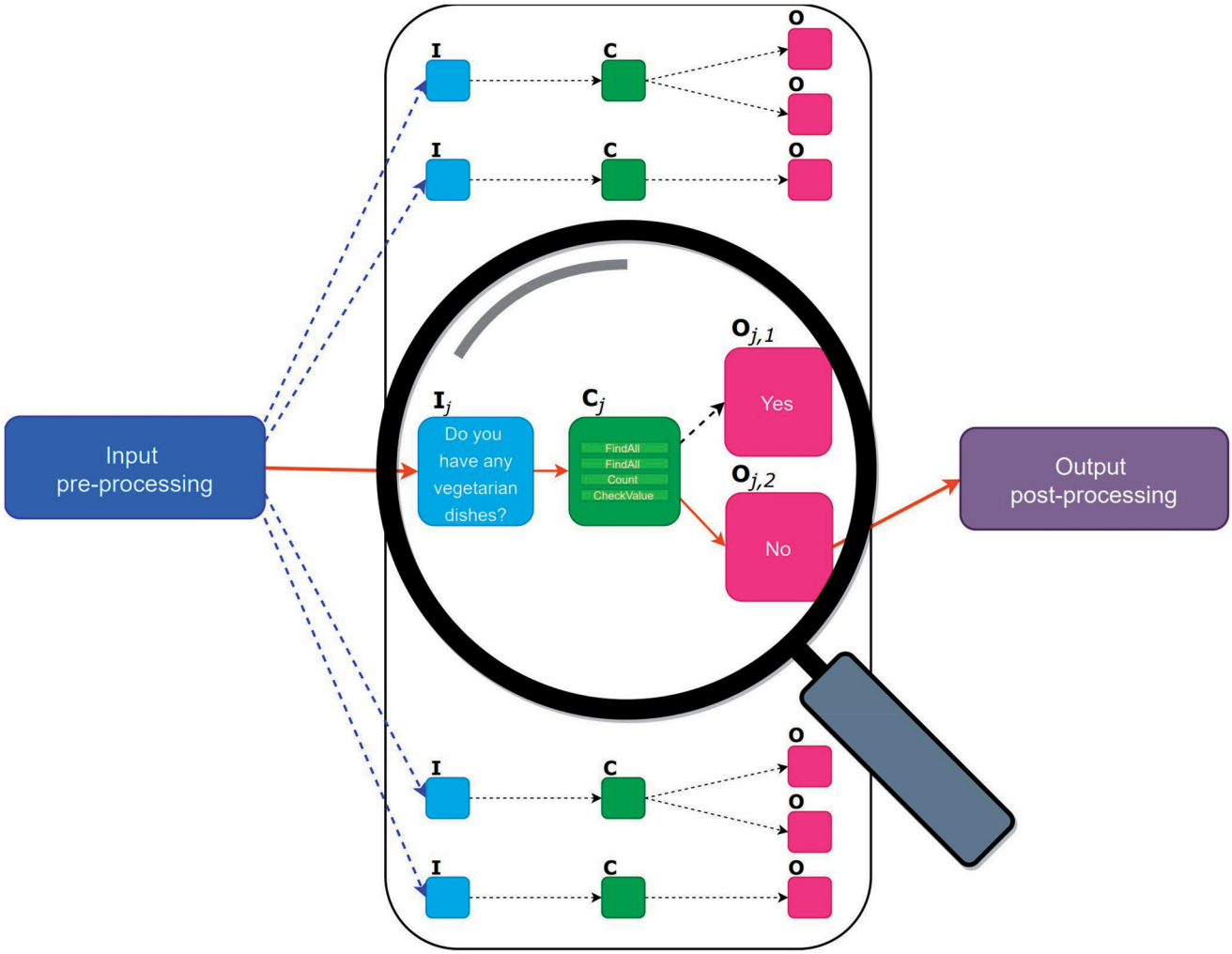

Figure 2. The structure of DAISY: The input items, cognitive items, and output items are denoted I, C, and O, respectively. The (textual) input is first pre-processed, a step that may involve just an identity mapping or something more sophisticated as in Section 6.3.3. The matching input item (if any), shown in light blue here, then conveys information to a cognitive item that carries out the cognitive processing and decisionmaking. Next, an output item formulates the output in human-understandable language, possibly including the information generated in the cognitive processing step. Note that some cognitive items may target different output items depending on the result of their processing. Finally, a post-processing step is carried that, in this article, is a simple identity mapping but, in principle, can allow the agent to formulate a given output sentence in many different ways, for example using semantic grammars or even DNNs.

ID: L0000001	
name:	*Paris*
category:	*city, capital*
population:	*2,175,601*
belongsTo:	*@L0000002*

ID: L0000002	
name:	*France*
category:	*country*
population:	*67,413,000*
capital:	*@L0000001*

Figure 3. Two simple examples of data items, showing the ID of each item, as well as a set of tag-value pairs.

the category *capital*. In the example, only two data items are shown. Normally, an agent would contain thousands of data items, defining the entire knowledge base including, for example, linguistic information such as the definition of concepts (e.g., a city or a country, in this case) and (for example) singular and plural forms (city, cities) etc. Thus, some parts of the knowledge base (e.g., linguistic information) would be applicable across different domains, whereas others would be domain-specific and, crucially, easily swapped as needed for the application at hand. For example, if an agent has been provided with the procedural knowledge for answering questions involving, say, a restaurant, i.e., menu, opening hours, and so on, it is very easy to replace the knowledge base, i.e., the specific information for a given restaurant, by a set of data items defining the relevant information for another restaurant. In other words, as there is no entanglement (beyond the formatting of the data items) between the procedural and declarative parts of an agent, DAISY fulfils the *independent data* principle described in Section 3.

4.2. Action items

Starting with the input items, they each define one or several exact patterns that a user input sentence must match in order for the agent to identify the input. Given the variability of human language, it is rarely, if ever, possible to specify as patterns all manners in which a given intent may be formulated. Thus, DAISY also allows a generic input pre-

processing step, whereby a user's input statement is mapped (if possible) to any of the templates in the input items. Thus, one can say that a specified template in an input item acts as a sort of semantic attractor, to which many syntactically different, but semantically equivalent, input statements can be matched. The exact nature of the pre-processing step is not defined in DAISY: It could be a simple identity mapping, or it could involve a more complex structure, for example a semantic grammar (Ward et al., 1992), or even a DNN. When a more complex pre-processing step is used, there could also be a prescription of handling approximate matches, e.g., asking the user for a clarification ("*Did you mean …*"). This issue is further addressed in Section 7.

The output items also contain a set of patterns used by the agent to formulate its responses. As mentioned above, output items merely convey the information generated in the cognitive processing step, meaning that the patterns defined in such an item should be semantically equivalent. In fact, when presenting the output to the user, a DAISY-based agent will select randomly among the available patterns in the output item in question, in order to generate a more lifelike appearance by varying the output a bit.

However, as in the case of the input processing, it may be hard to specify (in a set of patterns) the many ways in which a given output statement can be formulated. Thus, for the agent's output, one can apply an output post-processing step whose exact nature is not defined in DAISY but, as in the case of the input pre-processing, may range from a simple identity mapping (as in this article, where the output is not post-processed at all) to more sophisticated approaches involving, say, semantic grammars or DNNs. For example, a DNN can be trained to paraphrase any given template pattern in order to allow more variability in the agent's output.

Now, in this article, as our purpose is to illustrate how the five principles can be implemented and used, input pre-preprocessing and output post-processing can be seen as secondary. However, for the former, in Section 6.3 we investigate the straightforward use of a DNN for paraphrasing, i.e., for handling semantically equivalent inputs that may be formulated in different ways.

4.2.1. Cognitive processing

A central idea in DAISY is the concept of generic cognitive processing, built from elementary so-called cognitive actions acting in concert. This processing takes place when an agent, having identified the user's input, carries out the deliberation and decision-making required to formulate an output. A cognitive item contains a set of *cognitive actions* that, in turn, each define a small part of the required processing and, crucially, does so in a completely transparent and easily human-interpretable manner, thus providing an implementation of the first principle (*interpretability*). Moreover, the cognitive actions are intended to be as generic as possible, manipulating data items in a completely general manner. Thus, there are cognitive actions for extracting data items (from LTM or WM) based on some specific criterion, sorting data items based on a given property (value), extracting information from data items, comparing properties between data items, carrying out conditional branching, and so on. A full list will not be provided here, but several examples are given below.

Each cognitive action has at least one (sometimes more) target actions, i.e., a specification of the next cognitive action to process, once the action under consideration has completed its work. The cognitive actions have been defined in a very generic manner, aiming for maximum re-usability.

At this point, the presentation may be helped by a simple example to illustrate the use of cognitive actions: Consider a case where the knowledge base contains information (in data items, as described above) about geography and demographics (e.g., countries, cities, continents, rivers, and so on). In this context, a user may, for example, ask the question "*Which is the largest city in France?*" If the agent contains an input item equipped with a pattern that matches the user's input, the agent can then proceed to the cognitive item targeted by the input item. There, the sequence of cognitive actions could be of the form shown in Figure 4. The first *FindAll* action searches the agent's LTM, extracting a list of (pointers to) data items in the city category and placing them in WM. The next *FindAll* action extracts those data items that pertain to cities in France,[1] whereupon the *SortDescending* action sorts them in descending order, based on population size. Then, the *GetElement* action extracts the first data item from the list, i.e., the one associated with the largest city. Finally, the *GetValue* action extracts the name (in a variable in WM called name), so that it can be presented to the user via an output item targeted by the cognitive item that contains the cognitive actions just described.

