nature machine intelligence

Article

https://doi.org/10.1038/s42256-023-00692-8

Explaining machine learning models with interactive natural language conversations using TalkToModel

Received: 3 October 2022

Accepted: 22 June 2023

Published online: 27 July 2023

Check for updates

Dylan Slack[1], Satyapriya Krishna[2], Himabindu Lakkaraju[2,3,4] & Sameer Singh[1,4]

Practitioners increasingly use machine learning (ML) models, yet models have become more complex and harder to understand. To understand complex models, researchers have proposed techniques to explain model predictions. However, practitioners struggle to use explainability methods because they do not know which explanation to choose and how to interpret the explanation. Here we address the challenge of using explainability methods by proposing TalkToModel: an interactive dialogue system that explains ML models through natural language conversations. TalkToModel consists of three components: an adaptive dialogue engine that interprets natural language and generates meaningful responses; an execution component that constructs the explanations used in the conversation; and a conversational interface. In real-world evaluations, 73% of healthcare workers agreed they would use TalkToModel over existing systems for understanding a disease prediction model, and 85% of ML professionals agreed TalkToModel was easier to use, demonstrating that TalkToModel is highly effective for model explainability.

Due to their strong performance, machine learning (ML) models increasingly make consequential decisions in several critical domains, such as healthcare, finance and law. However, state-of-the-art ML models, such as deep neural networks, have become more complex and hard to understand. This dynamic poses challenges in real-world applications for model stakeholders who need to understand why models make predictions and whether to trust them. Consequently, practitioners have often turned to inherently interpretable ML models for these applications, including decision lists and sets[1,2] and generalized additive models[3–5], which people can more easily understand. Nevertheless, black-box models are often more flexible and accurate, motivating the development of post hoc explanations that explain the predictions of trained ML models. These explainability techniques either fit faithful models in the local region around a prediction or inspect internal model details, such as gradients, to explain predictions[6–11].

Yet, recent work suggests that practitioners often have difficulty using explainability techniques[12–15]. These challenges are due to difficulty in figuring out which explanations to implement, how to interpret the explanation and answering follow-up questions beyond the initial explanation. In the past, researchers have proposed several point-and-click dashboard techniques to help overcome these issues, such as the Language Interpretability Tool[16], which is designed to understand natural language processing models and the What-If Tool[17]—a tool aimed at performing counterfactual analyses for models. However, these methods still require a high level of expertise, because users must know which explanations to run, and lack the flexibility to support arbitrary follow-up questions that users might have. Overall, understanding ML models through simple and intuitive interactions is a key bottleneck in adoption across many applications.

Natural language dialogues are a promising solution for supporting broad and accessible interactions with ML models due to their ease

[1]Department of Computer Science, University of California Irvine, Irvine, CA, USA. [2]Department of Computer Science, Harvard University, Cambridge, MA, USA. [3]Harvard Business School, Boston, MA, USA. [4]These authors contributed equally: Himabindu Lakkaraju, Sameer Singh. ✉e-mail: dslack@uci.edu

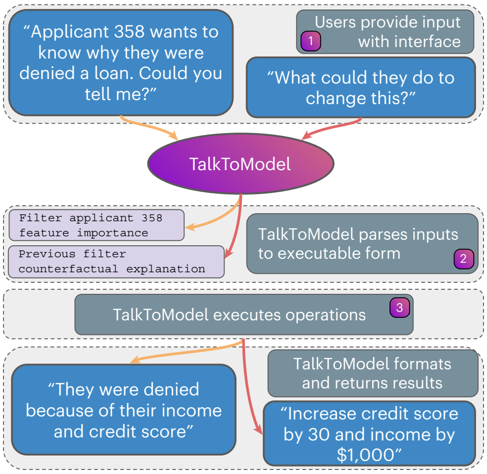
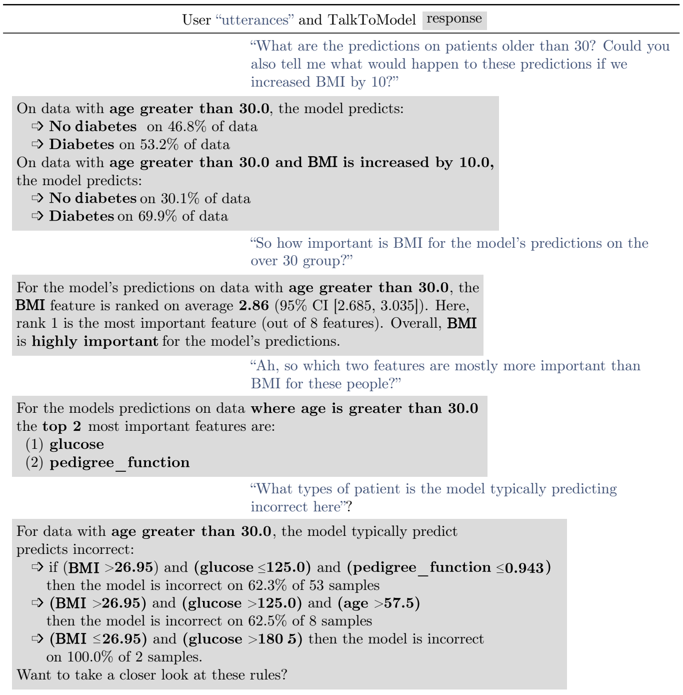

"Applicant 358 wants to know why they were denied a loan. Could you tell me?"
Users provide input with interface
1
"What could they do to change this?"
TalkToModel
Filter applicant 358 feature importance
Previous filter counterfactual explanation
TalkToModel parses inputs to executable form
2
TalkToModel executes operations
3
TalkToModel formats and returns results
"They were denied because of their income and credit score"
"Increase credit score by 30 and income by $1,000"

Fig. 1 | Overview of TalkToModel. Instead of writing code, users have conversations with TalkToModel as follows. (1) Users supply natural language inputs. (2) The dialogue engine parses the input into an executable representation. (3) The execution engine runs the operations and the dialogue engine uses the results in its response.

of use, capacity and support for continuous discussion. However, designing a dialogue system that enables a satisfying model understanding experience introduces several challenges. First, the system must handle many conversation topics about the model and data while facilitating natural conversation flow[18]. For instance, these topics may include explainability questions like the most important features for predictions and general questions such as data statistics or model errors. Further, the system must work for various model classes and data, and it should understand language usage across different settings[19]. For example, participants will use different terminology in conversations about loan prediction than disease diagnosis. Last, the dialogue system should generate accurate responses that address the users' core questions[20,21]. In the literature, researchers have suggested some prototype designs for generating explanations using natural language. However, these initial designs address specific explanations and model classes, limiting their applicability in general conversational explainability settings[22,23].

In this Article, we address these challenges by introducing TalkToModel: a system that enables open-ended natural language dialogues for understanding ML models for any tabular dataset and classifier (an overview of TalkToModel is provided in Fig. 1). Users can have discussions with TalkToModel about why predictions occur, how the predictions would change if the data change and how to flip predictions, among many other conversation topics (an example conversation is provided in Fig. 2). Further, they can perform these analyses on any group in the data, such as a single instance or a specific group of instances. For example, on a disease prediction task, users can ask 'How important is BMI for the predictions?' or 'So how would decreasing the glucose levels by 10 change the likelihood of men older than 20 having the disease?'. TalkToModel will respond by describing how, for instance, BMI is the most important feature for predictions, and decreasing glucose will decrease the chance of diabetes by 20%. From there, users can engage further in the conversation by asking follow-up questions. Conversations with TalkToModel make model explainability straightforward because users can talk with the system in natural language about the model, and the system will generate useful responses.

To support such rich conversations with TalkToModel, we introduce techniques for both language understanding and model explainability. First, we propose a dialogue engine that parses user text inputs (referred to as user utterances) into a structured query language-like programming language using a large language model (LLM). The LLM performs the parsing by treating the task of translating user utterances into the programming language as a seq2seq learning problem, where the user utterances are the source and parses in the programming language are the targets[24]. In addition, the TalkToModel language combines operations for explanations, ML error analyses, data manipulation and descriptive text into a single language capable of representing a wide variety of potential conversation topics most model explainability needs (an overview of the different operations is provided in Fig. 3). To support the system adapting to any dataset and model, we introduce lightweight adaption techniques to fine-tune LLMs to perform the parsing, enabling strong generalization to new settings. Second, we introduce an execution engine that runs the operations in each parse. To reduce the burden of users deciding which explanations to run, we introduce methods that automatically select explanations for the user. In particular, this engine runs many explanations, compares their fidelities and selects the most accurate ones. Finally, we construct a text interface where users can engage in open-ended dialogues using the system, enabling anyone, including those with minimal technical skills, to understand ML models.



User "utterances" and TalkToModel response

"What are the predictions on patients older than 30? Could you also tell me what would happen to these predictions if we increased BMI by 10?"

On data with **age greater than 30.0**, the model predicts:
➩ **No diabetes** on 46.8% of data
➩ **Diabetes** on 53.2% of data
On data with **age greater than 30.0 and BMI is increased by 10.0,** the model predicts:
➩ **No diabetes** on 30.1% of data
➩ **Diabetes** on 69.9% of data

"So how important is BMI for the model's predictions on the over 30 group?"

