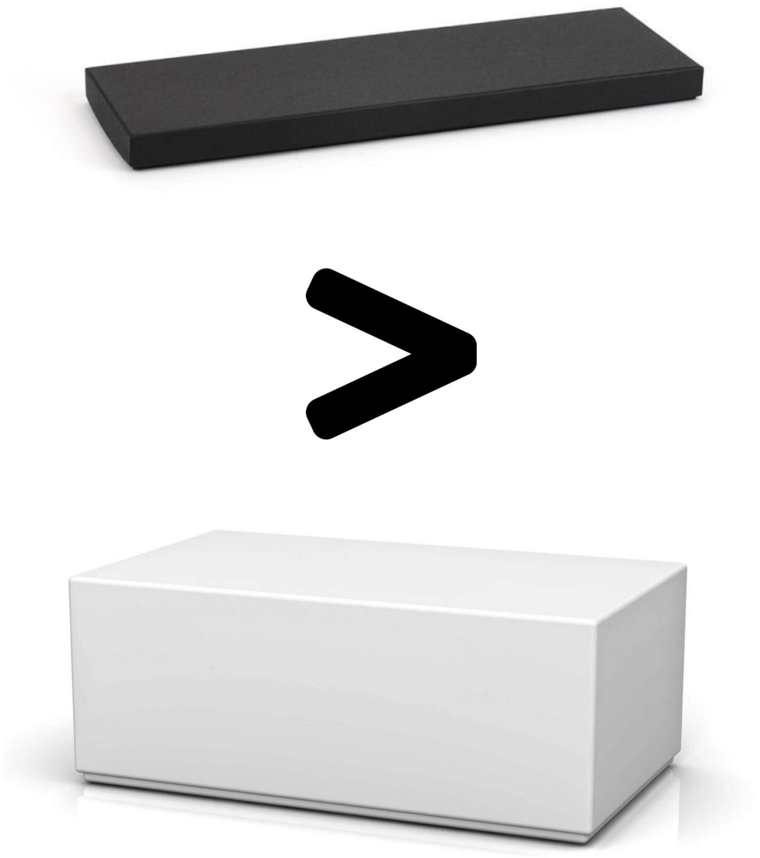
Explaining Machine Learning Models with Interactive Natural Language Conversations Using TalkToModel

Authored by Dylan Slack, Satyapriya Krishna, Himabindu Lakkaraju, and Sameer Singh

Presented by Oam Patel, Jason Wang, and Lucas Monteiro Paes

Motivation

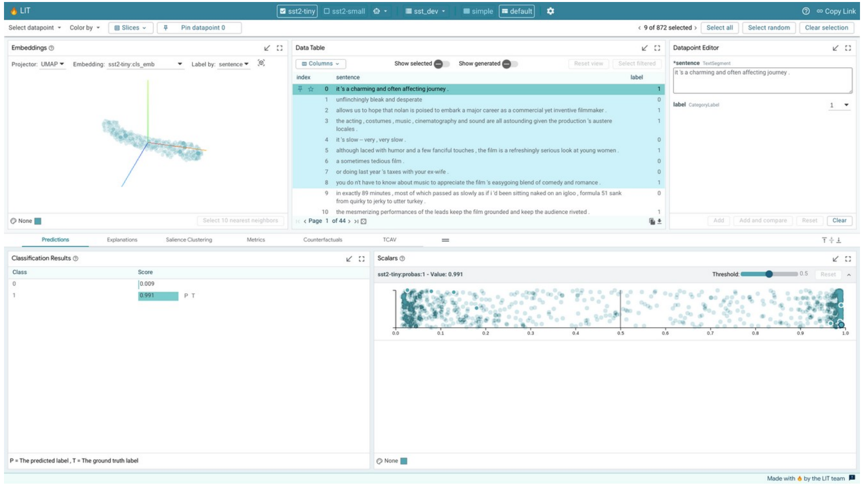
- Simple and intuitive explanations for ML models is a bottleneck to adoption
- Flexibility and accuracy tradeoff for inherently-interpretable models
- Using post-hoc methods is difficult empirically (which explanations, how to interpret, follow-up questions, etc)



Related Work

1. Language-interpretability Tool (LiT)
    - Open-source platform for understanding NLP models
    - Uses local explanations (ala LIME), aggregate data statistics, and counterfactual explanations
1. “What-If” Tool
    - Helps users perform counterfactual analysis for models
1. *explainerdashboard*
    - Used as a baseline for experiments

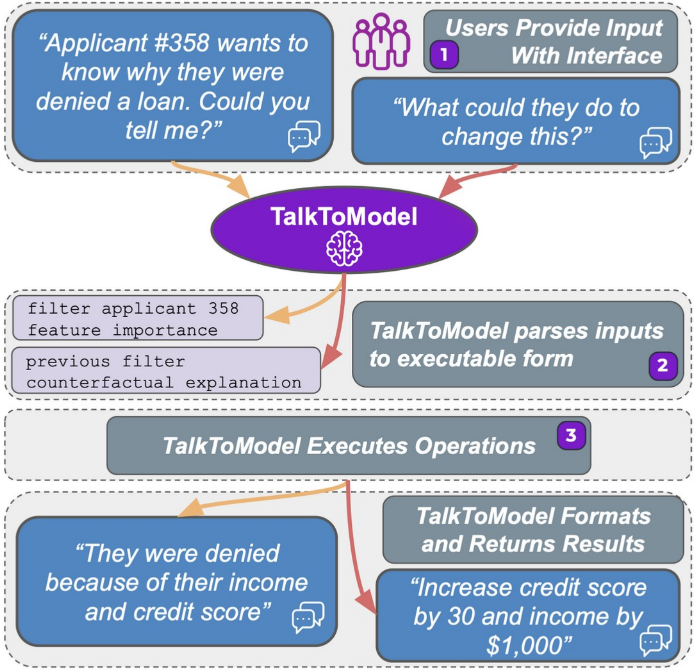
Unfortunately, relatively high barrier to entry and no follow up questions



Problem Statement

- Design a system that makes it easy for lay practitioners to apply post-hoc interpretability methods to black-box models
- Desiderata
  - Dialogue system that can handle many conversation topics (general data trends, questions about specific predictions, etc)
  - Usable for a variety of data types and model classes (i.e. treatment prediction, risk of relapse, interest rate calculations, etc)
  - Doesn’t require a high level of expertise

Summary of Contributions

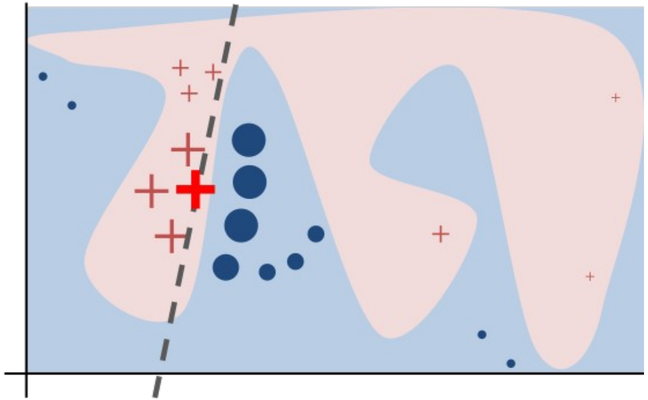


- Introduce TalkToModel which enables open-ended dialogue for understanding a given dataset+classifier pair
  - Why a prediction occurred, how it would change if data changed, how to flip predictions, general statistics regarding the data distribution, etc
- Three parts
  - Dialogue engine (LLM backend)
  - Execution engine (run many explanations, pick the best one)
  - Text interface (to enable conversations)

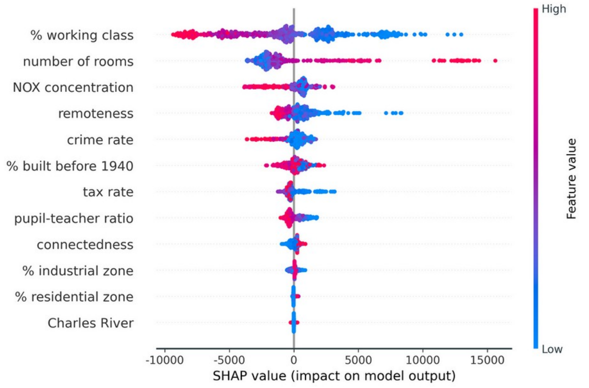
Demo Time!