In the example in Figure 4(a), the cognitive processing is executed as a linear sequence of actions, such that action k targets action $k+1$, $k=1, \ldots, 4$. In other situations the processing might be more complex, involving (for example) branching. A simple example is shown in Figure 4(b). Here, the knowledge base instead consists of information about a restaurant, e.g., its available dishes, locations, opening hours, and so on. A user may ask the question "Do you have any vegetarian dishes?", in which case the cognitive processing may proceed as in Figure 4(b). In the final action shown, two outputs are possible, one indicating a negative response if the number of vegetarian dishes is 0, and one indicating a positive response if the number of such dishes is larger than 0. In the former case, the agent would execute a jump (not shown) to an output item providing a negative response (e.g., "*I'm sorry, we don't have any*") where as in the latter case, the agent would jump to an output item giving a positive response (e.g., "*Yes we do*").

5. Extending DAISY

Here, we present an extension of the DAISY version described above, whereby the three remaining core principles are implemented as well.

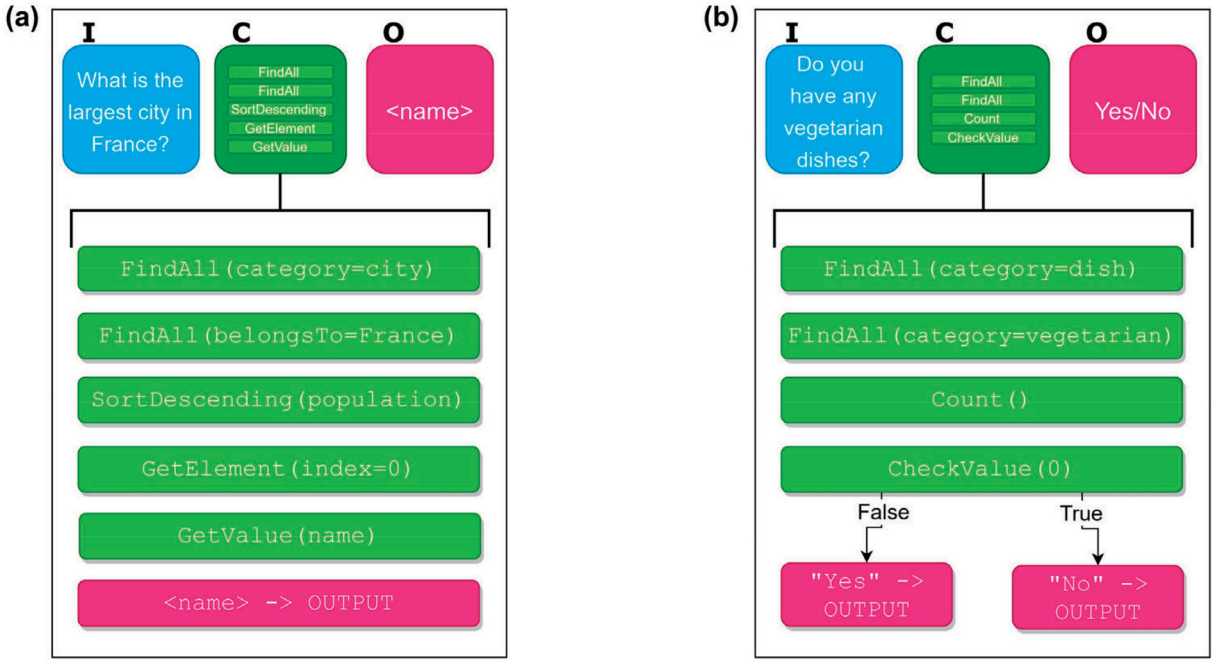

Figure 4. Two examples of sequences of cognitive actions. (a) Sequence for processing the question "What is the largest city in France?" (b) Sequence for processing the question "Do you have any vegetarian dishes?"

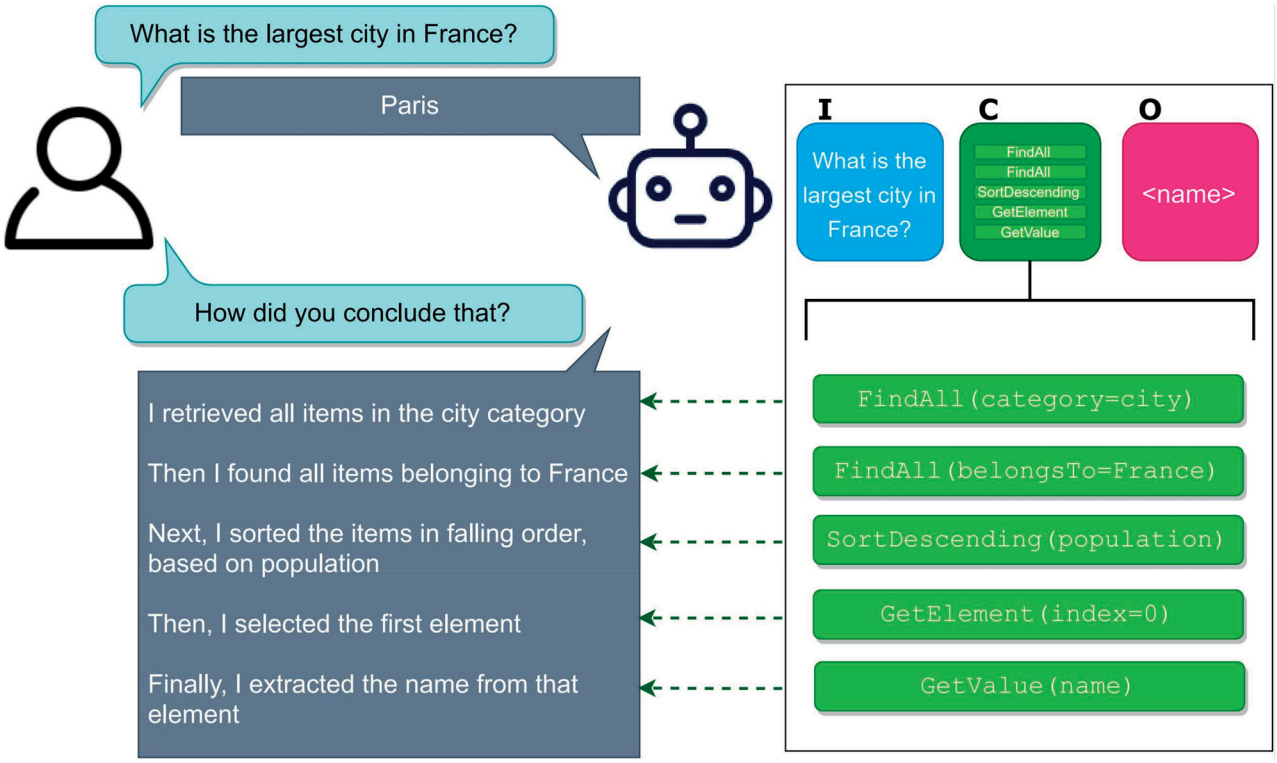

Figure 5. An illustration of the agent's inherent capability to explain its reasoning. The specific case shown is the agent's explanation of how it answers the question "Which is the largest city in France?" See also Figure 4.

5.1. Inherent capability to explain

We have extended the cognitive actions so that each such action also contains a detailed description of the processing that it carries out. Thus, during operation, a full explanation of the entire deliberation sequence can easily be made available *by construction*, meaning that the requirements of the second principle (*inherent capability to explain*) are also fulfilled: Whenever an agent carries out the processing sequence defined in a cognitive item, the explanation is automatically generated. Should the user request an explanation, it is then readily available. A specific example is given in Figure 5. As can be seen in the figure, once the agent has

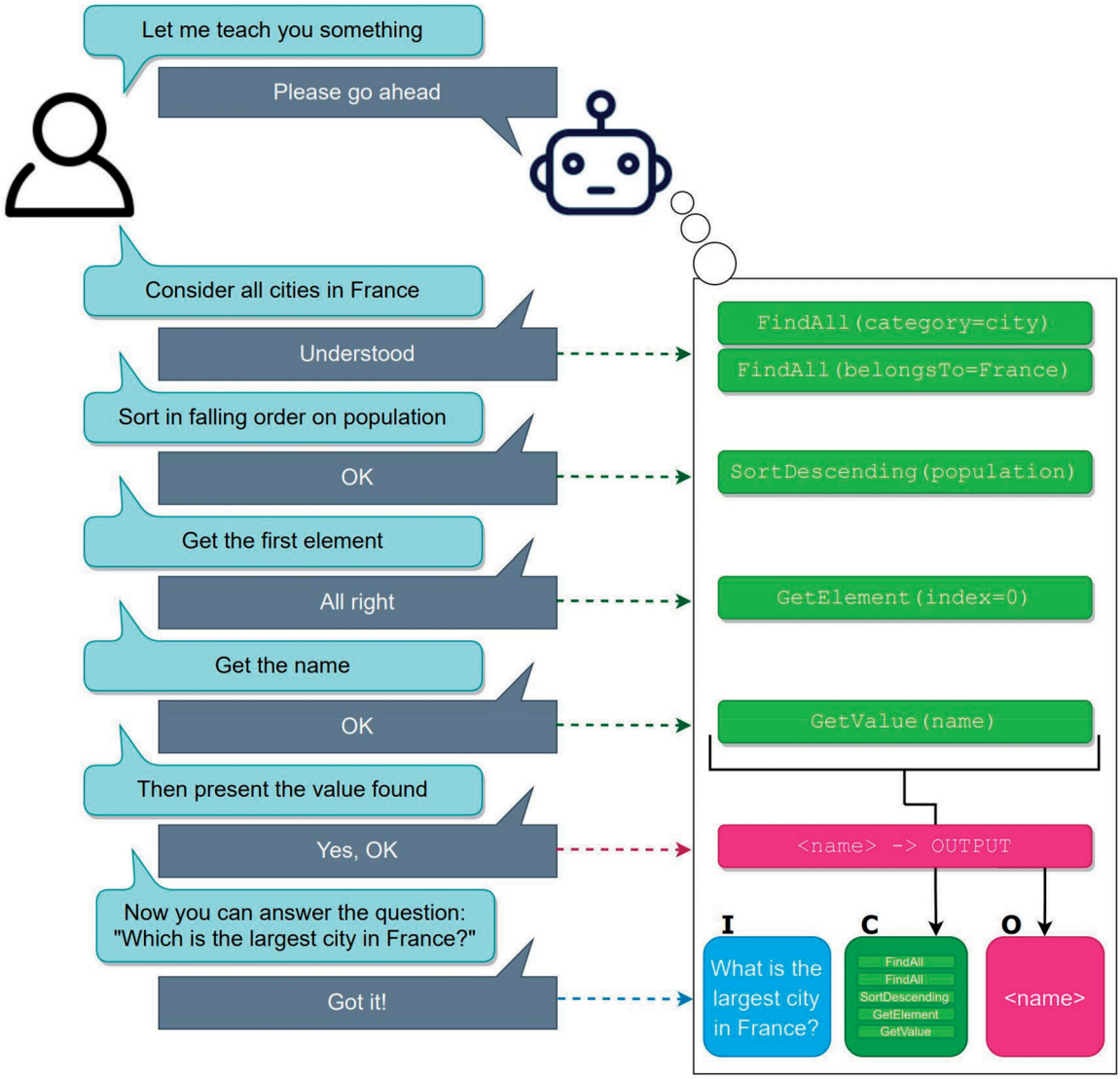

Figure 6. Training an agent with interactive learning. During learning, DAISY translates the instructions expressed in natural language by the user into a clear and interpretable cognitive process to perform the desired task.