For the model's predictions on data with **age greater than 30.0**, the **BMI** feature is ranked on average **2.86** (95% CI [2.685, 3.035]). Here, rank 1 is the most important feature (out of 8 features). Overall, **BMI** is **highly important** for the model's predictions.

"Ah, so which two features are mostly more important than BMI for these people?"

For the models predictions on data **where age is greater than 30.0** the **top 2** most important features are:
(1) **glucose**
(2) **pedigree_function**

"What types of patient is the model typically predicting incorrect here"?

For data with **age greater than 30.0**, the model typically predict predicts incorrect:
➩ if (**BMI >26.95**) and (**glucose ≤125.0**) and (**pedigree_function ≤0.943**) then the model is incorrect on 62.3% of 53 samples
➩ (**BMI >26.95**) and (**glucose >125.0**) and (**age >57.5**) then the model is incorrect on 62.5% of 8 samples
➩ (**BMI ≤26.95**) and (**glucose >180 5**) then the model is incorrect on 100.0% of 2 samples.
Want to take a closer look at these rules?

Fig. 2 | A conversation with TalkToModel. A conversation about diabetes prediction, demonstrating the breadth of different conversation points the system can discuss.

Results

In this section, we demonstrate that TalkToModel accurately understands users in conversations by evaluating its language understanding capabilities on ground-truth data. Next, we evaluate the effectiveness of TalkToModel for model understanding by performing a real-world human study on healthcare workers (for example, doctors and nurses) and ML practitioners, where we benchmark TalkToModel against existing explainability systems. We find users both prefer and are more effective using TalkToModel than traditional point-and-click explainability systems, demonstrating its effectiveness for understanding ML models.

Language understanding

Here we quantitatively assess the language understanding capabilities of TalkToModel by creating gold parse datasets and evaluating the system's accuracy on these data.

	operation, arguments, and description
Data	filter(dataset, feature, value, comparison): Filters dataset by using value and comparison operator
	change(dataset, feature, value, variation): Changes dataset by increasing, decreasing, or setting feature by value
	show(list): Shows items in list in the conversation
	statistic(dataset, metric, feature): Computes summary statistic for feature
	count(list): Length of list
	and(op1, op2): Logical @and< of two operations
	or(op1, op2): Logical <or< of two operations
Explainability	explain(dataset, method, class=predicted): Feature importances on dataset
	cfe(dataset, number, class=opposite): Gets number counterfactual explanations
	topk(dataset, k): Top k most important features
	important(dataset, feature): Importance ranking of feature
	interaction(dataset): Interaction effects between features
	mistakes(dataset): Patterns in the model's errors on dataset
ML	predict(dataset): Model predictions on dataset
	likelihood(dataset): Prediction probabilities on dataset
	incorrect(dataset): Incorrect predictions
	score(dataset, metric): Scores the model with metric
Conv.	prev_filter(conversation): Gets last filters
	prev_operation(conversation): Gets last non-filtering operations
	followup(conversation): Respond to system follow-ups
Description	function(): Overview of the system's capabilities
	data(dataset): Summary of dataset
	model(): Description of model
	define(term): Defines term

Fig. 3 | Overview of the operations supported by TalkToModel. The operations are incorporated into the conversation to generate responses. Note, Conv. refers to Conversation operations.

Gold parse collection. We construct gold datasets (that is, ground-truth (utterance, parse) pairs) across multiple datasets to evaluate the language understanding performance of our models. To construct these gold datasets, we adopt an approach inspired by ref. 25, which constructs a similar dataset for multitask semantic parsing.

Our gold dataset-generation process is as follows. First, we write 50 (utterance, parse) pairs for the particular task (that is, loan or diabetes prediction). These utterances range from simple 'How likely are people in the data to have diabetes?' to complex 'If these people were not unemployed, what's the likelihood they are good credit risk? Why?'. We include each operation (Fig. 3) at least twice in the parses, to make sure that there is good coverage. From there, we ask Mechanical Turk workers to rewrite the utterances while preserving their semantic meaning to ensure that the ground-truth parse for the revised utterance is the same but the phrasing differs. We ask workers to rewrite each pair 8 times for a total of 400 (utterance, parse) pairs per task. Next, we filter out low-quality mturk revisions. We ask the crowd-sourced workers to rate the similarity between the original utterance and revised utterance on a scale of 1 to 4, where 4 indicates that the utterances have the same meaning and 1 indicates that they do not have the same meaning. We collect 5 ratings per revision and remove (utterance, parse) pairs that score below 3.0 on average. Finally, we perform an additional filtering step to ensure data quality by inspecting the remaining pairs ourselves and removing any bad revisions.

As we want to evaluate TalkToModel's capacity to generalize across different scenarios, we perform this data collection process across three different tasks: Pima Indian Diabetes Dataset[26], German credit dataset[26] and the Correctional Offender Management Profiling for Alternative Sanctions (COMPAS) recidivism dataset[27]. After collecting revisions and ensuring quality, we are left with 200 pairs for the German credit dataset, 190 for the diabetes dataset and 146 for the COMPAS dataset.

Models. We compare two strategies for using pre-trained LLMs to parse user utterances into the grammar: (1) few-shot GPT-J [28] and GPT-3.5 models[29] and (2) fine-tuned T5[30]. The GPT-J and GPT-3.5 models are higher capacity and more amenable to be trained by in-context learning. This procedure includes examples of the input and target from the training prepended to the test instance[29,31,32]. In contrast, the T5 models require traditional fine-tuning on the input and target pairs. Consequently, the few-shot approach is quicker to set up because it does not require fine-tuning, making it easier for users to get started with the system. However, the fine-tuned T5 leads to improved performance and a better user experience overall while taking longer to set up. We expect that fine-tuned T5 leads to improved performance overall because it has access to all the training data, whereas, the few-shot models are limited by the context window size. To train these models through fine-tuning or prompting, we generate synthetic (utterance, parse) pairs because it is impractical to assume that we can collect ground-truth pairs for every new task we wish to use TalkToModel. We provide additional training details in Methods.

We evaluate both fine-tuned T5 models and few-shot models on the testing data. We additionally implement a naive nearest-neighbours baseline, where we select the closest user utterance in the synthetic training set according to cosine distance of all-mpnet-base-v2 sentence embeddings and return the corresponding parse[33]. For the GPT-J models, we compare N-shot performance, where N is the number of (utterance, parse) pairs from the synthetically generated training sets included in the prompt, and sweep over a range of N for each model. For the larger models, we have to use relatively smaller N for inference to fit on a single 48 GB graphics processing unit.

When parsing the utterances, one issue is that their generations are unconstrained and may generate parses outside the grammar, resulting in the system failing to run the parse. To ensure the generations are grammatical, we constrain the decodings to be in the grammar by recompiling the grammar at inference time into an equivalent grammar consisting of the tokens in the LLM's vocabulary [34]. While decoding from the LLM, we fix the likelihood of ungrammatical tokens to 0 at every generation step. Because the GPT-3.5 model must be called through an application programming interface, which does not support guided decoding, we decode greedily with temperature set to one.

Evaluating the parsing accuracy. To evaluate performance on the datasets, we use the exact match parsing accuracy[25,35,36]. This metric is whether the parse exactly matches the gold parse in the dataset. In addition, we perform the evaluation on two splits of each gold parse dataset, in addition to the overall dataset. These splits are the independent and identically distributed (IID) and compositional splits. The IID split contains (utterance, parse) pairs where the parse's operations and their structure (but not necessarily the arguments) are in the training data. The compositional split consists of the remaining parses that are not in the training data. Because language models struggle compositionally, this split is generally much harder for language models to parse[37,38].

Accuracy. We present the results in Table 1. T5 performs better overall than the few-shot GPT-J and GPT-3.5 models. Notably, the T5 small model performs better than the GPT-J 6B model, which has two orders of magnitude more parameters. While the few-shot models underperform the fine-tuned T5 models overall, GPT-3.5 is the best-performing few-shot model and performs considerably better than the GPT-J models, particularly in the compositional split. Overall, these results suggest using fine-tuned T5 for the best results, and we use T5 large in our human studies.

Utility of explainability dialogues

The results in the previous section show that TalkToModel understands user intentions to a high degree of accuracy. In this section, we evaluate how well the end-to-end system helps users understand ML models compared with current explainability systems.