Method

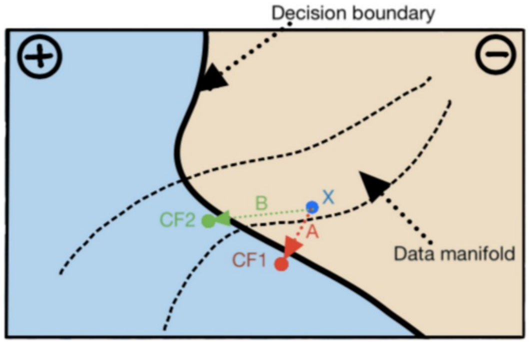
Background



Lime Illustration

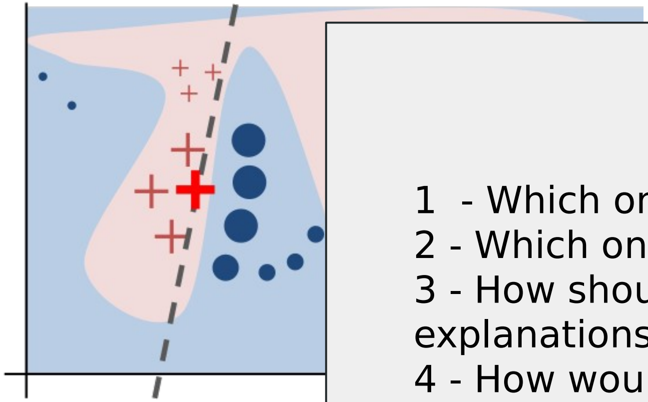


SHAP Illustration

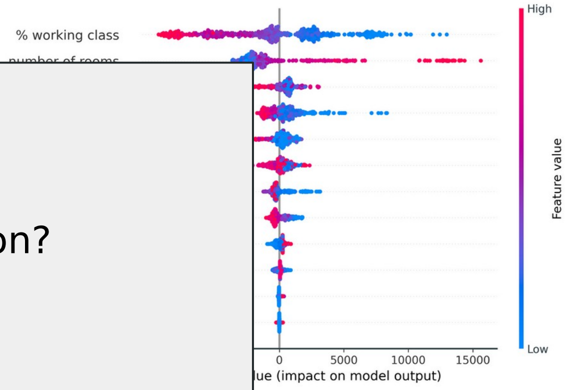


Counterfactual Illustration

Background

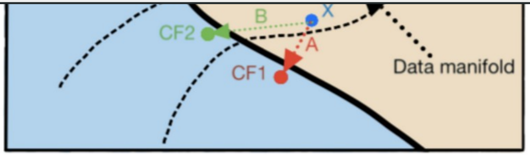


Lime Illu



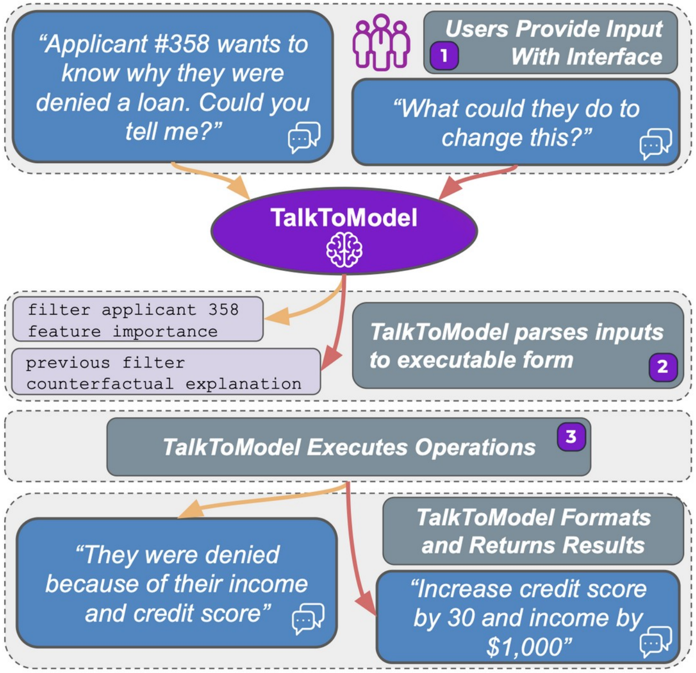
llustration

1 - Which one to choose for my application?
2 - Which one would a nurse\MD trust?
3 - How should a nurse\MD interpret the explanations?
4 - How would a nurse\MD use these methods?
5 - How would a nurse\MD interact with the method?
6 - …



Counterfactual Illustration

Talking to Model



1 - It uses **many** (with the possibility to add more) **post-hoc explanations**!

2 - **It chooses the "best"** explanation to practitioner!

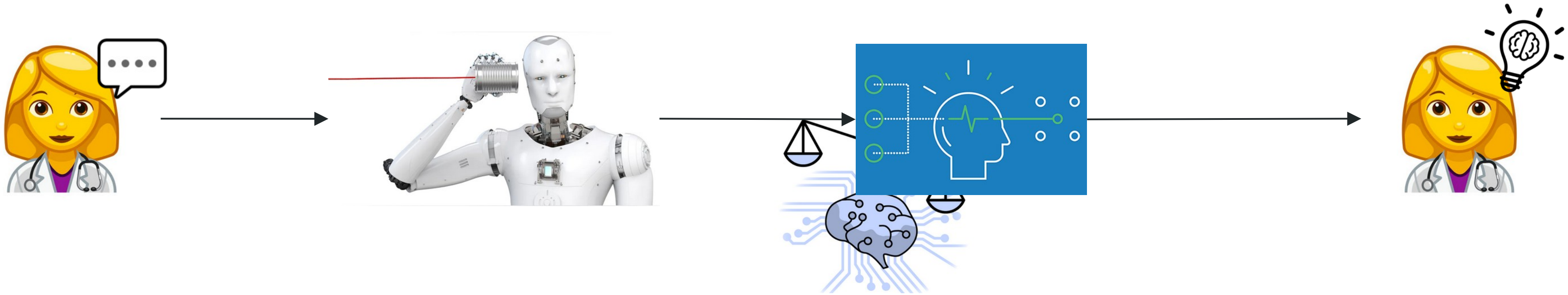
3 - It **answer** user's questions with **natural language**!

4 - Users **only** need to provide the **model and the data!**

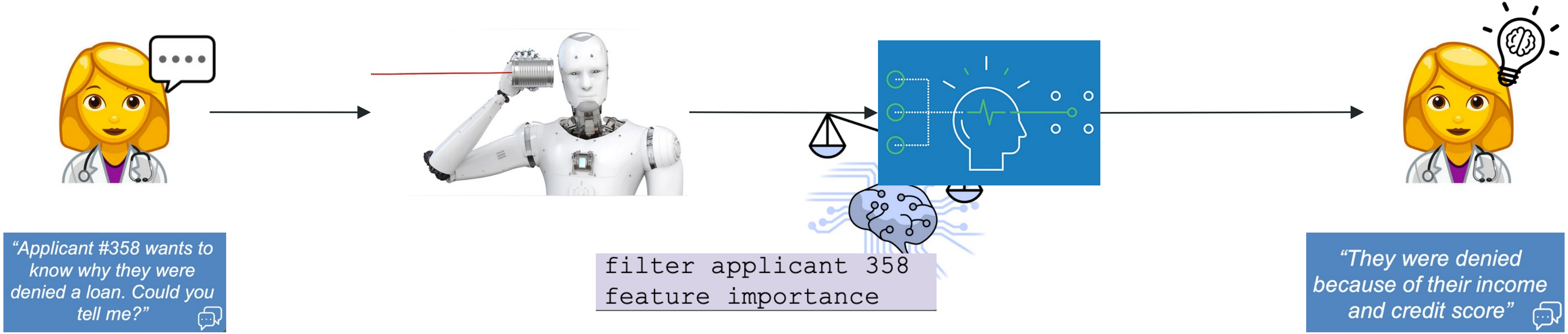
5 - They can **communicate via natural language**!