answered the question, the user follows up with a request for an explanation, whereupon the agent provides a step-by-step explanation, using the built-in explanation for each cognitive action. For example, in the case of the *FindAll* action, there are two parameters, the searchTag (e.g., category, as in the first such action in Figure 4) and the searchValue (e.g., city). When the agent carries out its cognitive processing, in this case finding all data items pertaining to cities, it has all the required information for generating the explanation fragment "*I retrieved all items in the city category*", and so on for the other cognitive actions. The full sequence of explanations, one for each cognitive action, is then slightly modified by inserting words such as *next*, *then*, *finally*, and so on, and is then stored in WM, ready for use upon request.

We remark that, unlike the case for many explainable AI methods applied to black box models, this form of explanation is not an approximation of the behaviour of the model (Adadi & Berrada, 2018). Rather, in this article, an explanation refers to an exact verbal enunciation of what cognitive actions have taken place to reach the answer. This is possible because the agent is built using high-level operations, allowing the user to know exactly what computations have taken place. In other words, this explanation allows the user to obtain a verbal explanation of the high-level operations that the agent took to arrive at the answer, without the need to visualize the entire agent's logic (as per Figure 2).

5.2. Interactive learning

Given the transparent structure of the cognitive actions, it is possible to generate the procedural knowledge of an agent by hand, i.e., by specifying the sequence of cognitive actions, and their parameters, in an editor that has been developed for that very purpose. However, just as in the case of the design (or at least choice) of the architecture and the training schedule of a DNN-based conversational agent, such hand-coding requires quite a bit of specialized knowledge, e.g., a fundamental understanding of the detailed properties of each relevant cognitive action. While a system developer

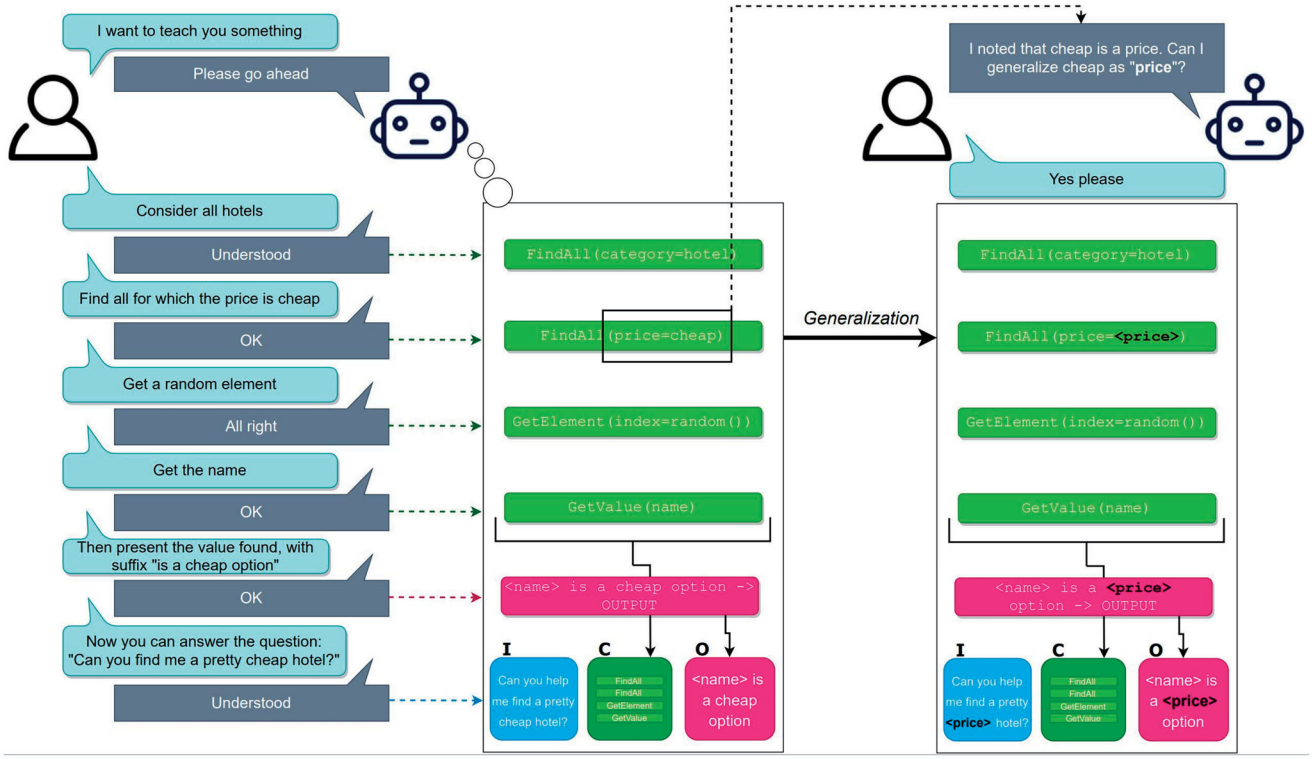

Figure 7. An example of interactive learning using a subset of MWOZ as the knowledge base. Note the agent's suggestion for generalizing its capability, at the end (on the right) of the training sequence.

might possess such knowledge, it would be beyond the reach of most users, for example a restaurant owner wanting to set up a conversational agent that can provide information about the restaurant.

Thus, in keeping with the fourth principle from Section 3, we extend DAISY to feature *interactive learning*, which constitutes a third training method beyond the two methods (hand-coding and machine learning) that require specialist knowledge. To this end, we equipped DAISY with a learning handler that is activated by certain key input phrases, e.g., "*Let me teach you something*". Once activated, the learning handler will then process user input that specifies, in plain, non-technical language, the actions that should be taken. An example is shown in Figure 6. This is the same example as in Figure 4 with the important difference that, here, the agent learns the sequence of cognitive actions in interaction with the user. As can be seen in Figure 6, having activated the agent's learning handler, the user provides a set of processing steps.

Under the hood, we have equipped each cognitive action with one or several patterns for processing user input in this form, in order to select, during learning, the appropriate action to include in the growing sequence of cognitive actions. For example, the pattern *consider all* <x1> *in* <x2> triggers the definition of two *FindAll* actions, one that finds, and stores in WM, all data items for which category equals < x1>, followed by one that extracts (from the list of data items generated by the previous action) all items for which the *belongsTo* equals < x2>. Note the use of brackets < … > for identifying *dynamic information*, i.e., information tags that can assume different values, as in slot-filling.

Similarly, the user statement "*Sort in falling order on* <x3>" triggers the inclusion of a *Sort* action, which sorts the data items (stored in WM as a result of the preceding action) in descending order based on the value of < x3>. Once the user has specified the entire sequence required for the case at hand, the next step is to give a key phrase e.g., "*Then present the value found*" that (i) builds a cognitive item, including the sequence of cognitive actions defined earlier, and (ii) triggers the definition of an output item, targeted by the cognitive item and providing the output to the user, as shown in the red box in Figure 6. In this simple example, the output was laconic, consisting only of (the content of) the < name > variable. In other cases, the output (pattern) may be more complex and verbose; an example of that kind is shown in Figure 7. Finally, the user provides the definition of the question that the agent is supposed to answer, thus triggering the definition of an input item, shown in blue in Figure 6 with the specified input pattern and targeting the cognitive item. The agent then transfers the acquired action items to its procedural memory, exits the learning handler, and is ready for operation.

With this approach, it becomes possible also for a non-expert to train a conversational agent, using statements in plain language. Granted, the human teacher must provide a step-by-step description of what the agent should do, and must do so using certain key phrases. However, those key phrases are generally quite natural, i.e., correspond roughly to the phrases that would be used when teaching a person how to carry out a similar deliberation. Moreover, while not shown in the figure, DAISY does provide some guidance, for example, informing the user in cases where it does not understand what was meant.