Table 1 | Exact match parsing accuracy (%) for the three gold datasets, on the IID and compositional splits, and overall

	German credit			COMPAS			Diabetes		
	IID	Compositional	Overall	IID	Compositional	Overall	IID	Compositional	Overall
Nearest neighbours	26.2	0.0	16.5	27.4	0.0	21.9	10.9	0.0	8.4
GPT-Neo 1.3B									
10-shot	41.3	4.1	27.5	35.9	0.0	28.8	40.1	7.0	32.6
20-shot	39.7	0.0	25.0	39.3	0.0	31.5	42.9	2.3	33.7
30-shot	42.9	0.0	27.0	39.3	0.0	31.5	41.5	4.7	33.2
GPT-Neo 2.7B									
5-shot	38.1	4.1	25.5	35.9	3.4	29.5	46.9	7.0	37.9
10-shot	38.1	6.8	26.5	40.2	3.4	32.9	40.8	9.3	33.7
20-shot	39.7	0.0	25.0	39.3	0.0	31.5	42.9	2.3	33.7
GPT-J 6B									
5-shot	51.6	14.9	38.0	51.3	6.9	42.5	55.8	7.0	44.7
10-shot	57.9	9.5	40.0	49.6	3.4	40.4	53.7	9.3	43.7
GPT-3.5									
10-shot	44.4	28.3	38.5	38.8	13.8	34.9	40.1	16.3	34.7
20-shot	60.3	41.2	53.5	41.9	17.2	37.0	45.6	23.3	40.5
30-shot	58.7	43.2	53.0	47.0	**24.1**	42.7	55.1	32.6	50.0
T5									
Small	61.1	32.4	50.5	71.8	10.3	59.6	77.6	30.2	66.8
Base	68.3	**48.6**	61.0	65.0	10.3	54.1	**84.4**	34.9	73.2
Large	**74.6**	44.6	**63.5**	**76.9**	**24.1**	**66.4**	**84.4**	**51.2**	**76.8**

All the evaluated models except for GPT-3.5 use guided decoding. GPT-3.5 does not use guided decoding because it is not supported through the OpenAI application programming interface. The most accurate model is bolded. The fine-tuned T5 models tended to perform better than the few-shot models, and T5 large performed the best. These results demonstrate that TalkToModel can understand user intentions with a high degree of accuracy using the T5 models.

Study overview. We compare TalkToModel against 'explainerdashboard', one of the most popular open-source explainability dashboards[39]. This dashboard has similar functionality to TalkToModel, considering it provides an accessible way to compute explanations and perform model analyses. Thus, it is a reasonable baseline. Last, we perform this comparison using the diabetes dataset, and a gradient-boosted tree trained on the data[40]. To compare both systems in a controlled manner, we ask participants to answer general ML questions with TalkToModel and the dashboard. Each question is about basic explainability and model analysis, and participants answer using multiple choice, where one of the options is 'Could not determine' if they cannot figure out the answer (although it is straightforward to answer all the questions with both interfaces). For example, questions are about comparing feature importances 'Is glucose more important than age for the model's predictions for data point 49?' or model predictions 'How many people are predicted not to have diabetes but do not actually have it?' Participants answer ten questions in total. We divide the ten questions into two blocks of five questions each. Both blocks have similar questions but different values to control for memorization (the exact questions are given in Supplementary Section A). Participants use TalkToModel to answer one block of questions and the dashboard for the other block. In addition, we provide a tutorial on how to use both systems before showing users the questions for the system. Last, we randomize question, block and interface order to control for biases due to showing interfaces or questions first.

Metrics. Following previous work on evaluating human and ML coordination and trust, we assessed several metrics to evaluate user experiences[41–43]. We evaluated the following statements along the 1–7 Likert scale at the end of the survey:

- Easiness: I found the conversational interface easier to use than the dashboard interface
- Confidence: I was more confident in my answers using the conversational interface than the dashboard interface
- Speed: I felt that I was able to more rapidly arrive at an answer using the conversational interface than the dashboard interface
- Likeliness to use: based on my experience so far with both interfaces, I would be more likely to use the conversational interface than the dashboard interface in the future

To control for bias associated with the ordering of the terms conversational interface and dashboard interface, we randomized their ordering. We also measured accuracy and time taken to answer each question. Last, we asked to participants to write a short description comparing their experience with both interfaces to capture participants qualitative feedback about both systems.

Recruitment. As TalkToModel provides an accessible way to understand ML models, we expect it to be useful for subject-matter experts with a variety of experience in ML, including users without any ML experience. As such, we recruited 45 English-speaking healthcare workers to take the survey using the Prolific service[44] with minimal or no ML expertise This group comprises a range of healthcare workers, including doctors, pharmacists, dentists, psychiatrists, healthcare project managers and medical scribes. The vast majority of this group (43) stated they had either no experience with ML or had heard about it from reading articles online, while two members indicated they had equivalent to an undergraduate course in ML. As another point of comparison, we recruited ML professionals with relatively higher ML expertise from ML Slack channels and email lists. We received 13 potential participants, all of which had graduate-course-level

Table 2 | User study results for respondents that agree TalkToModel is better than the dashboard

	Agree TalkToModel better (%)	
Comparison	Healthcare workers	ML graduate students
Easiness	82.2	84.6
Confidence	77.7	69.2
Speed	84.4	84.6
Likeliness to use	73.3	53.8

The percentage of respondents that agree (>neutral Likert score) that TalkToModel is better than the dashboard in the four comparison questions. A substantial portion of respondents agreed TalkToModel is better than the dashboard in all the categories except graduate students and 'Likeliness to use'. Still, a majority agreed that TalkToModel was superior in this case.

ML experience or higher, and included all of them in the study. We received institutional review board approval for this study from the University of California, Irvine institutional review board approval process and informed consent from participants.

Metric results. A substantial majority of healthcare workers agreed that they preferred TalkToModel in all the categories we evaluated (Table 2). The same is true for the ML professionals, save for whether they were more likely to use TalkToModel in the future, where 53.8% of participants agreed they would instead use TalkToModel in the future. In addition, participants' subjective notions around how quickly they could use TalkToModel aligned with their actual speed of use, and both groups arrived at answers using TalkToModel significantly quicker than using the dashboard. The median question answer time (measured at the total time taken from seeing the question to submitting the answer) using TalkToModel was 76.3 s, while it was 158.8 s using the dashboard.

Participants were also much more accurate and completed questions at a higher rate (that is, they did not mark 'Could not determine') using TalkToModel (Table 3). While both healthcare workers and ML practitioners clicked 'Could not determine' for a quarter of the questions using the dashboard, this was true for 13.8% of healthcare workers and 6.1% of ML professionals using TalkToModel, demonstrating the usefulness of the conversational interface. On completed questions, both groups were much more accurate using TalkToModel than the dashboard. Most surprisingly, although ML professionals agreed that they preferred TalkToModel only about half the time, they answered all the questions correctly using it, while they only answered 62.5% of questions correctly with the dashboard. Finally, we observed that TalkToModel's conversational capabilities were highly effective. There were only 6 utterances out of over 1, 000 total utterances that the conversational aspect of the system failed to resolve. These failure cases generally involved certain discourse aspects like asking for additional elaboration ('more description').

The largest source of errors for participants using the explainability dashboard were two questions concerning the top most important features for individual predictions. The errors for these questions account for 47.4% of healthcare workers and 44.4% of ML professionals' total mistakes. Solving these tasks with the dashboard requires users to perform multiple steps, including choosing the feature importance tab in the dashboard, while the streamlined text interface of TalkToModel made it much simpler to solve these tasks.

Qualitative results. For the qualitative user feedback, we provide representative quotes from similar themes in the responses. Users expressed that they could more rapidly and easily arrive at results, which could be helpful for their professions.

Displayquote 1. *"I prefer the conversational interface because it helps arrive at the answer very quickly. This is very useful especially in the hospital setting where you have hundreds of patients getting check ups and screenings for diabetes because it is efficient and you can work with medical students on using the system to help patient outcomes."* P39 medical worker at a tertiary hospital.

Participants also commented on the user friendliness of TalkToModel and its strong conversational capabilities, stating, "the conversational [interface] was straight to the point, way easier to use" (P35 nurse) and that "the conversational interface is hands-down much easier to use… it feels like one is talking to a human" (P45 ML professional). We did not find any negative feedback surrounding the conversational capabilities of the system. Overall, users expressed strong positive sentiment about TalkToModel due to the quality of conversations, presentation of information, accessibility and speed of use.

Several ML professionals brought up points that could serve as future research directions. Notably, participants stated that they would rather look at the data themselves rather than rely on an interface that rapidly provides an answer.

Displayquote 2. *"I would almost always rather look at the data myself and come to a conclusion than getting an answer within seconds."* P11 ML professional.

In the future, it would be worthwhile including visualizations of raw data and analyses performed by the system to increase trust with expert users, such as ML professionals, who may be sceptical of the high-level answers provided by the system currently.

Discussion

With ML models becoming increasingly complex, there is a need to develop techniques to explain model predictions to stakeholders. Nevertheless, it is often the case that practitioners struggle to use explanations and frequently have many follow-up questions they wish to answer. In this work, we show that TalkToModel makes explainable AI accessible to users from a range of backgrounds by using natural language conversations. Our experiments demonstrate that TalkToModel comprehends users with a high degree of accuracy and can help users understand the predictions of ML models much better than existing systems can. In particular, we showed that TalkToModel is a highly effective way for domain experts such as healthcare workers to understand ML models, like those applied to disease diagnosis. Lastly, we designed TalkToModel to be highly extensible and released the code, data and a demo for the system at https://github.com/dylan-slack/TalkToModel, making it straightforward for users and researchers to build on the system. In the future, it will be helpful to investigate applications of TalkToModel 'in the wild', such as in doctors' offices, laboratories or professional settings, where model stakeholders could use the system to understand their models.