6 - ...

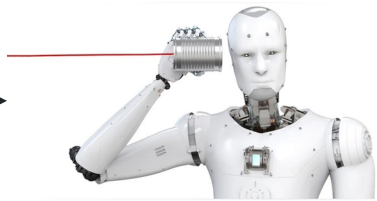
Method’s Structure

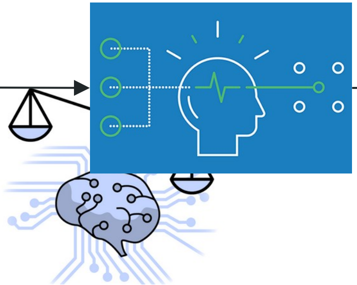


Method’s Structure



Method’s Structure



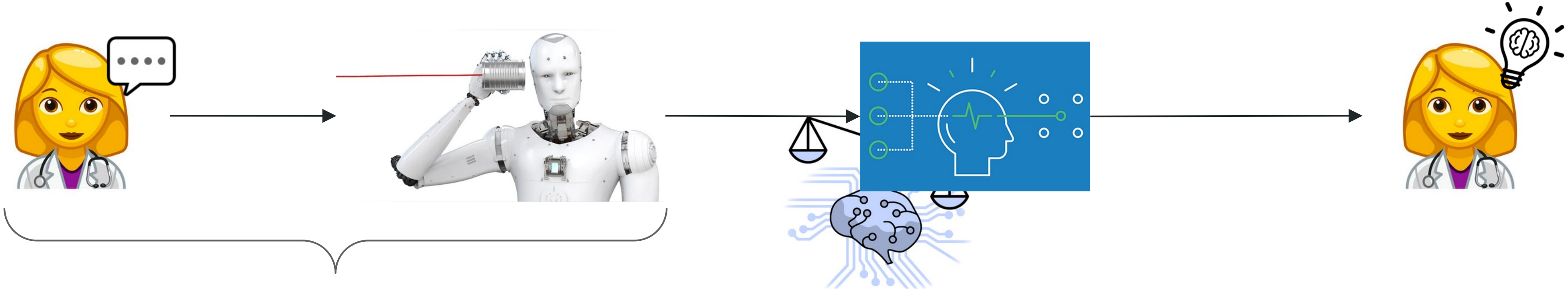


Dialogue engine

Human Query

Structured Instruction s

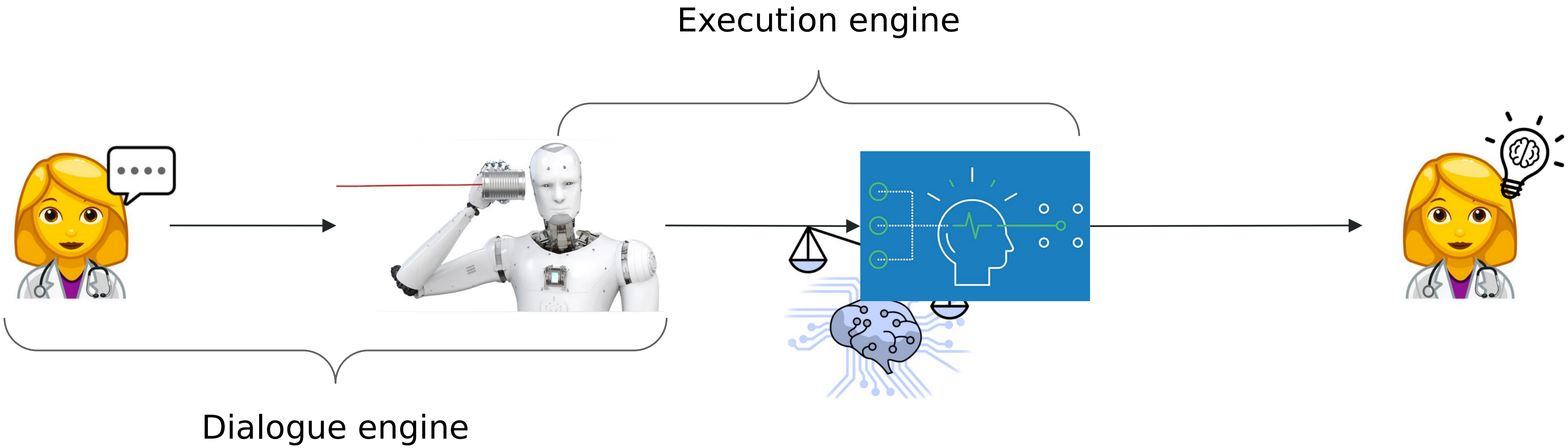
Method’s Structure



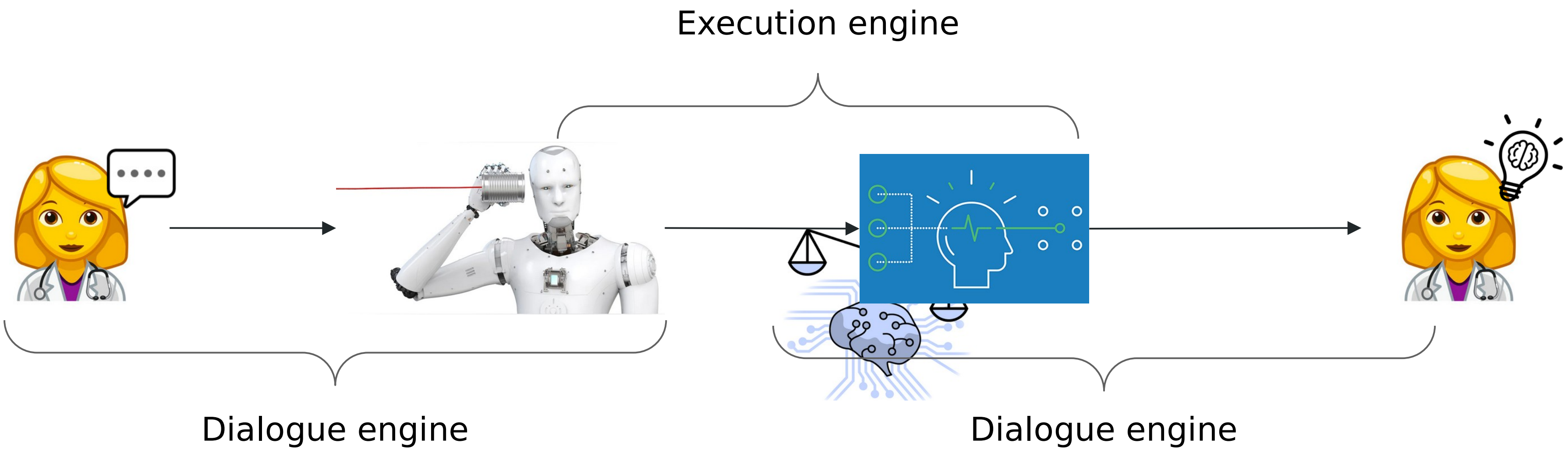
Dialogue engine

Human Query → LLMs → Structured Instructions

Method’s Structure



Method’s Structure



Structured Result $\rightarrow$ Human Query

Dialogue Engine

Constructing a 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.”

