Applying Counterfactual Explanations in Evolving Scenarios and Expert Domains

Karol Dobiczek

16.10.2025

.

Karol Dobiczek

Education:

- TU Delft
  2019-2024
  Bachelor's and Master's
- Since Oct 2025
  PhD candidate

Work:

- qualtrics XM
  Software Engineer
  Sep 2024 – Oct 2025
- MI
  Research Software Engineer
  Nov 2024 - now

Agenda

1. Endogenous Macrodynamics in Algorithmic Recourse
2. Natural Language Counterfactual Explanations in Financial Text Classification
3. “Positive” Red-Teaming for Large Language Models
4. Current and future directions

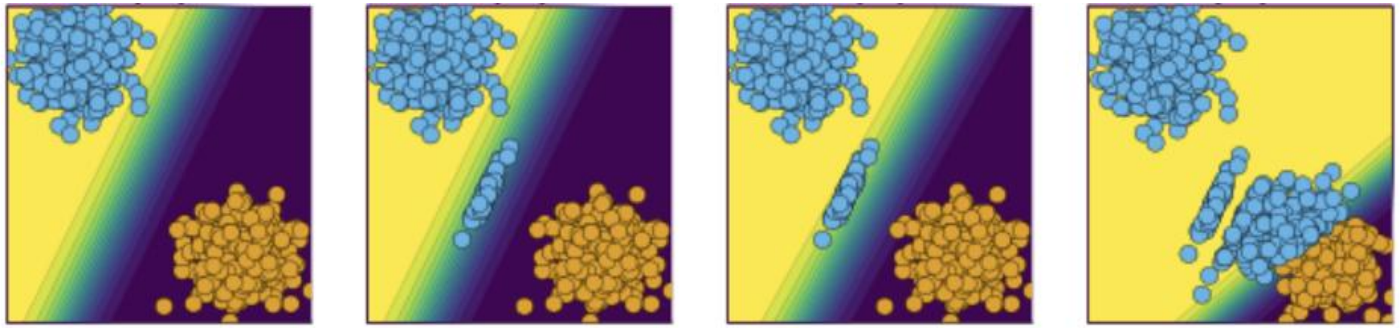
Endogenous Macrodynamics of Algorithmic Recourse

Patrick Altmeyer, Giovan Angela, Aleksander Buszydlik,
Karol Dobiczek, Arie van Deursen, Cynthia Liem

First IEEE Conference on Secure and Trustworthy Machine Learning. SaTML 2023.
Raleigh, North Carolina, United States

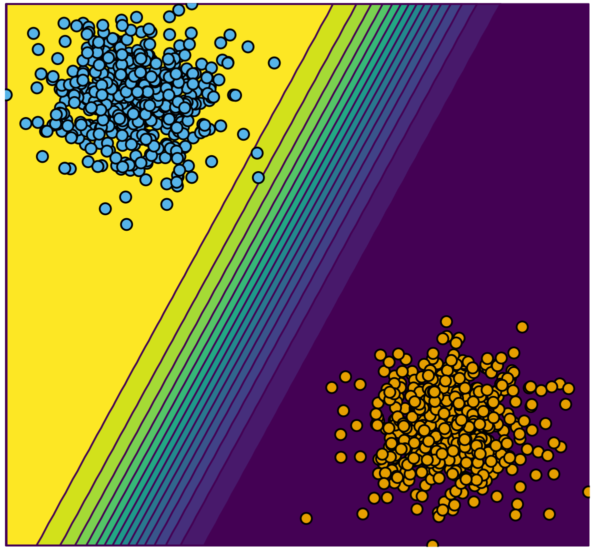
Motivation

- Counterfactual Explanations (CEs) that involve realistic and actionable changes can be used for the purpose of **Algorithmic Recourse** (AR) to help individuals who face adverse outcomes.
- What happens if we apply CEs and adjust our model?
  What happens if we do it again?



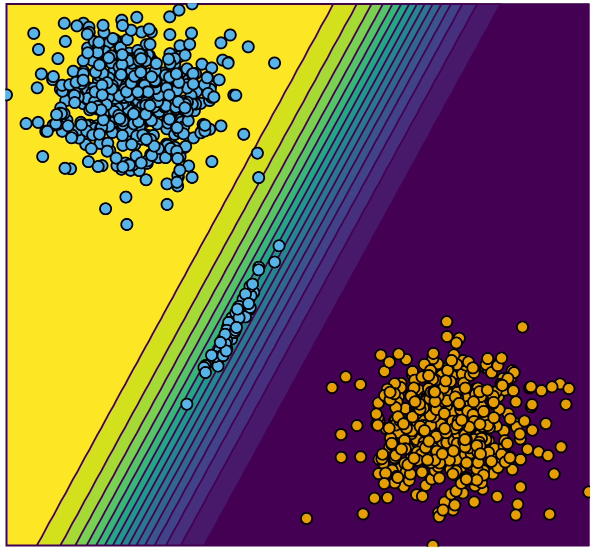
Motivation – Proof of Concept

(a)