Methods

In this section, we describe the components of TalkToModel. First, we introduce the dialogue engine and discuss how it understands user inputs, maps them to operations and generates text responses based on the results of running the operations. Second, we describe the execution engine, which runs the operations. Finally, we provide an overview of the interface and the extensibility of TalkToModel.

Text understanding

To understand the intent behind user utterances, the system learns to translate or parse them into logical forms. These parses represent the intentions behind user utterances in a highly expressive and structured programming language TalkToModel executes.

Compared with dialogue systems that execute specific tasks by modifying representations of the internal state of the conversation[45,46], our parsing-based approach allows for more flexibility in the conversations, supporting open-ended discovery, which is critical for model understanding. Also, this strategy produces a structured representation of user utterances instead of open-ended systems that generate

Table 3 | User study results for completion rate and accuracy across interfaces and participant groups

	Questions completed (%)		Accuracy on completed questions (%)	
	Dashboard	TalkToModel	Dashboard	TalkToModel
Healthcare workers	74.7	86.2	66.1	91.8
ML graduate students	73.8	93.9	62.5	100.0

We computed the completion rate as the questions users provided an answer for and did not mark 'Could not determine'. We measured accuracy on completed questions. Participants answered questions at a higher rate more accurately using TalkToModel than the dashboard.

unstructured free text[47]. Having this structured representation of user inputs is key for our setting where we need to execute specific operations depending on the user's input, which would not be straightforward with unstructured text.

TalkToModel performs the following steps to accomplish this: (1) the system constructs a grammar for the user-provided dataset and model, which defines the set of acceptable parses; (2) TalkToModel generates (utterance, parse) pairs for the dataset and model; (3) the system fine-tunes an LLM to translate user utterances into parses; and (4) the system responds conversationally to users by composing the results of the executed parse into a response that provides context for the results and opportunities to follow up.

Grammar. To represent the intentions behind the user utterances in a structured form, TalkToModel relies on a grammar, defining a domain-specific language for model understanding. While the user utterances themselves will be highly diverse, the grammar creates a way to express user utterances in a structured yet highly expressive fashion that the system can reliably execute. Compared with approaches that treat determining user intentions in conversations as a classification problem[48,49], using a grammar enables the system to express compositions of operations and arguments that take on many different values, such as real numbers, that would otherwise be combinatorially impossible in a prediction setting. Instead, TalkToModel translates user utterances into this grammar in a seq2seq fashion, overcoming these challenges[24]. This grammar consists of production rules that include the operations the system can run (an overview is provided in Table 3), the acceptable arguments for each operation and the relations between operations. One complication is that user-provided datasets have different feature names and values, making it hard to define one shared grammar between datasets. Instead, we update the grammar based on the feature names and values in a new dataset. For instance, if a dataset contained only the feature names 'age' and 'income', these two names would be the only acceptable values for the feature argument in the grammar.

To ensure that our grammar provides sufficient coverage for explainable artificial intelligence (XAI) questions, we verify our grammar supports the questions from the XAI question bank. This question bank was introduced in ref. 50 based on interviews with AI product designers and includes 31 core, prototypical questions XAI systems should answer, excluding socio-technical questions beyond the scope of TalkToModel (for example, 'What are the results of other people using the [model]'). The prototypical questions address topics such as the input/data to the model ('What is the distribution of a given feature?'), model output ('What kind of output does the system give?'), model performance ('How accurate are the predictions?'), global model behaviour ('What is the systems overall logic?'), why/why not the system makes individual predictions ('Why is this instance given this prediction?') and what-if or counterfactual questions ('What would the system predict if this instance changes to...?'). To evaluate how well TalkToModel covers these questions, we review each question and evaluate whether our grammar can parse it. Overall, we find that our grammar supports 30 out of 31 of the prototypical questions. We provide a table of each question and corresponding parse in Supplementary Tables 6 and 7. Overall, the grammar covers the vast majority of XAI related questions, and therefore, has good coverage of XAI topics.

Supporting context in dialogues. User conversations with TalkToModel naturally include complex conversational phenomena such as anaphora and ellipsis[51–53]. Meaning, conversations refer back to events earlier in the conversation ('What do you predict for them?') or omit information that must be inferred from conversation ('Now show me for people predicted incorrectly'). However, current language models parse only a single input, making it hard to apply them in settings where the context is important. To support context in the dialogues, TalkToModel introduces on a set of operations in the grammar that determine the context for user utterances. In contrast with approaches that maintain the conversation state using neural representations[45,54], grammar operations allow for much more trustworthy and dependable behaviour while still fostering rich interactions, which is critical for high-stakes settings, and similar mechanisms for incorporating grammar predicates across turns have been shown to achieve strong results[53]. In particular, we leverage two operations: previous filter and previous operation, which look back in the conversation to find the last filter and last operation, respectively. These operations also act recursively. Therefore, if the last filter is a previous filter operation, TalkToModel will recursively call previous filter to resolve the entire stack of filters. As a result, TalkToModel is capable of addressing instances of anaphora and ellipsis by using these operations to resolve the entity via co-reference or infer it from the previous conversation history. This dynamic enables users to have complex and natural conversations with TalkToModel.

Parsing dataset generation. To parse user utterances into the grammar, we fine-tune an LLM to translate utterances into the grammar in a seq2seq fashion. We use LLMs because these models have been trained on large amounts of text data and are solid priors for language understanding tasks. Thus, they can better understand diverse user inputs than training from scratch, improving the user experience. Further, we automate the fine-tuning of an LLM to parse user utterances into the grammar by generating a training dataset of (utterance, parse) pairs. Compared with dataset-generation methods that use human annotators to generate and label datasets for training conversation models[55,56], this approach is much less costly and time consuming, while still being highly effective, and supports users getting conversations running very quickly. This strategy consists of writing an initial set of user utterances and parses, where parts of the utterances and parses are wildcard terms. TalkToModel enumerates the wildcards with aspects of a user-provided dataset, such as the feature names, to generate a training dataset. Depending on the user-provided dataset schema, TalkToModel typically generates anywhere from 20,000 to 40,000 pairs. Last, we have already written the initial set of utterances and parses, so users only need to provide their dataset to set up a conversation.

Semantic parsing. Here we provide additional details about the semantic parsing approach for translating user utterances into the grammar. The two strategies for parsing user utterances using pre-trained LLMs that we considered were (1) few-shot GPT-J[28] and (2) fine-tuned T5[30]. With respect to the few-shot models, because the LLM's context window accepts only a fixed number of inputs, we introduce a technique to select the set of most relevant prompts for the user utterance. In particular, we embed all the utterances and identify the closest utterances to the user utterance according to the cosine distance of these embeddings. To ensure a diverse set of prompts, we select only one prompt per template. We prompt the LLM using these (utterance, parse) pairs, ordering the closest pairs immediately before the user utterance because LLMs exhibit recency biases[57]. Using this strategy, we experiment with the number of prompts included in the LLM's context window. In practice, we use the all-mpnet-base-v2 sentence transformer model to perform the embeddings[33], and we consider the GPT-J 6B, GPT-Neo 2.7B and GPT-Neo 1.3B models in our experiments.

We also fine-tune pre-trained T5 models in a seq2seq fashion on our datasets. To perform fine-tuning, we split the dataset using a 90%/10% train/validation split and train for 20 epochs to maximize the next token likelihood with a batch size of 32. We select the model with the lowest validation loss at the end of each epoch. We fine-tune with a learning rate of 1×10^{-4} and the AdamW optimizer[58]. Last, our experiments consider the T5 small, base and large variants.

Generating responses. After TalkToModel executes a parse, it composes the results of the operations into a natural language response that it returns to the user. TalkToModel generates these responses by filling in templates associated with each operation based on the results. The responses also include sufficient context to understand the results and opportunities for following up (examples in Table 2). Further, because the system runs multiple operations in one execution, TalkToModel joins response templates, ensuring semantic coherence, into a final response and shows it to the user. Compared with approaches that generate responses using neural methods[59], this approach ensures that the responses are trustworthy and do not contain useless information hallucinated by the system, which would be a very poor user experience for the high-stakes applications we consider. Further, because TalkToModel supports a wide variety of different operations, this approach ensures sufficient diversity in responses, so they are not repetitive.

Executing parses

In this section, we provide an overview of the execution engine, which runs the operations necessary to respond to user utterances in the conversation. Further, this component automatically selects the most faithful explanations for the user, helping ensure explanation accuracy.

Feature importance explanations. At its core, TalkToModel explains why the model makes predictions to users with feature importance explanations. Feature importance explanations $\phi(\mathbf{x}, f) \to \boldsymbol{\Phi}$ accept a data point $\mathbf{x} \in \mathbb{R}^d$ with d features and model as input $f(\mathbf{x}) \to \mathbf{y}$, where $\mathbf{y} \in [0, 1]$ is the probability for a particular class, and generates a feature attribution vector $\Phi \in \mathbb{R}^d$, where greater magnitudes correspond to higher importance features[6,7,60–63].