Dialogue Engine

Constructing a grammar

How is the grammar generated?

There is a predefined grammar that depends on the production rules that includes:

1 - All the operations that TalkToModel can run,

2 - The arguments for each operation,

3 - The relations between operations!

Dialogue Engine

Constructing a gra

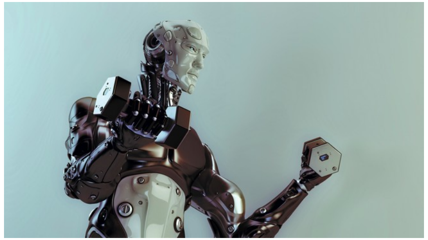
How to define a grammar for different dataset?

It is a challenge to use a general grammar that works for all dataset.

TalkToModel uses a grammar that is dependent on the dataset features!

3 - The relations between operations!

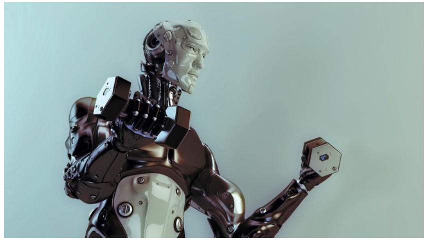
Dialogue Engine



Fine tuning LLM
Txt to Parses

“To parse user utterances into the grammar, we finetune an LLM to translate utterances into the grammar in a seq2seq fashion”

Dialogue Engine

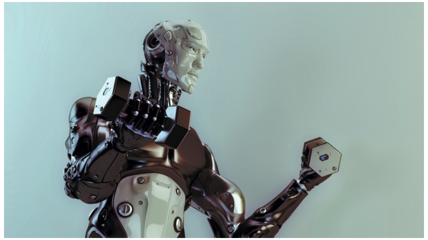


Fine tuning LLM
Txt to Parses

What is finetune?

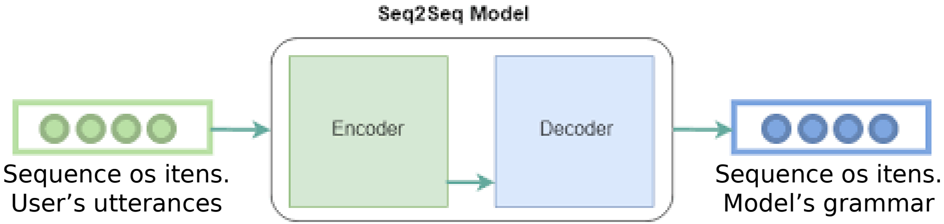
“To parse user utterances into the grammar, we finetune an LLM to translate utterances into the grammar in a seq2seq fashion”

Dialogue Engine



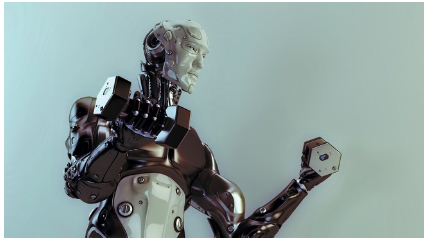
Fine tuning LLM
Txt to Parses

"To parse user utterances into the grammar, we finetune an LLM to translate utterances into the grammar in a seq2seq fashion"



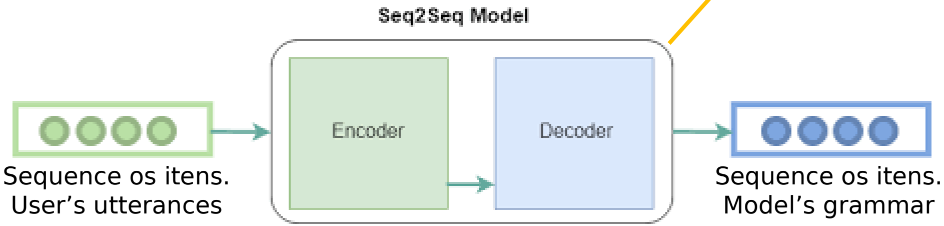
Dialogue Engine

- Few-shot GPT-J
- Finetuned T5



Fine tuning LLM
Txt to Parses

"To parse user utterances into the grammar, we finetune an LLM to translate utterances into the grammar in a seq2seq fashion"



Dialogue Engine

Generate fine tuning data

How to generate fine tuning data?

Human annotation?

Dialogue Engine

Generate fine tuning data

How to generate fine tuning data?

~~Human annotation?~~

Dialogue Engine

Generate fine tuning data

How to generate fine tuning data?

1 - Write a initial set of user’s utterances and parses (where part of utterances & parses are wildcards terms).

2 - TalkToModel enumerates the wildcards with terms in the user’s provided data

Dialogue Engine

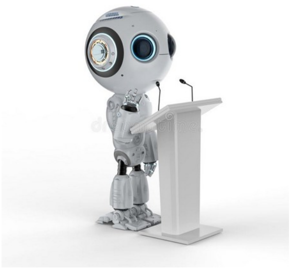
Generate fine tuning data

How to generate fine tuning data?

1 - Write a initial set of user’s utterances and parses (where part of utterances & parses are wildcards terms).
2 - TalkToModel enumerates the wildcards with terms in the user’s provided data

TalkToModel usually generates 20k to 40k pairs of parses.

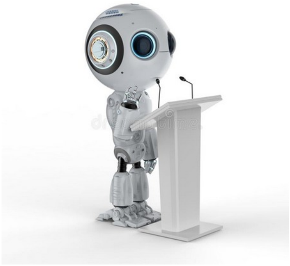
Dialogue Engine



Respond conversationally

"After TalkToModel executes a parse, it composes the results of the operations into a natural language response it returns to the user."

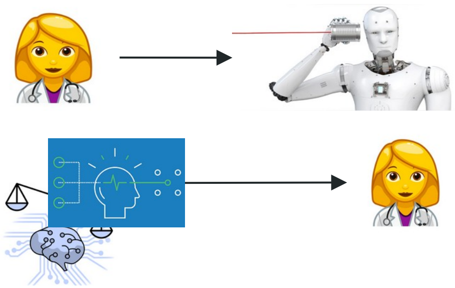
Dialogue Engine



Respond conversationally

- TalkToModel generates responses using **templates** associated with each operation!
- TalkToModel can run multiple operations at the same time. In this case, the model will join responses templates ensuring semantic coherence.