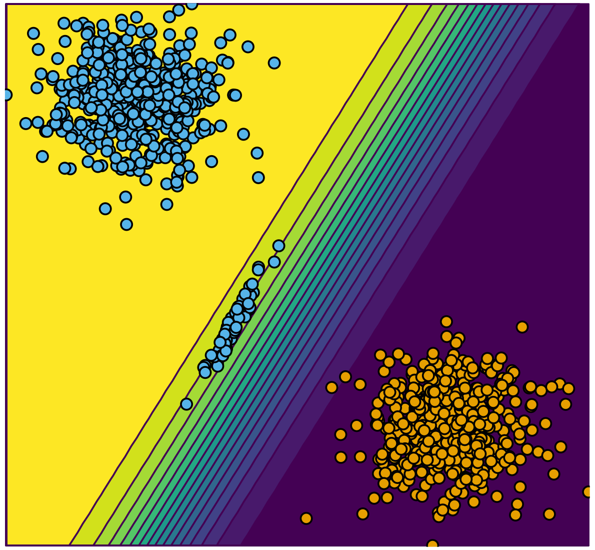
A bank trains a classifier to evaluate credit applicants.

(b)



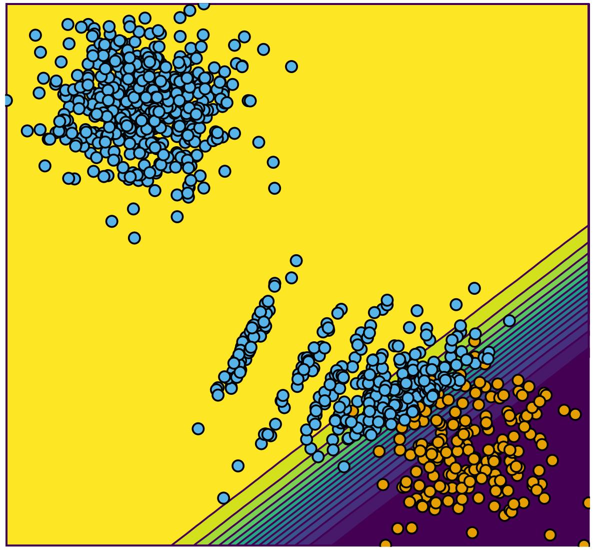
The bank gives CEs to unsuccessful applicants – endogenous domain shift

(c)



The bank retrains the classifier – endogenous model shift

(d)



Process repeated several times, sizeable shifts occur. The overall risk has increased.

Motivation – Background

Exogenous vs Endogenous

We have been able to identify only one recent work by Upadhyay et al. that considers the implications of **exogenous** domain and model shifts in the context of AR. Exogenous shifts are strictly of external origin.

We refer to these types of dynamics as **endogenous** because they are induced by the implementation of recourse itself.

Macrodynamics

The term **macrodynamics** is borrowed from the economics literature and used to describe processes involving whole groups or societies.

Motivation – Questions

1. Do the CEs generated by SOTA generators lead to shifts in domains and models?
2. Are the explanations valid if applied in practice?
3. Who should bear the cost and risks of algorithmic recourse?

Experiments – CE generators

1. Wachter (Generic) – Minimal distance (Wachter et al.)
2. DiCE – Diverse counterfactuals (Mothilal et al.)
3. CLUE – Latent with minimized predictive uncertainty (Antoran et al.)
4. REVISE – Latent, learning data generation process (Joshi et al.)
5. Greedy – Jacobian-based Saliency Map Attack (Schut et al.)