We implement the feature importance explanations using post hoc feature importance explanations. Post hoc feature importance explanations do not rely on internal details of the model f (for example, internal weights or gradients) and only on the input data $\mathbf{x}$ and predictions $\mathbf{y}$ to compute explanations, so users are not limited to only certain types of model[64–68]. Note that our system can easily be extended to other explanations that rely on internal model details, if required[4,8,69,70].

Explanation selection. While there exists several post hoc explanation methods, each one adopts a different definition of what constitutes an explanation[71]. For example, while local interpretable model agnostic explanations (LIME), Shapley additive explanations (SHAP) and integrated gradients all output feature attributions, LIME returns coefficients of local linear models, SHAP computes Shapley values and integrated gradients leverages model gradients. Consequently, we automatically select the most faithful explanation for users, unless a user specifically requests a certain technique. Following previous works, we compute faithfulness by perturbing the most important features and evaluating how much the prediction changes[72]. Intuitively, if the feature importance ϕ correctly captures the feature importance ranking, perturbing more important features should lead to greater effects.

While previous works[65,73], compute the faithfulness over many different thresholds, making comparisons harder, or require retraining entirely from scratch, we introduce a single metric that captures the prediction sensitivity to perturbing certain features called the fudge score. This metric is the mean absolute difference between the model's prediction on the original input and a fudged version on $\mathbf{m} \in \{0, 1\}^d$ features

$$\text{Fudge}(f, \mathbf{x}, \mathbf{m}) = \frac{1}{N}\sum_{n=1}^{N} |f(\mathbf{x}) - f(\mathbf{x} + \epsilon_n \odot \mathbf{m})| \tag{1}$$

where $\odot$ is the tensor product and $\epsilon \approx \mathcal{N}(0, I\sigma)$ is $N \times d$-dimensional Gaussian noise. To evaluate faithfulness for a particular explanation method, we compute area under the fudge score curve on the top-k most important features, thereby summarizing the results into a single metric

$$\mathbb{1}(k, \Phi) = \begin{cases} 1 & \text{if} \phi_i \in \arg\max\limits_{\Phi' \subset \{1 \ldots d\}, |\Phi'| = k} \sum_{i \in \Phi'} |\phi_i| \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

$$\text{Faith}(\Phi, f, \mathbf{x}, K) = \sum_{k=1}^{K} \text{Fudge}(f, \mathbf{x}, \mathbb{1}(k, \Phi)) \tag{3}$$

where $\mathbb{1}(k, \Phi)$ is the indicator function for the top-k most important features. Intuitively, if a set of feature importances $\boldsymbol{\Phi}$ correctly identifies the most important features, perturbing them will have greater effects on the model's predictions, resulting in higher faithfulness scores. We compute faithfulness for multiple different explanations and select the highest. In practice, we consider LIME[64] with the following kernel widths [0.25, 0.50, 0.75, 1.0] and KernelSHAP[74]. We leave all settings to default besides the kernel widths for LIME. In practice, we set $\sigma = 0.05$ to ensure that perturbations happen in the local region around the prediction, K to $\text{floor}(\frac{d}{2})$, and $N = 10{,}000$ to sample sufficiently. One complication arises for categorical features, where we cannot apply Gaussian perturbations. For these features, we randomly sample these features from a value in the dataset column 30% of the time to guarantee that the feature remains categorical under perturbation. Last, if multiple explanations return similar fidelities, we use the explanation stability metric proposed in ref. 75 to break ties, because it is much more desirable for the explanation to robust to perturbations[7,76]. To use the stability metric proposed in ref. 75 to break ties if the explanations fidelities are quite close (less than $\delta = 0.01$), we compute the Jaccard similarity between feature rankings instead of the L2 norm as is used in their work. The reason is that the norm might not be comparable between explanation types, because they have different ranges, while the Jaccard similarity should not be affected. Further, we compute the area under the top-k curve using the Jaccard similarity stability metric, as in equation (3), to make this measure more robust.

Additional explanation types. As users will have explainability questions that cannot be answered solely with feature importance explanations, we include additional explanations to support a wider array of conversation topics. In particular, we support counterfactual explanations and feature interaction effects. These methods enable conversations about how to get different outcomes and whether features interact with each other during predictions, supporting a broad set of user queries. We implement counterfactual explanations using diverse counterfactual explanations, which generates a diverse set of counterfactuals[77]. Having access to many plausible counterfactuals is desirable because it enables users to see a breadth of different, potentially useful, options. Also, we implement feature interaction effects using the partial dependence based approach from ref. 78 because it is effective and quick to compute.

Exploring data and predictions. Because the process of understanding models often requires users to inspect the model's predictions, errors and the data, TalkToModel supports a wide variety of data and model exploration tools. For example, TalkToModel provides options for filtering data and performing what-if analyses, supporting user queries that concern subsets of data or what would happen if data points change. Users can also inspect model errors, predictions, prediction probabilities, compute summary statistics, and evaluation metrics for individuals and groups of instances. TalkToModel additionally supports summarizing common patterns in mistakes on groups of instances by training a shallow decision tree on the model errors in the group. Also, TalkToModel enables descriptive operations, which explain how the system works, summarize the dataset and define terms to help users understand how to approach the conversation. Overall, TalkToModel supports a rich set of conversation topics in addition to explanations, making the system a complete solution for the model understanding requirements of end users.

Extensibility

While we implement TalkToModel with several different choices for operations such as feature importance explanations and counterfactual explanations, TalkToModel is highly modular and system designers can easily incorporate new operations or change existing ones by modifying the grammar to best support their user populations. This design makes TalkToModel straightforward to extend to new settings, where different operations may be desired.

Broader impact statement. The TalkToModel system and, more generally, conversational model explainability can be applied to a wide range of applications, including financial, medical or legal applications. Our research could be used to improve model understanding in these situations by improving transparency and encouraging the positive impact of ML systems, while reducing errors and bias. Although TalkToModel has many positive applications, the system makes it easier for those without high levels of technical expertise to understand ML models, which could lead to a false sense of trust in ML systems. In addition, because TalkToModel makes it easier to use ML model for those with lower levels of expertise, there is additionally a risk of inexperienced users applying ML models inappropriately. While TalkToModel includes several measures to prevent such risks, such as qualifying when explanations or predictions are inaccurate, and clearly describing the intended purpose of the ML model, it would be useful for researchers to investigate and possible adopters to be mindful of these considerations. While completing this research, the authors complied with all relevant ethical regulations of human research.

Reporting summary

Further information on research design is available in the Nature Portfolio Reporting Summary linked to this article.

Data availability

The German credit, COMPAS and diabetes datasets and models can be found at https://github.com/dylan-slack/TalkToModel/tree/main/data. The fine-tuned language models used for TalkToModel for each of these datasets can be found at https://huggingface.co/dslack/all-finetuned-ttm-models. The mturk-generated dataset used to assess parsing accuracy and the accuracy results can be found at https://github.com/dylan-slack/TalkToModel/tree/main/experiments/parsing_accuracy. The user study response data are provided at https://github.com/dylan-slack/TalkToModel/blob/main/data/ttm-user-study-responses.csv.

Code availability

We release an open-source implementation of TalkToModel at https://github.com/dylan-slack/TalkToModel[79]. Beyond the methods described so far, this release includes visualizations for conversations, interactive tooling to help users construct questions, saving results and conversation environments so they can be shared, abstractions for creating new operations and synthetic datasets, routines to adapt TalkToModel to new models and datasets automatically, and runtime optimizations (generating responses typically takes <2 seconds).