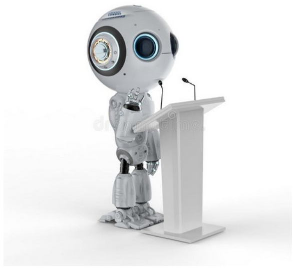
Dialogue Engine



Constructing a grammar

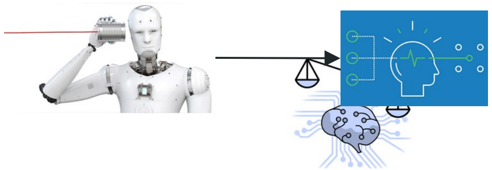
Generate fine tuning data

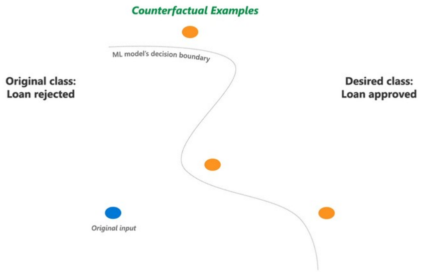
Fine tuning LLM
Txt to Parses

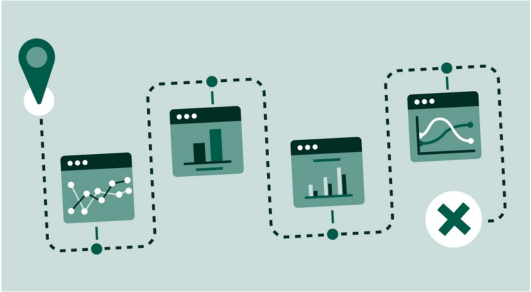


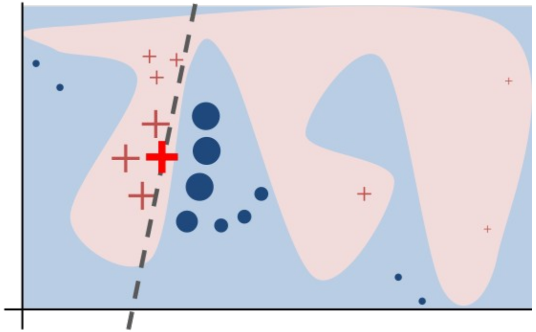
Respond
conversationally

Execution Engine





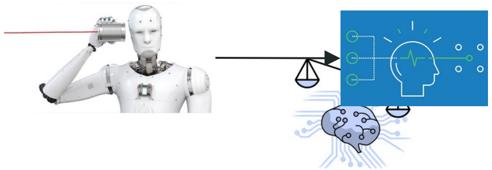




Counterfactual Explanations:
TalkToModel uses DiCE because it provides a diverse set of counterfactuals.

Data and Predictions Exploration
TalkToModel allow users to analyse predictions, inspect the model for errors, and analyse de data itself.

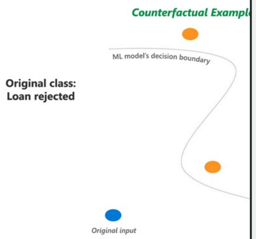
Post-hoc Feature Explanations
TalkToModel uses LIME and SHAP to make feature importance based explanations.

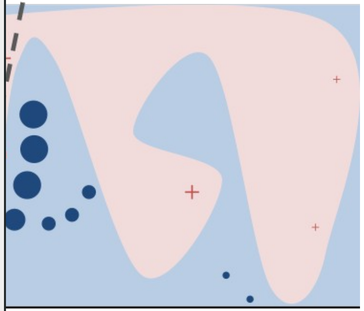


Execution Engine

TalkToModel selects the best explanation!

Instead of providing the LIME coefficients or the SHAP values, the model test the methods and provide the “best” feature based explanation.





Counter
Explana

TalkToModel uses DiCE because it provides a diverse set of counterfactuals.

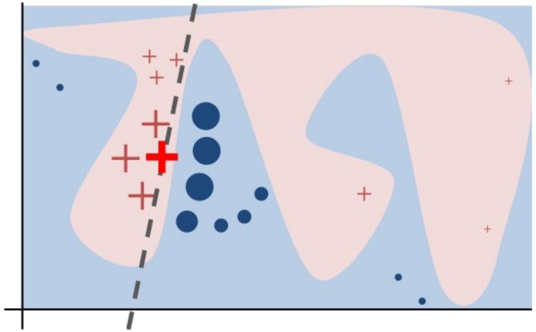
TalkToModel allow users to analyse predictions, inspect the model for errors, and analyse de data itself.

-hoc Feature
planations

TalkToModel uses LIME and SHAP to make feature importance based explanations.

Execution Engine

How to select the best explanation?



Explanation Selection

Setup:

$f(\mathbf{x}) \rightarrow \mathbf{y}$

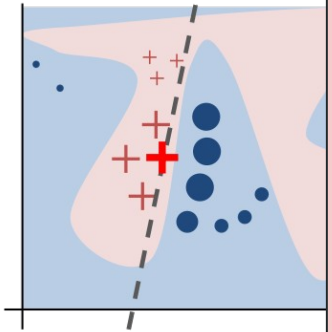
Is the model outputting the probability (y) of a class.

$\Phi(\mathbf{x},\ f) \rightarrow \phi$

Feature importances for model f on the data x. greater magnitudes correspond to higher importance features

Execution Engine

How to select the best explanation?



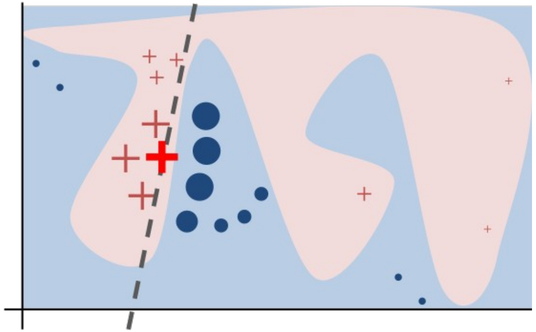
g the probability (y)

for model f on the
itudes correspond to
atures

How to choose the “best” feature importance?

More Important = **Small feature** perturbations lead to **Big score** perturbations

Execution Engine



Explanation Selection

How to select the best explanation?

The Fudge score!

Perturbed features are given by m

Gaussian Noise

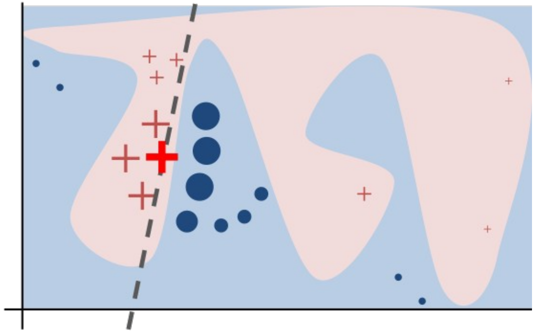
$$\mathrm{Fudge}(f, \mathbf{x}, \mathbf{m}) = \frac{1}{N} \sum_{n=1}^{N} |f(\mathbf{x}) - f(\mathbf{x} + \epsilon_n \odot \mathbf{m})|$$

original prediction

perturbed prediction

Execution Engine

How to select the best explanation?



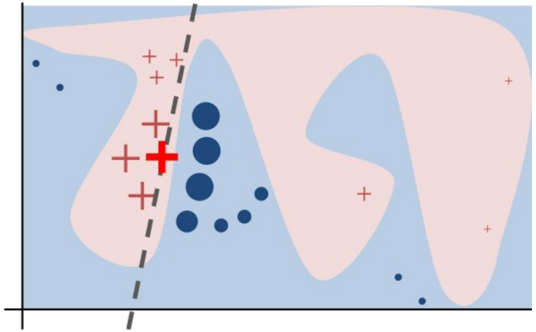
Explanation Selection

The Fudge score!

$$\text{Fudge}(f, \mathbf{x}, \mathbf{m}) = \frac{1}{N} \sum_{n=1}^{N} \underbrace{|f(\mathbf{x}) - f(\mathbf{x} + \epsilon_n \odot \mathbf{m})|}_{\text{Magnitude of perturbation by adding feature noise}}$$

Execution Engine

How to select the best explanation?

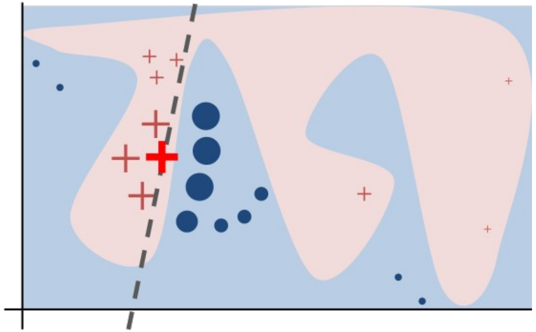


Explanation Selection

The Fudge score!

$$\mathrm{Fudge}(f, \mathbf{x}, \mathbf{m}) = \underbrace{\frac{1}{N}\sum_{n=1}^{N} |f(\mathbf{x}) - f(\mathbf{x} + \epsilon_n \odot \mathbf{m})|}_{\text{Average Magnitude of perturbations by adding feature noise}}$$

Execution Engine



Explanation Selection

How to select the best explanation?