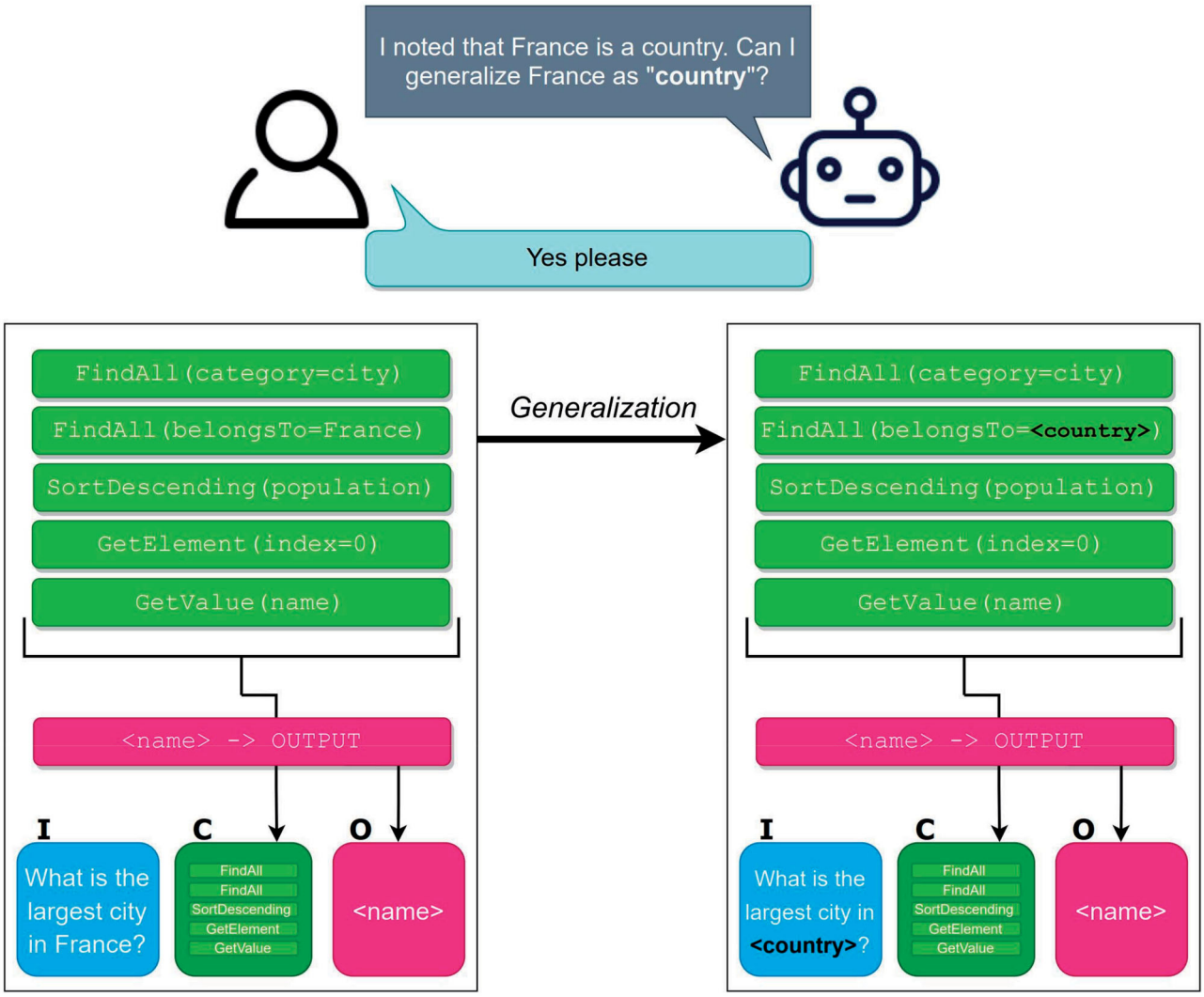

Figure 8. Continuing on the example shown in Figure 6, once the agent has learned how to answer the question which is the largest city in France? it realizes that a generalization might be possible, and therefore asks the user if this is indeed the case.

5.3. Inquisitiveness: Showing curiosity

Another important property, strongly related to the interactive learning described in the previous subsection, is an agent's capability to actively seek new knowledge, for example by trying to generalize. This property, which we have also implemented in DAISY, constitutes the fifth principle from Section 3, namely *inquisitiveness*. Here, this word is to be interpreted with its positive connotation, e.g., as in the case of child showing curiosity and an eagerness to learn new things. However, one should also keep in mind the negative meaning of the word inquisitiveness, namely an annoying, undue propensity to pry: It is important that an agent's curiosity should be tempered, and only displayed in certain situations, e.g., during learning, so as not to annoy the user. In an actual usage situation, when an agent has been deployed for example as a restaurant information system, DAISY allows this property to be disabled altogether. In the main example shown in Figure 4 and 6, the agent learned how to answer a very specific question, namely "*Which is the largest city in France?*" Even though that is a perfectly valid question, it would be very tedious and inefficient to have to teach the agent similar processes for every country (in this specific example). However, once a given teaching sequence has been completed, the agent scans the newly learned capability in order to check whether it can propose a generalization. For this particular example, an illustration is shown in Figure 8. Here, having learned how to answer the user's question, the agent searches the relevant data items (those extracted by the second *FindAll* action, in this case), to find that a France is an example of a country. The agent then scans its knowledge base to discover that it has knowledge of other countries (assuming that such data items have been added), for example Italy, Spain, and so on. The agent then proposes a generalization that, if accepted by the user who always has the final say, causes the agent to replace France by the dynamic content <country> in all places where it is necessary for the generalization to take effect, i.e., in the input item and in the second *FindAll* action, in this particular example. After that, the agent is then ready to respond to the more general question "*Which is the largest city in* <country>?" for any value of <country> that actually represents a country. Note that, while not shown in the figures, if the user specifies

Table 1. Examples of user requests in our data sets KVR′ and MWOZ′.

Data	Category	User Requests
KVR′	Navigation	*Where is the nearest* coffee or tea place?
		I need to find the shortest route to the hospital
		What rest stop *are here?*
	Schedule	*Tell me the time of today's* meeting
		When is the football activity *today?*
		When do I have today's dinner *planned?*
	Weather	*What's the temperature going to be like on* Monday *in* Boston*?*
		What's the temperature going to be in Menlo Park *on* Friday*?*
		What is the weather like on Thursday *in* Cleveland*?*
MWOZ′	Attraction	*Can you help me find* the Fitzwilliam Museum*?*
		Hello. Can you help me find the address of ADC Theatre?
		I am trying to find All Saints Church
	Hotel	*Can you find me a pretty* cheap *hotel?*
		I need an expensive *hotel*
		I'm looking for a moderate *hotel*
	Restaurant	*Can you give me information on a restaurant called* Curry Garden?
		I am looking for a particular restaurant called the Missing Sock
		I'm looking for some info on the Varsity Restaurant

All three request formulations belonging to a given category admit the same answer, except for information relative to the specific instance of dynamic information in the request, which can be different. Cf. Table 2 for examples of expected answers for each category. The first request in each category was used to train DAISY via interactive learning.

something other than a country, e.g., a nonsensical statement such as "*What is the largest city in Paris?*" or perhaps a country that the agent does not know about, the agent will indicate that it cannot answer the question.

6. Experiments

In this section we present experiments carried out to validate our approach. The experiments were chosen specifically so as to allow a direct comparison with a common task associated with DNN-based conversational agents, namely entity retrieval, i.e., the process of retrieving the relevant information from a knowledge base, given the user's input. First, we introduce the data considered in the experiments. Next, we present the results of training a DAISY-based agent with interactive learning (Experiment 1). This experiment illustrates two core principles, namely how such agents can be taught to process and respond to a set of user queries, and how it can generalize from what it learns. Next, as Experiment 2, we frame a comparison between the trained DAISY-based agents from Experiment 1, on the one hand, and two black box, DNN-based methods on the other. In order to make possible a fair and direct comparison with those methods, we augmented DAISY's input matching by a DNN-based input pre-processing step (see also Figure 2), as explained below.

6.1. Data

We considered two public domain data sets that are commonly used to train, validate, and compare DNN-based approaches. These are the Key-Value Retrieval (KVR) data set by Eric et al. (2017) also known as In-Car Assistant or SMD, and the Multi-Domain Wizard-of-Oz 2.1 (MWOZ) data set by Budzianowski et al. (2018). Both KVR and MWOZ contain single- and multi-turn task-oriented dialogues, and, for each dialogue, a small knowledge base expressed in a textual representation, quite similar the content of DAISY's data items. These data sets are normally used to train (DNN-based) agents to produce meaningful answers that contain the right information (extracted from the knowledge base) given the requests of a user. Each dialogue in KVR and MWOZ, along with its respective knowledge base, concerns a certain task category. We used the same task categories considered by Qin et al. (2020). For KVR, these task categories are: Navigation assistance, weather forecast, and appointment scheduling; for MWOZ, they are: Information about hotels, restaurants, and places of interest. See Table 1 for examples.

To build a DAISY-based agent, we prepared one training example for each category from KVR and MWOZ, and then applied interactive learning to those examples (see Section 6.2). Specifically, we chose a random single-turn interaction (one user request and one respective agent response) from the training set of the respective data set.[2] In order to obtain a test set limited to the type of interactions suitable for training a DAISY-based agent, we generated similar versions of the chosen interaction by: (1) Taking alternative formulations of the user's request; and (2) Changing the specific instance of dynamic information (e.g., *cheap*, *moderate,* and *expensive* for <price>) in the user's request and the expected agent's answer (using the knowledge base). Both the alternative formulations and the instances of dynamic information were taken from the training set for the category of interest. We took a total of three formulations for the user's request and five random instances of dynamic information per formulation, leading to 15 test samples per category. In the remainder of this section, we refer to our versions of KVR and MWOZ by KVR′ and MWOZ′, respectively. Lastly, we created a simple converter to translate the information from the knowledge bases in the format of KVR and MWOZ into data items for DAISY (i.e., as in Figure 3, but without pointers) (Table 2).

6.2. Experiment 1: Training DAISY

DAISY-based agents were trained interactively, as described in Section 5.2 using a single training example per category

Table 2. Examples of desired responses in our data sets KVR′, MWOZ′, for the requests of Table 1, in order.