References

1. Lakkaraju, H., Bach, S. H. & Leskovec, J. Interpretable decision sets: a joint framework for description and prediction. In *Proc. 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* 1675–1684 (Association for Computing Machinery, 2016).
2. Angelino, E., Larus-Stone, N., Alabi, D., Seltzer, M. & Rudin, C. Learning certifiably optimal rule lists. In *Proc. 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* 35–44 (Association for Computing Machinery, 2017).
3. Lou, Y., Caruana, R., Gehrke, J. & Hooker, G. Accurate intelligible models with pairwise interactions. In *Proc. 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (eds Ghani, R. et al.) 623–631 (Association for Computing Machinery, 2013).
4. Agarwal, R. et al. Neural additive models: interpretable machine learning with neural nets. *Adv. Neural Inf. Process. Syst.* **34**, 4699–4711 (2021).
5. Chang, C.-H., Caruana, R. & Goldenberg, A. Node-GAM: neural generalized additive model for interpretable deep learning. In *International Conference on Learning Representations* (2022).
6. Ribeiro, M. T., Singh, S. & Guestrin, C. Model-agnostic interpretability of machine learning. In *ICML Workshop on Human Interpretability in Machine Learning* (2016).
7. Slack, D., Hilgard, A., Singh, S. & Lakkaraju, H. Reliable post hoc explanations: modeling uncertainty in explainability. *Adv. Neural Inf. Process. Syst.* **34**, 9391–9404 (2021).
8. Selvaraju, R. R. et al. Grad-CAM: visual explanations from deep networks via gradient-based localization. In *2017 IEEE International Conference on Computer Vision* 618–626 (IEEE, 2017).
9. Slack, D., Rauschmayr, N. & Kenthapadi, K. Defuse: training more robust models through creation and correction of novel model errors. In *NeurIPS 2021 Workshop on Explainable AI Approaches for Debugging and Diagnosis* (2021).
10. Hase, P., Xie, H. & Bansal, M. The out-of-distribution problem in explainability and search methods for feature importance explanations. *Adv. Neural Inf. Process. Syst.* **34**, 3650–3666 (2021).
11. Simonyan, K., Vedaldi, A. & Zisserman, A. Deep inside convolutional networks: visualising image classification models and saliency maps. In *Workshop at International Conference on Learning Representations* (2014).
12. Lakkaraju, H., Slack, D., Chen, Y., Tan, C. & Sing, S. Rethinkingexplainability as a dialogue: a practitioner's perspective. *HAI Workshop @ NeurIPS* (2022).
13. Kaur, H. et al. Interpreting interpretability: understanding data scientists' use of interpretability tools for machine learning. In *Proc. 2020 CHI Conference on Human Factors in Computing Systems* 1–14 (Association for Computing Machinery, 2020).
14. Weld, D. S. & Bansal, G. The challenge of crafting intelligible intelligence. *Commun. ACM* **62**, 70–79 (2019).
15. Fok, R. & Weld, D. S. In search of verifiability: explanations rarely enable complementary performance in AI-advised decision making. Preprint at https://arxiv.org/abs/2305.07722 (2023).

16. Tenney, I. et al. The language interpretability tool: extensible, interactive visualizations and analysis for NLP models. In *Proc. 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations* (eds Liu, Q. & Schlangen, D.) 107–118 (Association for Computational Linguistics, 2020).
17. Wexler, J. et al. The what-if tool: interactive probing of machine learning models. *IEEE Trans. Vis. Comput. Graph.* **26**, 56–65 (2020).
18. Ward, N. G. & DeVault, D. Ten challenges in highly-interactive dialog systems. In *AAAI Conference on Artificial Intelligence* (2015).
19. Carenini, G., Mittal, V. O. & Moore, J. D. Generating patient-specific interactive natural language explanations. In *Proc. Annual Symposium on Computer Applications in Medical Care* 5–9 (1994).
20. Pennebaker, J. W., Mehl, M. R. & Niederhoffer, K. G. Psychological aspects of natural language use: our words, our selves. *Annu. Rev. Psychol.* **54**, 547–577 (2002).
21. Zhang, Z., Takanobu, R., Zhu, Q., Huang, M. & Zhu, X. Recent advances and challenges in task-oriented dialog systems. *Sci. China Technol. Sci.* **63**, 2011–2027 (2020).
22. Sokol, K. & Flach, P. Glass-box: explaining AI decisions with counterfactual statements through conversation with a voice-enabled virtual assistant. In *Proc. 27th International Joint Conference on Artificial Intelligence* (ed. Lang, J.) 5868–5870 (IJCAI, 2018).
23. Feldhus, N., Ravichandran, A. M. & Möller, S. Mediators: conversational agents explaining NLP model behavior. *IJCAI-ECAI Workshop on Explainable Artificial Intelligence (XAI)* (2022).
24. Sutskever, I., Vinyals, O. & Le, Q. V. Sequence to sequence learning with neural networks. In *Proc. 27th International Conference on Neural Information Processing Systems* Vol. 2 (eds Ghahramani, Z. et al.) 3104–3112 (MIT Press, 2014).
25. Yu, T. et al. Spider: a large-scale human-labeled dataset for omplex and cross-domain semantic parsing and text-to-SQL task. In *Proc. 2018 Conference on Empirical Methods in Natural Language Processing* (eds Riloff, E. et al.) 3911–3921 (Association for Computational Linguistics, 2018).
26. Dua, D. & Graff, C. *UCI Machine Learning Repository* (UCI, 2017); http://archive.ics.uci.edu/ml
27. Angwin, J., Larson, J., Mattu, S. & Kirchner, L. Machine bias. *ProPublica* (2016).
28. Wang, B. & Komatsuzaki, A. GPT-J-6B: a 6 billion parameter autoregressive language model. *GitHub* https://github.com/kingoflolz/mesh-transformer-jax (2021).
29. Brown, T. et al. Language models are few-shot learners. *Adv. Neural Inf. Process. Syst.* **33**, 1877–1901 (2020).
30. Raffel, C. et al. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.* **21**, 5485–5551 (2020).
31. Min, S. et al. Rethinking the role of demonstrations: what makes in-context learning work? In *Proc. 2022 Conference on Empirical Methods in Natural Language Processing* 11048–11064 (Association for Computational Linguistics, 2022).
32. Xie, S. M., Raghunathan, A., Liang, P. & Ma, T. An explanation of in-context learning as implicit bayesian inference. In *International Conference on Learning Representations* (2022).
33. Reimers, N. & Gurevych, I. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proc. 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing* (eds Padó, S. & Huang, R) 3982–3992 (Association for Computational Linguistics, 2019).
34. Shin, R. et al. Constrained language models yield few-shot semantic parsers. In *Proc. 2021 Conference on Empirical Methods in Natural Language Processing* (eds Moens, M.-F. et al.) 7699–7715 (Association for Computational Linguistics, 2021).
35. Talmor, A., Geva, M. & Berant, J. Evaluating semantic parsing against a simple web-based question answering model. In *Proc. 6th Joint Conference on Lexical and Computational Semantics* (eds Ide, N. et al.) 161–167 (Association for Computational Linguistics, 2017).
36. Gupta, S., Singh, S. & Gardner, M. Structurally diverse sampling for sample-efficient training and comprehensive evaluation. In *Findings of the Association for Computational Linguistics: EMNLP 2022* (eds Goldberg, Y. et al.) 4966–4979 (Association for Computational Linguistics, 2022).
37. Oren, I., Herzig, J., Gupta, N., Gardner, M. & Berant, J. Improving compositional generalization in semantic parsing. In *Findings of the Association for Computational Linguistics: EMNLP 2020* (eds Cohn, T. et al.) 2482–2495 (Association for Computational Linguistics, 2020).
38. Yin, P. et al. Compositional generalization for neural semantic parsing via span-level supervised attention. In *Proc. 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (eds Toutanova, K. et al.) 2810–2823 (Association for Computational Linguistics, 2021).
39. Dijk, O. et al. oegedijk/explainerdashboard: v0.3.8.2: reverses set_shap_values bug introduced in 0.3.8.1. *Zenodo* https://doi.org/10.5281/zenodo.6408776 (2022).
40. Pedregosa, F. et al. Scikit-learn: machine learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011).
41. Chen, Q., Schnabel, T., Nushi, B. & Amershi, S. Hint: integration testing for AI-based features with humans in the loop. In *27th International Conference on Intelligent User Interfaces* 549–565 (ACM, 2022).
42. Freed, M. et al. RADAR: a personal assistant that learns to reduce email overload. In *Proc. 23rd National Conference on Artificial Intelligence* Vol. 3 (ed. Cohn, A.) 1287–1293 (AAAI Press, 2008).
43. Glass, A., McGuinness, D. L. & Wolverton, M. Toward establishing trust in adaptive agents. In *Proc. 13th International Conference on Intelligent User Interfaces* 227–236 (Association for Computing Machinery, 2008).
44. Palan, S. & Schitter, C. Prolific.ac—a subject pool for online experiments. *J. Behav. Exp. Finance* **17**, 22–27 (2018).
45. Chen, H., Liu, X., Yin, D. & Tang, J. A survey on dialogue systems: recent advances and new frontiers. *SIGKDD Explor. Newsl.* **19**, 25–35 (2017).
46. Li, X., Chen, Y.-N., Li, L., Gao, J. & Celikyilmaz, A. End-to-end task-completion neural dialogue systems. In *Proc. Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)* (eds Kondrak, G. & Watanabe, T.) 733–743 (2017).
47. Dong, C. et al. A survey of natural language generation. *ACM Comput. Surv.* **55**, 1–38 (2022).
48. Liu, Y., Han, K., Tan, Z. & Lei, Y. Using context information for dialog act classification in DNN framework. In *Proc. 2017 Conference on Empirical Methods in Natural Language Processing* (eds Palmer, M. et al.) 2170–2178 (Association for Computational Linguistics, 2017).
49. Cai, W. & Chen, L. Predicting user intents and satisfaction with dialogue-based conversational recommendations. In *Proc. 28th ACM Conference on User Modeling, Adaptation and Personalization* (eds Kuflik, T. et al.) 33–42 (Association for Computing Machinery, 2020).
50. Liao, Q. V., Gruen, D. & Miller, S. Questioning the AI: informing design practices for explainable AI user experiences. In *Proc. 2020 CHI Conference on Human Factors in Computing Systems* 1–15 (Association for Computing Machinery, 2020).