The Fudge score

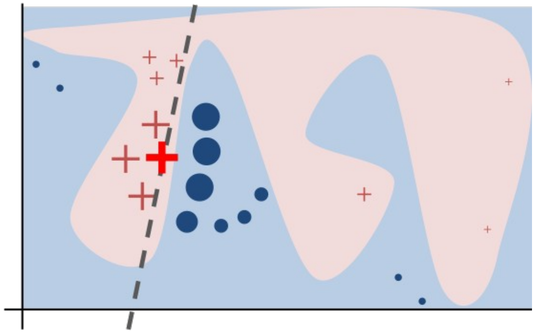
$$\mathrm{Fudge}(f, \mathbf{x}, \mathbf{m}) = \frac{1}{N}\sum_{n=1}^{N} |f(\mathbf{x}) - f(\mathbf{x} + \epsilon_n \odot \mathbf{m})|$$

Feature Importance Faith

$$\mathrm{Faith}(\phi,\ f,\ \mathbf{x},\ K) = \sum_{k=1}^{K} \mathrm{Fudge}(f,\ \mathbf{x},\ \mathbb{1}(k, \phi))$$

Indicator of top k features

Execution Engine



Explanation Selection

How to select the best explanation?

The Fudge score

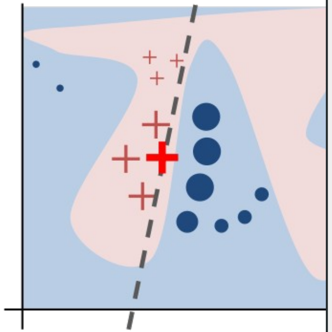
$$\mathrm{Fudge}(f, \mathbf{x}, \mathbf{m}) = \frac{1}{N}\sum_{n=1}^{N} |f(\mathbf{x}) - f(\mathbf{x} + \epsilon_n \odot \mathbf{m})|$$

Feature Importance Faith

$$\mathrm{Faith}(\phi,\ f,\ \mathbf{x},\ K) = \sum_{k=1}^{K} \underbrace{\mathrm{Fudge}(f,\ \mathbf{x},\ \mathbb{1}(k, \phi))}_{\text{Fudge Score of top k features}}$$

Execution Engine

How to select the best explanation?



Explanation Se

Choose the BEST

TalkToModel computes the faithfulness of:

- LIME with kernels [0.25, 0.5, 0.75, 1.0]
- KernelSHAP

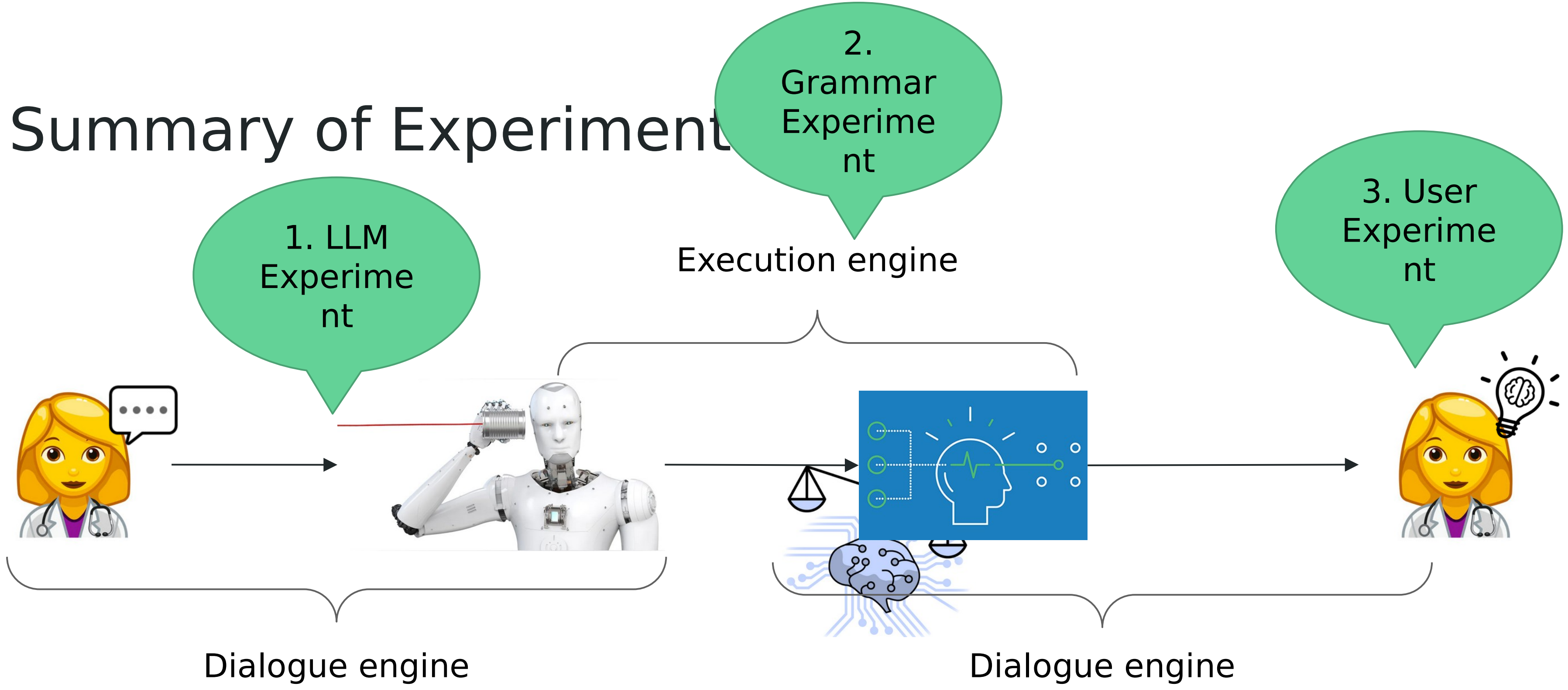
Report the one with highest faith!

$\odot\ \mathbf{m})|$

$, \phi))$

Results

Summary of Experiments



1. LLM Experiment

Is the LLM accurately interpreting the user’s question?

- Create a “Gold Dataset” - ground truth (utterance, parse) pairs specific to one application domain
- Evaluate “Exact Match Accuracy” of LLM translation

1. LLM Experiment

Is the LLM accurately interpreting the user’s question?

- Create a “Gold Dataset” - ground truth (utterance, parse) pairs specific to one application domain
- Evaluate “Exact Match Accuracy” of LLM translation

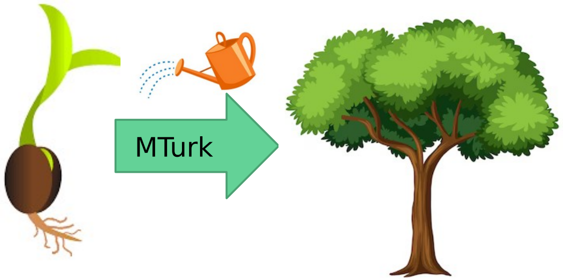
Compare along easy and hard splits of the data

Compare along n-shot and fine-tuning for different LLM sizes

Data Collectior

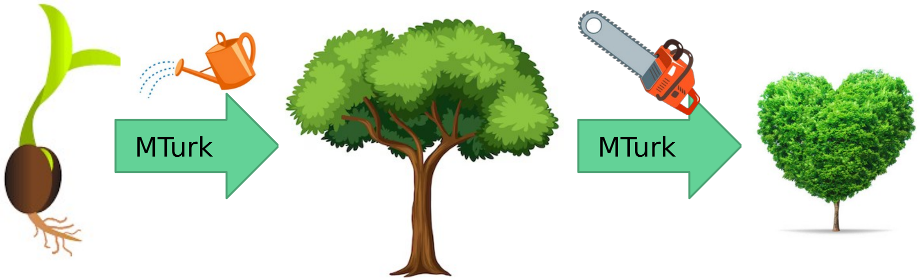
- Authors handwrote 50 (utterance, parse) pairs for each domain
  - Enforce that every operation appears at least twice for good coverage