Experiments

Datasets:

- Synthetic data (Moons, Overlapping)
- Give Me Some Credit
- UCI defaultCredit
- California Housing

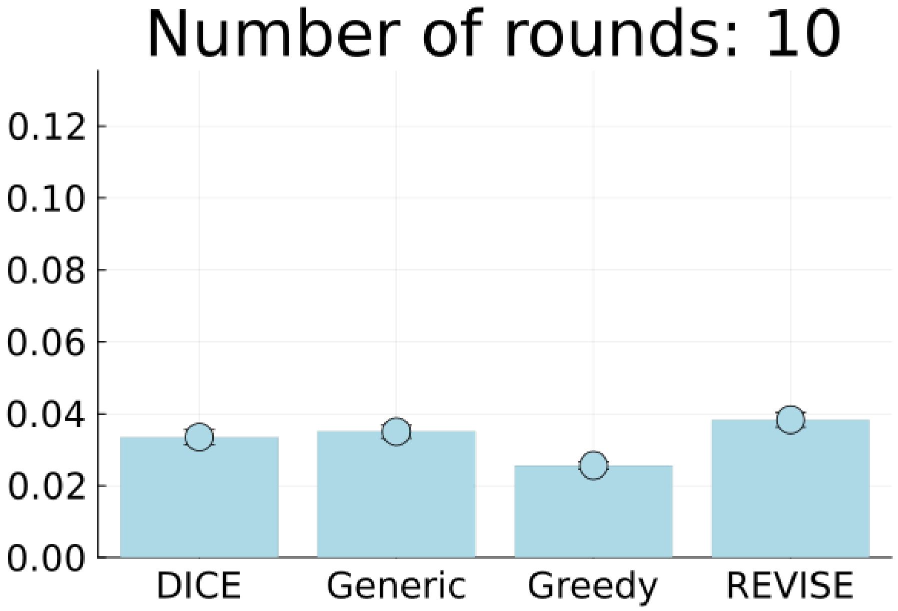
Models:

- Logistic Regression
- Deep Ensemble
- Multilayer Perceptron

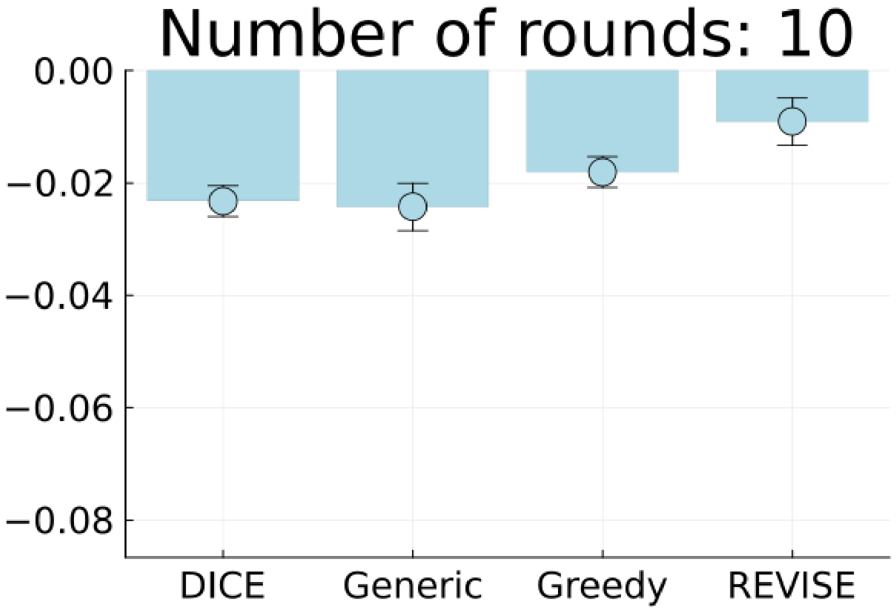
Metrics:

- Maximum Mean Discrepancy (MMD)
- Model “Decisiveness”
- Accuracy Difference
- Model MMD
- Model Disagreement