51. Grosz, B. J., Joshi, A. K. & Weinstein, S. Providing a unified account of definite noun phrases in discourse. In *21st Annual Meeting of the Association for Computational Linguistics* 44–50 (Association for Computational Linguistics, 1983).
52. Tseng, B.-H. et al. CREAD: combined resolution of ellipses and anaphora in dialogues. In *Proc. 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (eds Toutanova, K. et al.) 3390–3406 (Association for Computational Linguistics, 2021).
53. Guo, D., Tang, D., Duan, N., Zhou, M. & Yin, J. Dialog-to-action: conversational question answering over a large-scale knowledge base. In *Proc. 32nd International Conference on Neural Information Processing Systems* (eds Bengio, S. et al.) 2946–2955 (Curran Associates Inc., 2018).
54. Gao, S., Sethi, A., Agarwal, S., Chung, T. & Hakkani-Tur, D. Dialog state tracking: s neural reading comprehension approach. In *Proc. 20th Annual SIGdial Meeting on Discourse and Dialogue* (eds Nakamura, S. et al.) 264–273 (Association for Computational Linguistics, 2019).
55. Gao, J., Galley, M. & Li, L. Neural approaches to conversational AI. In *Proc. 56th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts* (eds Artzi, Y. & Eisenstein, J.) 2–7 (Association for Computational Linguistics, 2018).
56. Rieser, V. & Lemon, O. in *Data-Driven Methods for Adaptive Spoken Dialogue Systems* (eds Lemon, O. & Pietquin, O.) 5–17 (Springer, 2012).
57. Zhao, Z., Wallace, E., Feng, S., Klein, D. & Singh, S. Calibrate before use: improving few-shot performance of language models. In *Proc. 38th International Conference on Machine Learning* (eds Meila, M. & Zhang, T.) 12697-12706 (PMLR, 2021).
58. Loshchilov, I. & Hutter, F. Decoupled weight decay regularization. In *International Conference on Learning Representations* (2019).
59. Shao, Y. et al. Generating high-quality and informative conversation responses with sequence-to-equence models. In *Proc. 2017 Conference on Empirical Methods in Natural Language Processing* (eds Palmer, M. et al.) 2210–2219 (Association for Computational Linguistics, 2017).
60. Smilkov, D., Thorat, N., Kim, B., Viégas, F. & Wattenberg, M. Smoothgrad: removing noise by adding noise. In *Workshop on Visualization for Deep Learning* (2017).
61. Yeh, C.-K., Hsieh, C.-Y., Suggala, A., Inouye, D., & Ravikumar, P. On the (In)fidelity and sensitivity of explanations. In *Proc. 33rd International Conference on Neural Information Processing Systems* (eds Wallach, H. M. et al.) 10967–10978 (Curran Associates, Inc. 2019).
62. Chen, J., Song, L., Wainwright, M. J. & Jordan, M. I. L-Shapley and c-Shapley: efficient model interpretation for structured data. In *International Conference on Learning Representations* (2019).
63. Agarwal, S. et al. Towards the unification and robustness of perturbation and gradient-based explanations. In *Proc. 38th International Conference on Machine Learning* (eds Meila, M. & Zhang, T.) 110–119 (PMLR, 2021).
64. Ribeiro, M. T., Singh, S. & Guestrin, C. 2016. "Why should I trust you?": explaining the predictions of any classifier. In *Proc. 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* 1135–1144 (Association for Computing Machinery, 2016).
65. Lundberg, S. M. et al. From local explanations to global understanding with explainable AI for trees. *Nat. Mach. Intell.* **2**, 56–67 (2020).
66. Lakkaraju, H., Kamar, E., Caruana, R. & Leskovec, J. Faithful and customizable explanations of black box models. In *Proc. 2019 AAAI/ACM Conference on AI, Ethics, and Society* 131–138 (Association for Computing Machinery, 2019).
67. Plumb, G., Molitor, D. & Talwalkar, A. Model agnostic supervised local explanations. In *Proc. 32nd International Conference on Neural Information Processing Systems* (eds Bengio, S. et al.) 2520–2529 (Curran Associates, 2018).
68. Li, J., Nagarajan, V., Plumb, G. & Talwalkar, A. A learning theoretic perspective on local explainability. In *International Conference on Learning Representations* (2020).
69. Murdoch, W. J., Singh, C., Kumbier, K., Abbasi-Asl, R. & Yu, B. Definitions, methods, and applications in interpretable machine learning. *Proc. Natl Acad. Sci. USA* **116**, 22071–22080 (2019).
70. Sundararajan, M., Taly, A. & Yan, Q. Axiomatic attribution for deep networks. In *Proc. 34th International Conference on Machine Learning* Vol. 70 (eds Precup, D. & Teh, Y. W.) 3319–3328 (JMLR.org, 2017).
71. Krishna, S. et al. The disagreement problem in explainable machine learning: a practitioner's perspective. *ICML Workshop on Interpretable Machine Learning in Healthcare* (2022).
72. Meng, C., Trinh, L., Xu, N., Enouen, J. & Liu, Y. Interpretability and fairness evaluation of deep learning models on MIMIC-IV dataset. *Sci. Rep.* **12**, 7166 (2022).
73. Hooker, S., Erhan, D., Kindermans, P.-J. & Kim, B. *A Benchmark for Interpretability Methods in Deep Neural Networks* (Curran Associates, 2019).
74. Scott M. Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Proc. 31st International Conference on Neural Information Processing Systems* (eds von Luxburg, U. et al.) 4768–4777 (Curran Associates Inc., 2017).
75. Alvarez-Melis, D. & Jaakkola, T. S. On the robustness of interpretability methods. *ICML Workshop on Human Interpretability in Machine Learning* (2018).
76. Agarwal, C. et al. Rethinking stability for attribution-based explanations. *ICLR Pair2Struct Workshop* (2022).
77. Mothilal, R. K., Sharma, A. & Tan, C. Explaining machine learning classifiers through diverse counterfactual explanations. In *Proc. 2020 Conference on Fairness, Accountability, and Transparency* 607–617 (Association for Computing Machinery, 2020).
78. Greenwell, B. M., Boehmke, B. C. & McCarthy, A. J. A simple and effective model-based variable importance measure. Preprint at https://arxiv.org/abs/1805.04755 (2018).
79. Slack, D., Krishna, S., Lakkaraju, H. & Singh, S. TalkToModel: explaining machine learning models with interactive natural language conversations. *Zenodo* https://doi.org/10.5281/zenodo.7502206 (2022).

Acknowledgements

We acknowledge helpful feedback from P. Hase, J. Ugander, M. T. Ribeiro, B. Lim and the UCI NLP lab concerning earlier versions of the system and papers. This work is supported in part by the NSF awards IIS-2008461 to H.L., IIS-2040989 to S.S. and H.L., IIS-2046873 to S.S. and IIS-2008956 to S.S., and research awards from Google, JP Morgan, Amazon, Harvard Data Science Initiative, the D^3 Institute at Harvard and the Hasso Plattner Institute. H.L. thanks S. and M. Lakkaraju for their continued support and encouragement. The views expressed here are those of the authors and do not reflect the official policy or position of the funding agencies.

Author contributions

D.S. designed and developed the TalkToModel system and studies, and prepared the paper. S.K. designed and implemented the explanation selection procedure, drafted sections in the paper and edited the paper. S.S. and H.L. contributed equally to advising the development of the system and experiments, conceiving the system, reviewing the paper and editing the paper.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s42256-023-00692-8.

Correspondence and requests for materials should be addressed to Dylan Slack.

Peer review information *Nature Machine Intelligence* thanks Mengnan Du, Shafiq Rayhan Joty and Sherin Mathews for their contribution to the peer review of this work.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

nature portfolio

Corresponding author(s): Dylan Slack

Last updated by author(s): May 26, 2023

Reporting Summary

Nature Portfolio wishes to improve the reproducibility of the work that we publish. This form provides structure for consistency and transparency in reporting. For further information on Nature Portfolio policies, see our Editorial Policies and the Editorial Policy Checklist.

Statistics

For all statistical analyses, confirm that the following items are present in the figure legend, table legend, main text, or Methods section.

n/a	Confirmed	
□	☒	The exact sample size (*n*) for each experimental group/condition, given as a discrete number and unit of measurement
□	☒	A statement on whether measurements were taken from distinct samples or whether the same sample was measured repeatedly
□	☒	The statistical test(s) used AND whether they are one- or two-sided *Only common tests should be described solely by name; describe more complex techniques in the Methods section.*
□	☒	A description of all covariates tested
☒	□	A description of any assumptions or corrections, such as tests of normality and adjustment for multiple comparisons
□	☒	A full description of the statistical parameters including central tendency (e.g. means) or other basic estimates (e.g. regression coefficient) AND variation (e.g. standard deviation) or associated estimates of uncertainty (e.g. confidence intervals)
☒	□	For null hypothesis testing, the test statistic (e.g. *F*, *t*, *r*) with confidence intervals, effect sizes, degrees of freedom and *P* value noted *Give P values as exact values whenever suitable.*
☒	□	For Bayesian analysis, information on the choice of priors and Markov chain Monte Carlo settings
☒	□	For hierarchical and complex designs, identification of the appropriate level for tests and full reporting of outcomes
☒	□	Estimates of effect sizes (e.g. Cohen's *d*, Pearson's *r*), indicating how they were calculated

Our web collection on statistics for biologists contains articles on many of the points above.