Data Collectior



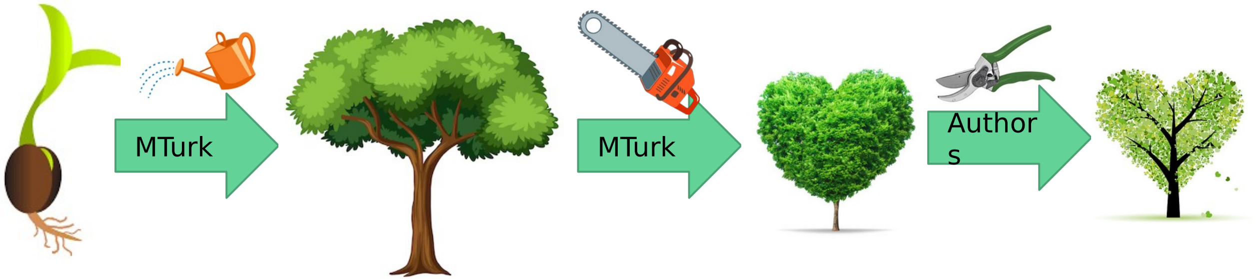
- Authors handwrote 50 (utterance, parse) pairs for each domain
  - Enforce that every operation appears at least twice for good coverage
- Use MTurk to paraphrase utterances in 8 different ways
  - For a total of 400

Data Collection



- Authors handwrote 50 (utterance, parse) pairs for each domain
  - Enforce that every operation appears at least twice for good coverage
- Use MTurk to paraphrase utterances in 8 different ways
  - For a total of 400
- Use MTurk to rate fidelity of paraphrase to original utterance
  - Keep those that rate 3/4 or higher averaged over 5 raters

Data Collectior



- Authors handwrote 50 (utterance, parse) pairs for each domain
  - Enforce that every operation appears at least twice for good coverage
- Use MTurk to paraphrase utterances in 8 different ways
  - For a total of 400
- Use MTurk to rate fidelity of paraphrase to original utterance
  - Keep those that rate 3/4 or higher averaged over 5 raters
- Manual Filtering by Authors
  - Unclear (in paper) what proportion of filtering is done by MTurk vs. Authors

Example Paraphrases from TTM’s Open Source Data

“What is your reasoning for determining if people older than 20 are likely to commit crimes?”

Example Paraphrases from TTM’s Open Source Data

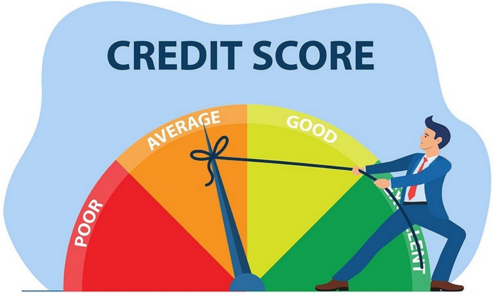
“What is your reasoning for determining if people older than 20 are likely to commit crimes?”

- “Why do you think people over the age of twenty are likely to commit a crime?” 4 4 4 3 3
- “How did you determine the likelihood of people over 20 committing crimes?” 2 4 3 4 4
- “Can you reason why people over twenty would likely commit crimes?” 3 2 4 3 2

Domains/Datasets

- <u>Diabetes:</u> Pima Indian Diabetes Dataset

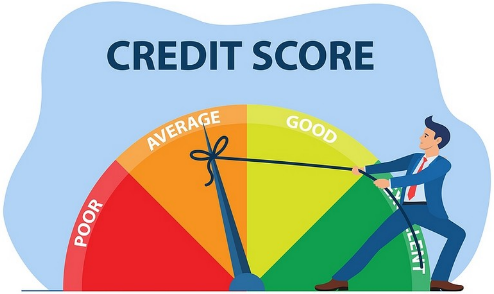
- <u>Credit:</u> German Credit Dataset



- <u>Recidivism:</u> COMPAS

Domains/Datasets

- <u>Diabetes:</u> Pima Indian Diabetes Dataset
  - 768 Women from Phoenix, AZ
  - 8 Health Features, Diabetes or Not
  - Questions: 400 → 190



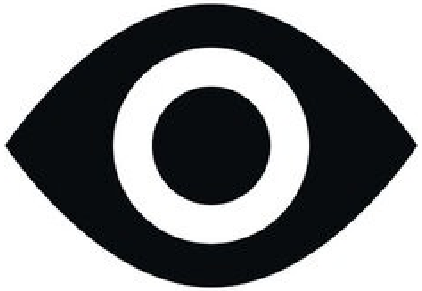
- <u>Credit:</u> German Credit Dataset
  - 1000 Loan Applicants
  - 20 Financial Features, Good or Bad
  - Questions: 400 → 200

- <u>Recidivism:</u> COMPAS
  - 11757 Men and Women Criminal Defendants
  - 43 Demographic/History Features, Risk Score 1
  - Questions: 400 → 146

Splits of the Gold Dataset

IID (Easy)

- Order of operations are in the training data
  - Allowing different arguments



Compositional (Hard)

- Order of operations not seen before in the training dataset

	German			Compas			Diabetes		
	IID	Comp.	Overall	IID	Comp.	Overall	IID	Comp.	Overall
Nearest Neighbors	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
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**

	German			Compas			Diabetes		
	IID	Comp.	Overall	IID	Comp.	Overall	IID	Comp.	Overall
Nearest Neighbors	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
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**

	German			Compas			Diabetes		
	IID	Comp.	Overall	IID	Comp.	Overall	IID	Comp.	Overall
Nearest Neighbors	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
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**

2. Grammar Experiment

Is the grammar expressive enough to capture all XAI questions?

- Use an XAI question bank
  - Previous work, informed by design expert interviews
- Manually review if grammar can answer questions
- 30/31 questions can be answered!
  - More questions deemed out of scope

2. Grammar Experiment

Is the grammar expressive enough to capture all XAI questions?

- Use an XAI question bank
  - Previous work, informed by design expert interviews
- Manually review if grammar can answer questions
- 30/31 questions can be answered!

deemed ou

What features does the system consider?
`topk(test_data, all)`

What would the system predict if a given feature A changes to..?
`predict(change(filter(test_data, id, A), feature, value, set))`

What kind of mistakes is the system likely to make?
`mistakes(test_data)`

How should instance A change to get a different prediction Q?
`cfe(filter(test_data, id, A, =), 10, Q)`

	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 followups
Description	function(): Overview of the system's capabilities
	data(dataset): Summary of dataset
	model(): Description of model
	define(term): Defines term