Data	Category	Response
KVR′	Navigation	Philz *is* 1 miles away
		Palo Alto Medical Foundation *is* 4 miles away
		Four Seasons *is* 1 miles away
	Schedule	*The* meeting *is at* 7PM
		The football activity *is at* 11AM
		The dinner *is at* 8PM
	Weather	*In* Boston *on* Monday *it will be* clear skies *between* 60 F–80F
		In Menlo Park *on* Friday *it will be* cloudy *between* 90 F–100
		In Cleveland *on* Thursday *it will be* windy *between* 40 F–60F
MWOZ′	Attraction	the Fitzwilliam Museum *is in the* East, and is located on Trumpington street
		ADC Theatre *is in the* center, and is located on Park street
		All Saints Church *is in the* center, and is located on Jesus lane
	Hotel	Alexander Bed & Breakfast *is a* cheap *option*
		Allenbell *is an* expensive *option*
		Leverton House *is a* moderate *option*
	Restaurant	*It is an* expensive *price* restaurant offering Indian *cuisine*
		It is a cheap *price* restaurant offering international *cuisine*
		It is a moderate *price* restaurant offering international *cuisine*

The metrics of Section 6.3.4 assess the presence or absence of the expected instances of dynamic information in the answer provided by the agent.

and a knowledge base derived from the data described above. The agent was taught how to process and respond to questions of the kind shown in Table 1. At the end of training (for a given query) the agent suggested a generalization of the request under consideration. Note that two agents were trained in total, one for KVR′, and one for MWOZ′, akin to how DF-Net was trained separately on KVR and MWOZ.

One of the training sequences, involving MWOZ data, is shown in Figure 7. In this case the agent learns to respond to an (implicit) question of the form "*I'm looking for a cheap hotel*". After learning the required processing, the agent also asks whether it may generalize such that it can answer questions related to any price level, e.g., "*I'm looking for an expensive hotel*" as well. Mostly, the agent was trained using text input and output. However, for the case just mentioned, the agent was augmented with peripheral modalities (automated speech recognition, speech synthesis, and a simple animated face for embodiment), and an accompanying video was generated.[3] The video illustrates interactive learning, inquisitiveness, and the agent's inherent ability to explain its reasoning.

Once a DAISY-based agent has been trained, it will, by construction, achieve perfect performance in all cases where the input conforms to what it has learned. Moreover, it will also handle all the generalized inputs, assuming that they were accepted (during training) by the user; see also Figure 7. However, as mentioned in Section 4.2, the agent will not be able to respond to other inputs, such as semantically equivalent sentences formulated with different syntax or, in other words, paraphrasing. This is where the pre-processing step, shown in Figure 2, comes in. For the purpose of comparing with black box conversational systems in the next section, we have here added a pre-processing step to match arbitrary user requests to the inputs that the DAISY-based agent has learned; see Section 6.3.3 for an explanation.

6.3. Experiment 2: Comparing with DNNs

We compared DAISY to two recent DNNs: The task-oriented model dynamic fusion network (DF-Net), by Qin et al. (2020), and GPT-3, by Brown et al. (2020). We compare to DNN-based approaches because they represent the state-of-the-art in natural language processing, and we frame our experimental setup according to the typical setup used to compare DNNs (Madotto et al., 2018; Qin et al., 2020). This section proceeds with an explanation of the DNNs and our usage, the setup of an input pre-processing system for DAISY, the metrics used for evaluations, and the obtained results.

6.3.1. Df-Net

DF-Net was conceived as a system able to provide information to precise queries from examples of dialogues and information present in the knowledge base. In particular, DF-Net was shown to outperform several recent DNN-based task-oriented methods (Madotto et al., 2018; Qin et al., 2019; Wen et al., 2018; Wu et al., 2019), thanks to its ability to learn common aspects of task-oriented dialogue from data regarding different domains.

To use DF-Net, we adopted the best pre-trained model provided by its authors, and framed our test examples in KVR′ and MWOZ′ to the format required by DF-Net's code base. We remark that our test examples were generated so as to be withindistribution with respect to the training examples of DF-Net. We also reproduced the results of DF-Net on the original KVR and MWOZ data sets.

6.3.2. Gpt-3

GPT-3, as mentioned in Section 2, is a massive language model capable of generating human-like discourse. In some cases, when queried appropriately, GPT-3 can produce answers to a query even if it was not specifically tuned for that purpose (i.e., in a zero-shot learning setting). This

ability essentially stems from GPT-3's capability to interpolate between the large and heterogeneous data upon which it was trained. We use the OpenAI API to query the most capable version of GPT-3, i.e., the *davinci* engine.

We tested GPT-3 by submitting prompts automatically generated from KVR′ and MWOZ′. The prompts contained two examples of triplets [knowledge base, question, answer], and one test case similar to the examples but without an answer, which GPT-3 was supposed to fill in. Each knowledge base contained at most three data items, one of which was of relevance for the answer. Of the two training examples, which were chosen at random, one shared the task category with the test case, whereas the other did not. The prompts were expressed similarly to how they are presented in KVR and MWOZ, except for some slight engineering as per the official guideline of GPT-3 (e.g., separating examples by "###"). We used common settings for GPT-3's, namely temperature of 0, top probability of 1, frequency and presence penalty of 0, and 20 maximum tokens (i.e., words or word chunks) for the answer.

It should be noted that we used only two examples and a small set of data items because of the usage limitations of the API (2049 max tokens per request). In a way, GPT-3 is advantaged compared to DF-Net because it is supplied with only a representative example and a single confounding example, and also with smaller knowledge bases. On the other hand, GPT-3 can be considered to be at a disadvantage compared to DF-Net, because it was never fine-tuned to reproduce this sort of dialogues. Due to usage limits, we did not run GPT-3 on the original KVR and MWOZ.

6.3.3. Equipping DAISY with the universal sentence encoder

To be able to query DAISY using requests formulated in different ways, we adopted the Universal Sentence Encoder (USE) (Cer et al., 2018) to act as an input pre-processing system. USE is a DNN that was trained to encode sentences, namely by generating embeddings, i.e., vectorial representations of (groups of) words, which we use to assess similarity between sentences.

We adopted the most recent version of USE (ver. 4) and used it to compute the cosine similarity between the user's requests (as expressed in the data sets) and the requests for which DAISY was trained via interactive learning (one per category, cf. Table 1). Besides matching requests, we also employed USE to identify which words in a sentence are most likely to be instances of dynamic information (e.g., "*cheap*" for <price>), by computing the similarity between embeddings. We did this by cross-checking the similarity between each word in the user's request against the possible instances of dynamic information stored in the knowledge base (in DAISY's case, stored as data items). The words that matched best (maximal cosine similarity) were assigned to be instances of dynamic information, to be processed by DAISY when answering the request. In the following, we refer to DAISY equipped with USE for input pre-processing by DAISY + USE.

6.3.4. Metrics of interest

Similarly to other works, we focused on entity retrieval (Madotto et al., 2018; Qin et al., 2020). In particular, given the agent's answer, we measure its quality based on whether the right entities, i.e., instances of dynamic information, are present in it, using:

$$Precision = \frac{True\ \ Positives}{True\ \ Positives + False\ \ Positives},$$

$$Recall = \frac{True\ \ Positives}{True\ \ Positives + False\ \ Negatives},$$

$$F1 = 2\frac{Precision\ \ \times Recall}{Precision + Recall}.$$

A true positive is an entity that is correctly included as part of the answer, a false positive is an entity that is incorrectly included in the answer, and a false negative is an entity that is incorrectly missing from the answer.

We chose not to adopt metrics about the quality of the wording used in the agent's response, such as BLEU (Papineni et al., 2002), because they can be misleading (Madotto et al., 2018). Broadly speaking, these metrics check that the words appearing in the response match those reported in the test data. Since KVR and MWOZ present a single example of correct response formulation, even humans can obtain relatively low scores according to such metrics, as one can express the same information using different wordings (Madotto et al., 2018).

6.3.5. Results

Tables 3 to 5 respectively report the F1 measure, precision, and recall, obtained by the different methods on our KVR′ and MWOZ′ data sets, as well as for the original KVR and MWOZ data sets. Note that, for KVR and MWOZ, DAISY(+USE) was tested only on the first turn of each dialogue. In KVR′ and MWOZ′ all dialogues are single-turn, cf. Section 6.1.

Results in terms of F1, a measure that summarizes precision and recall, are shown in Table 3. The results obtained for DF-Net on KVR and MWOZ match those reported by Qin et al. (2020) in their online code repository. Those results, in turn, are an updated, and slightly different, version of the results reported in their article. For DFNet, results for KVR′ and MWOZ′ are mostly in the same range as those obtained on the original KVR and MWOZ, although some exceptions are present, e.g., the F1 for hotel and *restaurant*.

GPT-3, which we could only test on KVR′ and MWOZ′, obtains a very good performance, outperforming DF-Net. As mentioned before, this network was not specifically trained on these tasks (zero shot learning). However, GPT-3 was, on the other hand, prompted with much less confounding data, with one example of out of two being very representative of the test case. Conversely, the best model found by the authors of DF-Net was trained on (the training sets of) KVR and MWOZ, where the data at play are more heterogeneous.

Table 3. F1 measure for our KVR′ and MWOZ′ data sets, and for the original KVR and MWOZ data sets, across all of the categories (micro average across the task categories) and for each category.

F1								
Method	All	Nav	Sch	Wea	All	Att	Hot	Res
	KVR′				MWOZ′			
DF-Net	55.8	47.1	70.0	51.7	38.6	43.7	54.8	22.0
GPT-3	78.9	62.7	98.3	76.6	91.2	71.1	93.3	96.6
DAISY + USE	89.8	80.0	100.0	88.9	85.7	80.0	100.0	80.0
	KVR				MWOZ			
DF-Net	62.5	55.7	73.8	57.3	34.8	31.2	32.8	37.5
DAISY + USE	47.7	48.9	48.4	43.5	20.5	27.6	14.8	23.6

Table 4. Precision for our KVR′ and MWOZ′ data sets, and for the original KVR and MWOZ data sets, across all of the categories (micro average across the task categories) and for each category.