Software and code

Policy information about availability of computer code

Data collection	The TalkToModel system was used to collect data within the study. The TalkToModel system is provided in our open source package: https://github.com/dylan-slack/TalkToModel. TalkToModel relies on several software packages to produce results. These include dice-ml version 0.7.2, lime version 0.2.0.1, numpy version 1.21.6, pandas version 1.4.1, scikit-learn version 1.0.2, scipy version 1.8.0, shap version 0.40.0, and transformers version 4.17.0.
Data analysis	We used numpy version 1.21.6 and pandas version 1.4.1 and matplotlib version 3.5.1 to analyze the data in the study.

For manuscripts utilizing custom algorithms or software that are central to the research but not yet described in published literature, software must be made available to editors and reviewers. We strongly encourage code deposition in a community repository (e.g. GitHub). See the Nature Portfolio guidelines for submitting code & software for further information.

Data

Policy information about availability of data

All manuscripts must include a data availability statement. This statement should provide the following information, where applicable:

- Accession codes, unique identifiers, or web links for publicly available datasets
- A description of any restrictions on data availability
- For clinical datasets or third party data, please ensure that the statement adheres to our policy

The German, Compas, and Diabetes datasets and models can be found at https://github.com/dylan-slack/TalkToModel/tree/main/data. The finetuned language models used for TalkToModel for each of these datasets can be found at https://huggingface.co/dslack/all-finetuned-ttm-models. The mturk generated dataset used to assess parsing accuracy and the accuracy results can be found at https://github.com/dylan-slack/TalkToModel/tree/main/experiments/parsing_accuracy. The user study response data is provided at https://github.com/dylan-slack/TalkToModel/blob/main/data/ttm-user-study-responses.csv.

Human research participants

Policy information about studies involving human research participants and Sex and Gender in Research.

Reporting on sex and gender	Sex and gender were not considered in the study design. The healthcare worker group sex information was split with 47% female 53% male participants (N=45). This information was self reported. We did not collect sex and gender information for the machine learning graduate student and professional group.
Population characteristics	See above.
Recruitment	We recruited the group of healthcare workers from the Prolific research service. Prolific provides an online interface to recruit high quality participants for user studies. Prolific enables us to filter participants by different characteristics, such as population characteristics and industry of occupation. Because we were interested in recruiting a group with healthcare experience, we included English speaking healthcare workers in our study. We informed the group of participants fitting our criteria provided by prolific the study would take around 30 minutes (we gauged this time based on pilot studies) and offered them $7.50 ($15/hour) to complete the study. We accepted the responses of the first 45 participants to complete the study. Note, participants were not allowed to begin the study if sufficient participants were already taking the study, to ensure work would not go uncompensated. We recruited the graduate and machine learning professional group from from machine learning Slack groups and email lists by advertising a paid study related to evaluating machine learning tools. Because individuals self report whether they are machine learning professionals, they may overestimate the extent of their expertise in machine learning, resulting in self-selection biases. For instance, there could be computer scientists with only a small amount of experience in machine learning that participate in the machine learning graduate student and professional group. This self-selection could impact the comparison between groups with differing levels of machine learning experience, because some individuals overestimate the scope of their expertise in machine learning. To try and ensure participants did not unduly self-select into the machine learning professional group, we specifically asked participants for the extent of their experience in machine learning by asking an equivalent in terms of educational attainment and planned to remove participants without graduate experience or higher (e.g., no experience/read articles online/undergraduate course/graduate course/published machine learning papers). We found our participants all stated they had graduate course experience or higher. Along the same lines, participants in the healthcare group may self-select into this group with occupations only tangentially related to healthcare, which could reduce the effects of the comparisons with this group. To help ensure that participants work in health care and control for self-selection biases in this regard, we asked them to state their occupation in the survey and found participants all had occupations directly related to healthcare (doctors, nurses, pharmacists, etc.). In general, we expect the measures we took in our study to help reduce adverse effects due to self-selection biases.
Ethics oversight	We received approval from the University of California, Irvine Institutional Review Board.

Note that full information on the approval of the study protocol must also be provided in the manuscript.

Field-specific reporting

Please select the one below that is the best fit for your research. If you are not sure, read the appropriate sections before making your selection.

□ Life sciences ☒ Behavioural & social sciences □ Ecological, evolutionary & environmental sciences

For a reference copy of the document with all sections, see nature.com/documents/nr-reporting-summary-flat.pdf

Behavioural & social sciences study design

All studies must disclose on these points even when the disclosure is negative.

Study description

The user study presented in the work includes mixed methods data, where we collected both quantitative metrics concerning accuracy and time taken solving a series of model understanding tasks with talktomodel compared to a dashboard style system. The specific metrics were the accuracy on the tasks (each user solved 10 tasks) and time taken per task (recorded as the time between question open and close). We additionally collected likert scores for several perceived quantitative metrics for each tasks, such as how confident they were in their answer, effort, usefulness of the interface, trust in the system, and likeliness to use again [1]. Each user rated these 5 metrics for each of the 10 questions. We additionally collected these same liker metrics at the end of the study, asking users to compare their preference between the two systems (e.g.,, whether they agree TalkToModel performed better than the dashboard, with respect to the likert metric). Finally, we collected qualitative feedback, where asked users to write a short description comparing their experience using both systems.

[1] Quanze Chen, Tobias Schnabel, Besmira Nushi, and Saleema Amershi. Hint: Integration testing
for ai-based features with humans in the loop. In International Conference on Intelligent User Interfaces. ACM, March 2022

Research sample

The research sample is a group of healthcare workers (N=45), consisting of 47% Female 53% Male, 20% Black, 29% White, 27% Mixed 14% Other, average 26.5 years old (min 20, max 67). This sample is not guaranteed to be representative. The other group recruited in the study is machine learning graduate students and professionals. We did not collect covariate information for this group. This group is not guaranteed to be representative. These two groups were appropriate for our study because we wished to analyze the effectiveness of TalkToModel for groups with varying skills and expertise in machine learning.

Sampling strategy

The populations were sampled with convenience sampling. We determined the sample sizes need to achieve statistical significance by following prior related studies within machine learning and natural language processing communities that evaluate the performance of machine learning systems with end users and that achieved statistical significance. We used similar sample sizes with our human evaluation studies to achieve statistical significance [1, 2, 3]. For instance, our study with graduate students had 13 participants, where the related works [1] had 18 participants and [2] had 13 participants. Similarly, we used 45 participants within the health care worker group, which is again comparable to prior works on studying the machine learning systems with end users, e.g., [3] which uses 31 participants. Because TalkToModel shares many similar characteristics with these prior works in machine learning and natural language processing that achieved statistical significance (e.g., evaluating performance of a machine learning system with end users such as machine learning graduate students), using comparable sample sizes to those presented in the related literature is an appropriate choice for our work.

[1] Beyond Accuracy: Behavioral Testing of NLP models with CheckList. Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, Sameer Singh Association for Computational Linguistics (ACL), 2020
[2] Quanze Chen, Tobias Schnabel, Besmira Nushi, and Saleema Amershi. Hint: Integration testing
for ai-based features with humans in the loop. In International Conference on Intelligent User Interfaces. ACM, March 2022
[3] Dylan Slack, Sophie Hilgard, Sameer Singh, Himabindu Lakkaraju. Reliable Post hoc Explanations: Modeling Uncertainty in Explainability. NeurIPS, 2021.

Data collection

We sent out an online survey using Qualtrics. Results were stored in a qualtrics database. We did not observe the participants taking the study. The researchers were blinded to the experimental conditions (the tasks and order of the machine learning systems given to the study participants were randomized). The study hypothesis was not blind to the researchers.

Timing

We started collection on June 12th, 2022 and closed collection on June 23rd, 2022.

Data exclusions

We excluded 1 data point for a healthcare worker who failed a sanity check question in our survey. We included sanity check questions as a pre-established exclusion criteria, to ensure we received data from participants who were actively paying attention to the survey.

Non-participation

There were 10 participants who dropped out, either by opening the survey and timing out, or closing the survey tab, having not returned responses.

Randomization

The ordering of the tasks and sections of the survey were randomized.

Reporting for specific materials, systems and methods

We require information from authors about some types of materials, experimental systems and methods used in many studies. Here, indicate whether each material, system or method listed is relevant to your study. If you are not sure if a list item applies to your research, read the appropriate section before selecting a response.

Materials & experimental systems

n/a	Involved in the study
☒	□ Antibodies
☒	□ Eukaryotic cell lines
☒	□ Palaeontology and archaeology
☒	□ Animals and other organisms
☒	□ Clinical data
☒	□ Dual use research of concern

Methods

n/a	Involved in the study
☒	□ ChIP-seq
☒	□ Flow cytometry
☒	□ MRI-based neuroimaging