3. User Experiment

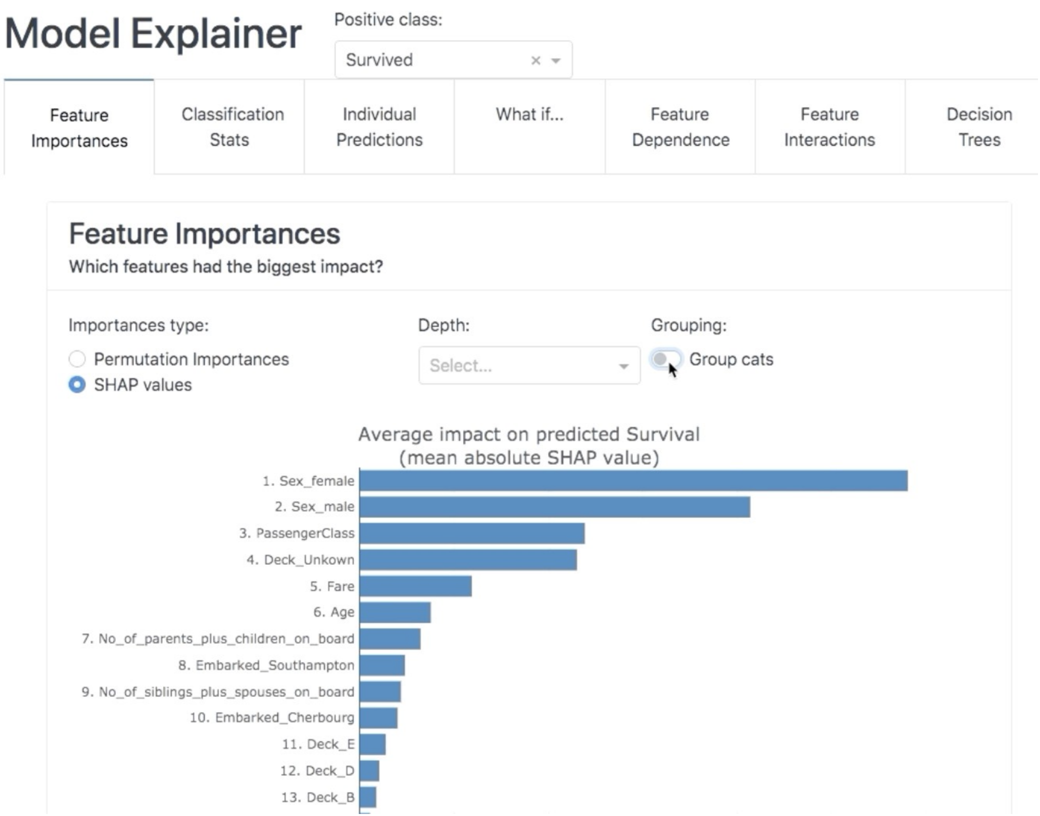
- Diabetes dataset with a gradient-boosted tree model
- 45 healthcare workers; 12 ML grads
- Answer 10 XAI MC questions
- Survey user preference vs. *explainerdashboard*
    - Ease of use, confidence, speed, and likability

Example question: “Is glucose more important than age for the model’s predictions for data point 49?”

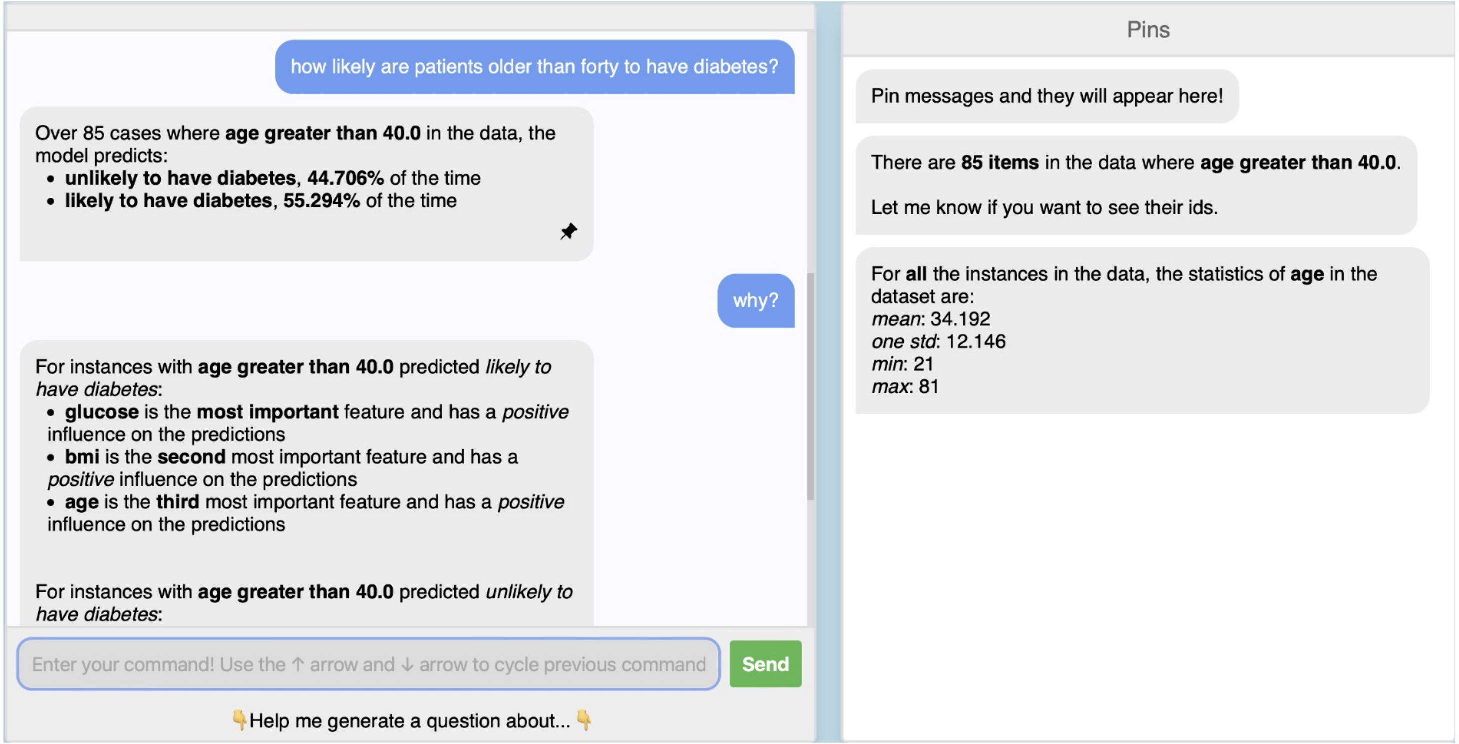
Comparison	% Agree TalkToModel Better: Health Care Workers	% Agree TalkToModel Better: ML Grad. Students
Easiness	82.2	84.6
Confidence	77.7	69.2
Speed	84.4	84.6
Likeliness To Use	73.3	53.8

	% Questions Completed		% Accuracy On Completed Questions	
	Dash.	TalkToModel	Dash.	TalkToModel
Health Care Workers	74.7	86.2	66.1	91.8
ML Grad. Students	73.8	93.9	62.5	100.0

explainerdashboard



TalkToModel



Findings

Users using TalkToModel (as compared to explainerdashboard)

- Over 90% accurate compared to 60% otherwise
- Got answers in half the time
- Consistently preferred it on ease of use, confidence, speed, and likability

Conclusions

- TalkToModel provides an elegant UI and conversation tool that makes interpreting models easier for laypeople and ML practitioners alike
- Highly extensible to handle a variety of explainability needs, problem domains, and XAI methods
- Reasonably accurate at interpreting user intent and provides well-formatted templated responses
- A dataset of your own is all you need to use TalkToModel

Limitations

- Authors do not test the system in real world settings
- Automation of explainability methods gives no flexibility on user side (e.g., “what is the most feasible CFE” or “what is the most stable post-hoc explanation”)
- No guarantees on data quality for both training and fine-tuning, and still requires some manual labor
- System possesses no domain knowledge besides grammar
- Accuracy on hard split is still very low

Discussion Questions

- Are ML practitioners the most responsible for accessibility of XAI?
- Is the TalkToModel LLM itself interpretable?
  - When is it acceptable to improve XAI with more black box AI?
- How much control over the explanation should we give to the user while still being accessible to the layperson?
- Does the existence of TalkToModel excuse the need
  other XAI methods to be accessible?
- Dashboard vs. Dialogue?