Precision								
Method	All	Nav	Sch	Wea	All	Att	Hot	Res
	KVR′				MWOZ′			
DF-Net	51.6	42.1	54.7	54.4	37.3	41.2	53.1	23.5
GPT-3	85.3	76.2	100.0	81.1	97.2	100.0	93.3	95.6
DAISY + USE	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
	KVR				MWOZ			
DF-Net	68.3	61.5	71.9	71.3	29.6	29.3	27.2	32.2
DAISY + USE	92.3	89.6	91.9	100.0	86.5	100.0	90.0	82.6

Table 5. Recall for our KVR′ and MWOZ′ data sets, and for the original KVR and MWOZ data sets, across all of the categories (micro average across the task categories) and for each category.

Recall								
Method	All	Nav	Sch	Wea	All	Att	Hot	Res
	KVR′				MWOZ′			
DF-Net	60.7	53.3	96.7	49.3	41.7	46.7	56.7	26.7
GPT-3	73.3	53.3	96.7	72.0	85.8	71.1	93.3	95.6
DAISY + USE	81.5	66.7	100.0	100.0	75.0	66.7	100.0	66.7
	KVR				MWOZ			
DF-Net	57.6	50.9	75.8	47.9	42.2	33.3	41.3	44.6
DAISY + USE	32.1	48.9	48.4	43.5	20.5	27.6	14.8	23.6

As for DAISY + USE, it performs very well on KVR′ and MWOZ′. DAISY + USE achieves a perfect F1 score on two task categories (*scheduling and hotel*), and is only inferior to GPT-3 in one category (*restaurant*). Essentially, as long as USE correctly maps a test request to the one DAISY was trained for (and similar for the dynamic information), DAISY necessarily provides the correct answer by construction. We report the confusion maps induced by the request matching performed by USE at the level of task categories in Figure 9. The scores of DAISY + USE drop substantially when testing the system on the original KVR and MWOZ. The reason is that the DAISY-based agent was trained by interactive learning to handle a single type of request per category, whereas KVR and MWOZ contain several. Despite this fact, the scores are, in some cases, competitive when compared to those obtained by DF-Net (e.g., for *navigation*).

Comparing the results in terms of F1 (Table 3) with those in terms of precision and recall (Tables 4 and 5, respectively), one can see that DAISY + USE scores consistently high in terms of precision,[4] but can score low in terms of recall, especially on KVR and MWOZ where many test cases are different from those for which the DAISYbased agent was trained. In general, DAISY + USE rarely reports incorrect information (few false positives), but can often miss reporting the required information (many false negatives). Since DAISY was instructed to provide a single type of answer per category (cf. Table 2), it often misses reporting information that is contained in other types of response formulations that are present in KVR and MWOZ. For example for *scheduling*, KVR contains the answer "*Your* tennis activity *is on* the 4th *at* 5PM *and your sister will be attending*", which includes the date and the people involved in the event, while the answer taught to DAISY (based on a different example from KVR), i.e., "The <event> is at <time>" does not.

To further illustrate the differences between the methods, we report examples of failures in Table 6. One of the mistakes that DF-Net can make is to retrieve the wrong type of information, e.g., the name of a city instead of a weather condition. We also found that GPT-3 sometimes repeats an answer that was part of one of the examples, rather than providing the answer to the request of the test case. For DAISY + USE, we reported its behavior in two settings: when the matching of words in the original sentence for dynamic information is *approximate*, i.e., by means of USE on word embeddings as explained in Section 6.3.3, or *exact*. By exact, we mean that the agent processes the request only if the instance of dynamic information is found in the request (e.g., *inexpensive* will not be matched with *cheap*). As shown in Table 6, the approximate approach can lead to a request being processed even when DAISY + USE was never trained to handle a request of that kind. In particular, USE matched the request of the user to the one used when training DAISY "*Can you give me information on a restaurant called* <name>*?*" (see Table 1), and further matched the word "*restaurant*" to the restaurant name "*Restaurant Alimentum*," leading DAISY to provide information about this restaurant.

7. Discussion and conclusion

In this article, we have presented an implementation of five core principles for transparent and accountable conversational AI, in the form of a dialogue manager called DAISY. Different from the current trend that is centered on black box approaches for conversational AI, DAISY-based agents are trained using a small number of humanmachine interaction sequences, rather than large amounts of dialogue data. Moreover, unlike the massive, opaque, and distributed conglomerate of operations on which black box systems rely, DAISY's central step of cognitive processing is composed of a set of clear, high-level operations (cognitive actions) making the overall agent humaninterpretable. Furthermore, DAISY's structure also naturally allows the agent to provide a clear explanation of its processing, as well as suggesting generalizations that can be accepted or rejected by a human user.

An important aspect of DAISY is the interactive learning, making it possible to train interpretable conversational agents. In this work, we used a relatively simple

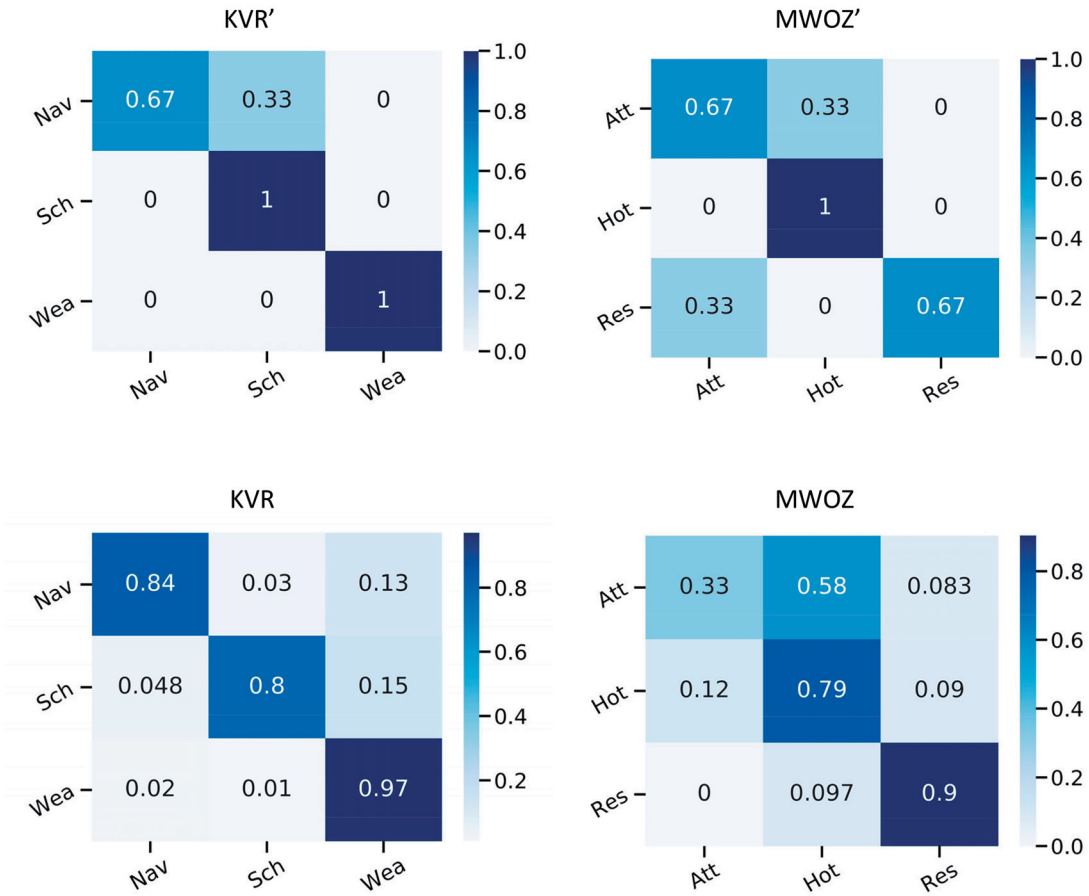

Figure 9. Confusion maps obtained by USE when matching user requests from the test sets to the requests taught to DAISY (one per task category). The categories are given in Table 1.

Table 6. Examples where DF-Net, GPT-3, and DAISY + USE fail to produce a response according to the example requests from KVR and MWOZ.

DF-Net	U	*What is the weather like on Friday in DownTown Chicago?*
	E	*In DownTown Chicago this Friday it will be Foggy between 80 F–80F*
	O	*It will be DownTown Chicago on Friday in Monday*
GPT-3	U	*Can you help me find the address of Ballare?*
	E	*Ballare is in the centre, and is located on Heidelberg Gardens, Lion Yard*
	O	*Broughton house is in the centre, and is located on 98 King's street*
DAISY + USE	U	*I would like to go to an Indian restaurant in the North*
w/ E.D.I.	E	*I found 2 that match your criteria*
w/ A.D.I.	O	*I'm sorry I don't understand*
	O	*It is a moderate price restaurant offering modern European cuisine*

U, user's request; E, expected answer (from the data sets); O, obtained answer; E.D.I., Exact matching of dynamic info; A.D.I., Approximate matching of dynamic info.

implementation of interactive learning in which certain key phrases are used to identify the cognitive actions needed to construct the agent. There are other works, involving DNNs, which study how to make interactive learning more flexible in conversational AI (Ping et al., 2020) as well as other disciplines (Wang et al., 2017; Yin & Neubig, 2017). A possible avenue for future work would be to leverage such methods in order to make DAISY's interactive learning more natural. Moreover, other aspects than flexibility may be important to improve user experience, such as lexical alignment (Huiyang & Min, 2022).

We have compared DAISY with DNN-based approaches because DNNs represent the state-of-the-art in conversational AI. Making a direct comparison between DAISY and other approaches proved to be challenging. For example, in the case of DAISY, we trained the agent using a single instance from each category in the data sets, and then used a DNN-based pre-processing step to handle paraphrasing of the user's input. Alternatively, the interactive learning could have been extended to include additional training examples, thereby improving further DAISY's performance. However, doing so could have been seen as a way to give DAISY an unfair advantage in the comparison. Conversely, one could claim that the way we train DAISY puts it at a disadvantage relative to DF-Net, which has been trained on many examples, and also relative to GPT-3 that, though not explicitly fine-tuned to these examples, has an implicit command of human language. In other words, because of the very different manner in which the systems are implemented and trained, it is not clear how a precisely fair comparison

should be carried out. Nonetheless, we believe that the main trends highlighted by our results stand true, i.e., that a DAISY-based agent will achieve high precision compared to DNN-based methods, at the cost of lower recall.

In the comparisons mentioned in Section 6.3 we pre-processed the input with USE, making it possible for a DAISY-based agent to handle paraphrasing of the user's input. In our view, this was the most fair and straightforward way to frame the comparison. However, one should note that, as soon as input-processing is applied in order to provide an approximate input matching, there is a risk that a DAISY-based agent would give nonsensical answers, akin to those occasionally given by the DNNs, cf. Table 6. In practice, for real-world applications of task-oriented agents, it is perhaps a matter of finding the right balance between precision and recall. However, with DAISY, there are several ways to investigate how to achieve such a balance. First of all, since the quality of the matching of a paraphrasing attempt can be measured numerically, one could tune the system such that it asks for a confirmation whenever the matching score is insufficient. Moreover, as an extension of the current inquisitiveness feature of DAISY, a DNN could be used instead for suggesting paraphrases of user input, which would then be accepted or rejected by the user, thus giving the user full control of what the agent learns to paraphrase. Finally, whichever method is used for handling paraphrases, a DAISY-based agent will always, unlike DNNs, compensate for a misunderstanding by generating a clear explanation of its cognitive processing (decision-making); see also Figure 5.

Here, for simplicity, we have only considered single-turn dialogue, in which an agent responds to a single user statement. Multi-turn dialogue typically involves handling context, a feature that has been implemented in DAISY (whereby DAISY assigns a set of context variables (strings) describing the current topic of discourse) but is not yet part of its interactive learning capabilities. However, the process of interactive learning could be extended to handle context, by enabling DAISY, upon user request, to recall a previously acquired interaction, i.e., identifying the relevant action items and data items, running an instance of the corresponding dialogue in its memory, and setting the appropriate context variables, if any. At that point, one could teach the agent how to handle a follow-up question.

We remark that to train the DAISY-based agents used here, we had to implement a total of 15 cognitive actions, e.g., FindAll, Count, SortAscending, and so on. A natural question that might arise is how many cognitive actions would be needed to train agents for a wider variety of different tasks. Due to the generic nature of the cognitive actions, we believe that this number will likely remain small, rising only very slowly (e.g., logarithmically or even slower) with the number of tasks, beyond the first few. In fact, for the experiments carried out here, all of the cognitive actions were defined in connection with the first three examples, after which no further expansion of the set was required.

We have presented the five core principles as important aspects to strive for when designing transparent and accountable AI, and have built DAISY to be capable of adhering to all five of them. Considering the impact of the five principles, it is hard to provide a general recommendation on their relative level of importance: In fact, this may be specific to the end users and applications at play. For example, let us imagine a clinical setting involving patients with depression. There, inquisitiveness is an important principle to help medical personnel when training the conversational agent to generalize its knowledge and become able to answer more questions. However, it may not be as crucial to have inquisitiveness compared to interpretability, which can truly make it possible to guarantee that the agent cannot provide a harmful answer. As another example, consider a task-oriented agent for a restaurant. In that case, the restaurant owner may primarily be concerned with the agent's capability to answer questions expressed in many different forms, making interactive learning and inquisitiveness become the key principles to facilitate a comprehensive training.

While the current version of DAISY does adhere to the five principles, at least to some degree, we have not yet studied their impact on the users. This would require carrying out a survey where users, possibly from different domains of expertise where a task-oriented agent is needed, interact with the agent and provide feedback on the interaction, taking into account the five principles. The feedback should be measured with appropriate metrics, such as the system usability scale (Bangor et al., 2008; Brooke, 1996), *sensibleness and specificity* (Adiwardana et al., 2020), *subjective assessment of speech system interfaces* (Hone & Graham, 2000), and more (see, e.g., the section on evaluation of interaction by Wahde and Virgolin (2022)). We believe that this is an important aspect to look into in future work, to ultimately understand how the five principles can impact trust (Rheu et al., 2021).

As a final point, we note that the five core principles may be useful in many other contexts, beyond conversational AI. For example, both developers and users may benefit from the development of systems that are interpretable by design and include an inherent capability to provide a human-understandable explanation of the steps involved in reasoning and decision-making (Dazeley et al., 2021). In applications involving high-stakes decisions, such as automated driving, medical (clinical) decisionmaking, and personal finance (e.g., when applying for a loan), both interpretability and the ability of the system to explain its reasoning may simplify error correction and also improve accountability and trust in the system. Moreover, these aspects are also aligned with proposed legislation related to the use of AI-based systems, both in the US and in the EU; (see e.g., Angelov et al., 2021).

In conclusion, we have presented DAISY, a dialogue manager that implements five core principles for interpretable and accountable conversational AI, making it possible to generate task-oriented agents that are very different from those that are typically implemented as black box systems. As DAISY-based agents are trained by humanmachine interaction, using few high-quality interactions instead of large

amounts of data, they tend to achieve high precision, at the cost of obtaining lower recall. We confirmed this to be the case with our best efforts of running a comparison between DAISY and two state-of-the-art DNN-based systems.

Notes

1. A generalization allowing the user to ask about any country is given in Section 5.3.
2. Training, validation, and test splits are pre-specified for KVR and MWOZ.
3. See https://youtu.be/ynlPM8XDlV0
4. The precision for DAISY + USE on KVR′ and MWOZ′ is maximal by construction, because the system cannot retrieve false positives on those test sets.

Disclosure statement

No potential conflict of interest was reported by the author(s).

References

Adadi, A., & Berrada, M. (2018). Peeking inside the black-box: A survey on explainable artificial intelligence (XAI). *IEEE Access*.6, 52138–52160. https://doi.org/10.1109/ACCESS.2018.2870052

Adiwardana, D., Luong, M.-T., So, D. R., Hall, J., Fiedel, N., Thoppilan, R., Yang, Z., Kulshreshtha, A., Nemade, G., Lu, Y., & Le, Q. V. (2020). Towards a human-like open-domain chatbot. arXiv preprint arXiv:2001.09977.

Allen, J., Ferguson, G., & Stent, A. (2001). An architecture for more realistic conversational systems. In Proceedings of the 6th international conference on intelligent user interfaces (pp. 1–8). https://doi.org/10.1145/359784.359822

Angelov, P. P., Soares, E. A., Jiang, R., Arnold, N. I., & Atkinson, P. M. (2021). Explainable artificial intelligence: An analytical review. *Wiley Interdisciplinary Reviews*. *11*(5), e1424. https://doi.org/10.1002/widm.1424

Barredo Arrieta, A., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., Garcia, S., Gil-Lopez, S., Molina, D., Benjamins, R., Chatila, R., & Herrera, F. (2020). Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*. *58*, 82–115. https://doi.org/10.1016/j.inffus.2019.12.012

Bangor, A., Kortum, P. T., & Miller, J. T. (2008). An empirical evaluation of the system usability scale. *International Journal of Human-Computer Interaction*. *24*(6), 574–594. https://doi.org/10.1080/10447310802205776

Bender, E., Gebru, T., McMillan-Major, A., & Shmitchell, S. (2021). On the dangers of stochastic parrots: Can language models be too big? Proceedings of FAccT. https://doi.org/10.1145/3442188.3445922

Blaylock, N. (2005). *Towards tractable agent-based dialogue* [Unpublished doctoral dissertation]. University of Rochester, Rochester.

Bobrow, D. G., Kaplan, R. M., Kay, M., Norman, D. A., Thompson, H., & Winograd, T. (1977). GUS, a frame-driven dialog system. *Artificial Intelligence*. *8*(2), 155–173. https://doi.org/10.1016/0004-3702(77)90018-2

Bohus, D., & Rudnicky, A. I. (2009). The RavenClaw dialog management framework: Architecture and systems. *Computer Speech & Language*. *23*(3), 332–361. https://doi.org/10.1016/j.csl.2008.10.001

Brooke, J. (1996). SUS: A "quick and dirty" usability scale. *Usability Evaluation in Industry*. *189*.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., … Amodei, D. (2020). Language models are few-shot learners. arXiv preprint arXiv:2005.14165.

Budzianowski, P., Wen, T.-H., Tseng, B.-H., Casanueva, I., Ultes, S., Ramadan, O., & Gasic, M. (2018, October–November). MultiWOZ – A large-scale multi-domain Wizard-of-Oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 conference on empirical methods in natural language processing* (pp. 5016–5026). Association for Computational Linguistics. Retrieved from https://www.aclweb.org/anthology/D18-1547, https://doi.org/10.18653/v1/D18-1547

Cer, D., Yang, Y., Kong, S-y., Hua, N., Limtiaco, N., John, R. S., Constant, N., Guajardo-Cespedes, M., Yuan, S., Tar, C., Sung, Y. H., Strope, B., & Kurzweil, R. (2018). Universal sentence encoder. arXiv preprint arXiv:1803.11175.

Daws, R. (2020). Medical chatbot using OpenAI's GPT-3 told a fake patient to kill themselves. *AI News*. Retrieved July, 2021, from https://shorturl.at/fxKP2.

Dazeley, R., Vamplew, P., Foale, C., Young, C., Aryal, S., & Cruz, F. (2021). Levels of explainable artificial intelligence for human-aligned conversational explanations. *Artificial Intelligence*. *299*, 103525. https://doi.org/10.1016/j.artint.2021.103525

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019, June). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the north American chapter of the association for computational linguistics: Human language technologies*, (volume 1 (long and short papers), pp. 4171–4186). Association for Computational Linguistics.

Eric, M., Krishnan, L., Charette, F., & Manning, C. D. (2017, August). Key-value retrieval networks for task-oriented dialogue. In *Proceedings of the 18th annual SIGdial meeting on discourse and dialogue* (pp. 37–49). Association for Computational Linguistics. Retrieved from https://www.aclweb.org/anthology/W17-5506 https://doi.org/10.18653/v1/W17-5506

Hone, K. S., & Graham, R. (2000). Towards a tool for the subjective assessment of speech system interfaces (SASSI). *Natural Language Engineering*, *6*(3&4), e02497. https://doi.org/10.1017/S135132490002497

Huiyang, S., & Min, W. (2022). Improving interaction experience through lexical convergence: The prosocial effect of lexical alignment in human-human and humancomputer interactions. *International Journal of Human–Computer Interaction*, *38*(1), 28–41. https://doi.org/10.1080/10447318.2021.1921367

Jurafsky, D., & Martin, J. H. (2009). *Speech and language processing* (2nd ed.). Prentice-Hall.

Laranjo, L., Dunn, A. G., Tong, H. L., Kocaballi, A. B., Chen, J., Bashir, R., Surian, D., Gallego, B., Magrabi, F., Lau, A. Y. S., & Coiera, E. (2018). Conversational agents in healthcare: A systematic review. *Journal of the American Medical Informatics Association*, *25*(9), 1248–1258. https://doi.org/10.1093/jamia/ocy072

Lee, K. J., Chauhan, A., Goh, J., Nilsen, E., & Law, E. (2021). Curiosity notebook: The design of a research platform for learning by teaching. *Proceedings of the ACM on Human-Computer Interaction*, *5*(CSCW2), 1–26. https://doi.org/10.48550/arXiv.2108.09809

Linardatos, P., Papastefanopoulos, V., & Kotsiantis, S. (2020). Explainable AI: A review of machine learning interpretability methods. *Entropy*, *23*(1), 18. https://doi.org/10.3390/e23010018

Madotto, A., Cahyawijaya, S., Winata, G. I., Xu, Y., Liu, Z., Lin, Z., & Fung, P. (2020). Learning knowledge bases with parameters for task-oriented dialogue systems. arXiv preprint arXiv:2009.13656.

Madotto, A., Wu, C.-S., & Fung, P. (2018). Mem2Seq: Effectively incorporating knowledge bases into end-to-end task-oriented dialog systems. In *Proceedings of the 56th annual meeting of the association for computational linguistics* (volume 1 (Long papers), pp. 1468–1478). https://doi.org/10.18653/v1/P18-1136

McTear, M. (2020). Conversational AI: Dialogue systems, conversational agents, and chatbots. *Synthesis Lectures on Human Language Technologies*, *13*(3), 1–251. https://doi.org/10.2200/S01060ED1V01Y202010HLT048

Nobani, N., Mercorio, F., Mezzanzanica, M. (2021). Towards an explainer-agnostic conversational xai. In *Proceedings of the thirtieth international joint conference on artificial intelligence*, ijcai-21 (pp. 4909–4910).

Otter, D. W., Medina, J. R., & Kalita, J. K. (2021). A survey of the usages of deep learning for natural language processing. *IEEE*

Transactions on Neural Networks and Learning Systems, *32*(2), 604–624. https://doi.org/10.1109/TNNLS.2020.2979670

Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002). BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the association for computational linguistics* (pp. 311–318).

Ping, Q., Niu, F., Thattai, G., Chengottusseriyil, J., Gao, Q., Reganti, A., Rajagopal, P., Tur, G., Hakkani-Tur, D., & Nataraja, P. (2020). Interactive teaching for conversational AI. arXiv:2012.00958. https://doi.org/10.48550/arXiv.2012.00958

Qin, L., Liu, Y., Che, W., Wen, H., Li, Y., & Liu, T. (2019, November). Entityconsistent end-to-end task-oriented dialogue system with KB retriever. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing* (emnlp-ijcnlp) (pp. 133–142). Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-1013

Qin, L., Xu, X., Che, W., Zhang, Y., & Liu, T. (2020, July). Dynamic fusion network for multi-domain end-to-end task-oriented dialog. In *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 6344–6354). Association for Computational Linguistics. Retrieved from https://www.aclweb.org/anthology/2020.acl-main.565, https://doi.org/10.18653/v1/2020.acl-main.565

Raghu, D., Gupta, N., Mausam. (2019). Disentangling language and knowledge in task-oriented dialogs. In *Proceedings of the 2019 conference of the north American chapter of the association for computational linguistics: Human language technologies*, (volume 1 (long and short papers), pp. 1239–1255).

Rheu, M., Shin, J. Y., Peng, W., & Huh-Yoo, J. (2021). Systematic review: Trustbuilding factors and implications for conversational agent design. *International Journal of Human–Computer Interaction*. *37*(1), 81–96. https://doi.org/10.1080/10447318.2020.1807710

Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, *1*(5), 206–215. https://doi.org/10.1038/s42256-019-0048-x

Sejnowski, T. J. (2018). *The deep learning revolution*. MIT Press.

Serban, I. V., Sankar, C., Germain, M., Zhang, S., Lin, Z., Subramanian, S., Kim, T., Pieper, M., Chandar, S., Rosemary Ke, N., Rajeshwar, S., de Brebisson, A., Sotelo, J. M. R., Suhubdy, D., Michalski, V., Nguyen, A., Pineau, J., & Bengio, Y. (2017). A deep reinforcement learning chatbot. arXiv preprint arXiv:1709.02349.

Stephanidis, C., Salvendy, G., Antona, M., Chen, J. Y. C., Dong, J., Duffy, V. G., Fang, X., Fidopiastis, C., Fragomeni, G., Fu, L. P., Guo, Y., Harris, D., Ioannou, A., Jeong, K-a., Konomi, S., Krömker, H., Kurosu, M., Lewis, J. R., Marcus, A., … Zhou, J. (2019). Seven HCI grand challenges. *International Journal of Human–Computer Interaction*, *35*(14), 1229–1269. https://doi.org/10.1080/10447318.2019.1619259

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems, 3104–3112*. https://doi.org/10.48550/arXiv.1409.3215

Tiedemann, J. (2009). News from OPUS–A collection of multilingual parallel corpora with tools and interfaces. *In Recent Advances in Natural Language Processing 5*, 237–248. https://doi.org/10.1075/cilt.309.19tie

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems*, 5998–6008. arXiv:1706.03762v5

Vinyals, O., & Le, Q. (2015). A neural conversational model. arXiv preprint arXiv:1506.05869.

Wahde, M. (2019). A dialogue manager for task-oriented agents based on dialogue building-blocks and generic cognitive processing. In *2019 IEEE International Symposium on INnovations in Intelligent SysTems and Applications* (INISTA) (pp. 1–8). https://doi.org/10.1109/INISTA.2019.8778354

Wahde, M., & Virgolin, M. (2021). The five Is: Key principles for interpretable and safe conversational AI. *Submitted to the 23rd ACM International Conference on Multimodal Interaction (ICMI)*. https://doi.org/10.1145/3507623.3507632

Wahde, M., & Virgolin, M. (2022). Conversational agents: Theory and applications. Handbook of Computer Learning and Intelligence (to appear), Retrieved from https://arxiv.org/abs/2202.03164

Wang, S. I., Ginn, S., Liang, P., Manning, C. D. (2017). Naturalizing a programming language via interactive learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics* (volume 1 (Long papers), pp. 929–938).

Ward, W., Issar, S., Huang, X., Hon, H.-W., Hwang, M.-Y., Young, S., Matessa, M., Liu, F.-H., & Stern, R. M. (1992). Speech understanding in open tasks. In *Speech and natural language: Proceedings of a workshop held at harriman*, New York February 23–26, 1992.

Wen, H., Liu, Y., Che, W., Qin, L., Liu, T. (2018). Sequence-to-sequence learning for task-oriented dialogue with dialogue state representation. In *Proceedings of the 27th International Conference on Computational Linguistics* (pp. 3781–3792).

Wen, T.-H., Vandyke, D., Mrksic, N., Gasic, M., Rojas-Barahona, L. M., Su, P.-H., Ultes, S., & Young, S. (2017). A network-based end-to-end trainable task-oriented dialogue system. In *Proceedings of the 15th conference of the European Chapter of the Association for Computational Linguistics* (pp. 438–449). Association for Computational Linguistics.

Werner, C. (2020). Explainable ai through rule-based interactive conversation. In *Edbt/icdt workshops*.

Wu, C.-S., Socher, R., & Xiong, C. (2019). Global-to-local memory pointer networks for task-oriented dialogue. arXiv preprint arXiv:1901.04713.

Yin, P., Neubig, G. (2017). A syntactic neural model for general-purpose code generation. In *Proceedings of the 55th annual meeting of the association for computational linguistics* (volume 1: Long papers) (pp. 440–450).

Young, T., Hazarika, D., Poria, S., & Cambria, E. (2018). Recent trends in deep learning based natural language processing. *IEEE Computational Intelligence Magazine*, *13*(3), 55–75. https://doi.org/10.1109/MCI.2018.2840738

Zhou, L., Gao, J., Li, D., & Shum, H.-Y. (2020). The design and implementation of XiaoIce, an empathetic social chatbot. *Computational Linguistics*, *46*(1), 53–93. https://doi.org/10.1162/coli_a_00368

About the Authors

Mattias Wahde is professor of Applied Artificial Intelligence at Chalmers University of Technology in Gothenburg, Sweden. His research interests cover several aspects of artificial intelligence, e.g., conversational AI, mobile robots (particularly human-machine interaction), and stochastic optimization. He also teaches several courses covering various topics in artificial intelligence.

Marco Virgolin is a junior researcher at Centrum Wiskunde & Informatica, the Netherlands. He works on explainable and interpretable artificial intelligence (AI), mostly by means of evolutionary machine learning methods. He is also interested in medical applications of machine learning, conversational AI, and human-machine interaction.