Results – Synthetic Data

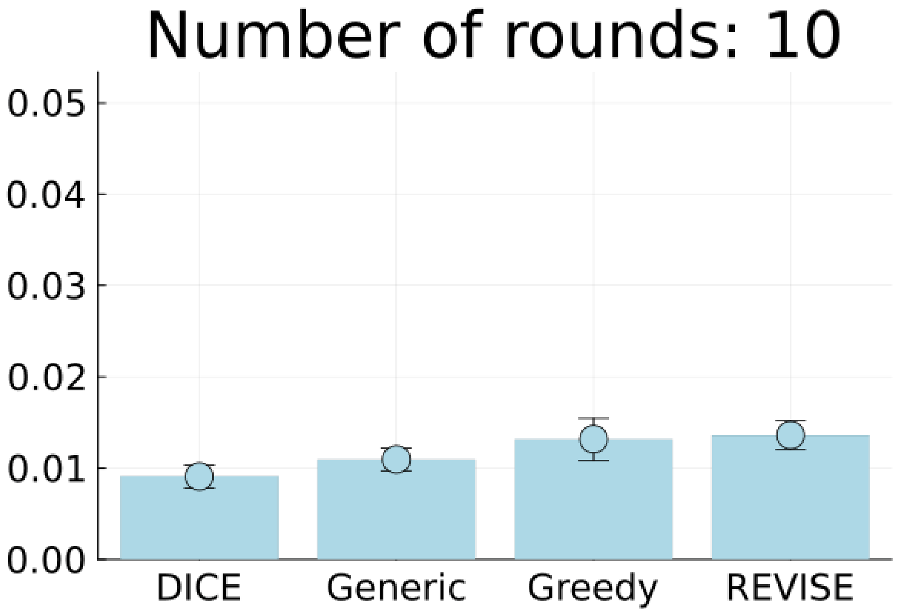


Domain shift for overlapping data using deep ensemble model

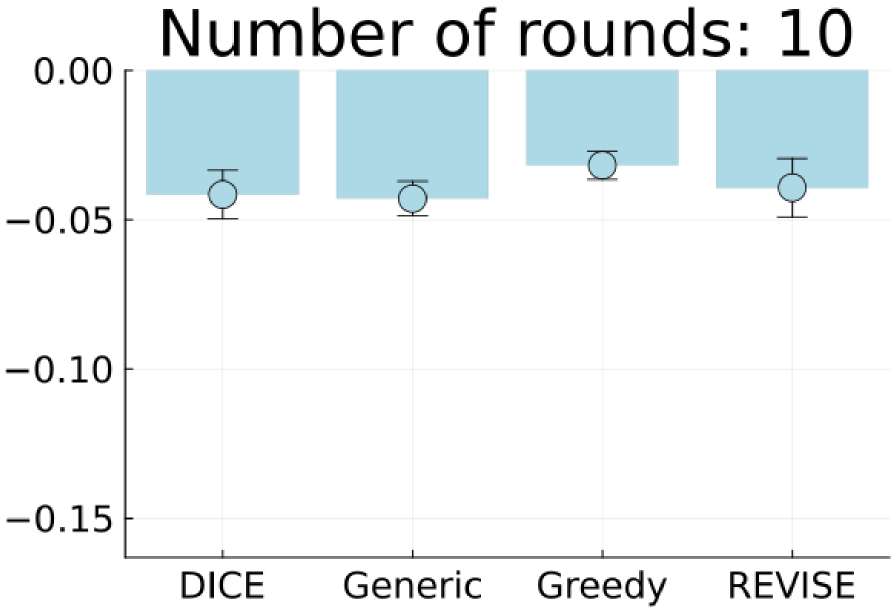


Performance shift for overlapping data using deep ensemble model

Results – Real-world Data



Domain shift for Default Credit using deep ensemble model



Performance shift for Default Credit using deep ensemble model

Mitigating Endogenous Shifts

What are potential mitigation strategies with respect to endogenous macrodynamics in AR?

Externalities of Algorithmic Recourse

Usual baseline for Counterfactual Explanations:

$$x' = \arg\min_{x'}\{yloss(M(x'), y') + \lambda cost(x')\}$$

Minimizing costs for a single individual

Proposed extension to the formula:

$$x' = \arg\min_{x'}\{yloss(M(x'), y') + \lambda_1 cost(x') + \lambda_2 extcost(x')\}$$

Capturing external costs introduced by the CE

Externalities of Algorithmic Recourse

Two strategies for minimizing external AR costs:

1. Classifier Preserving ROAR (ClaPROAR)

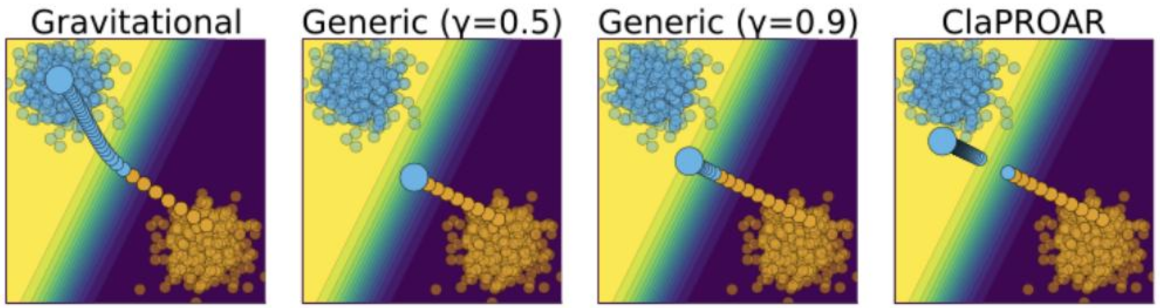
$$extcost(x') = l_M(M(x'), y')$$

2. Gravitational Counterfactual Explanations

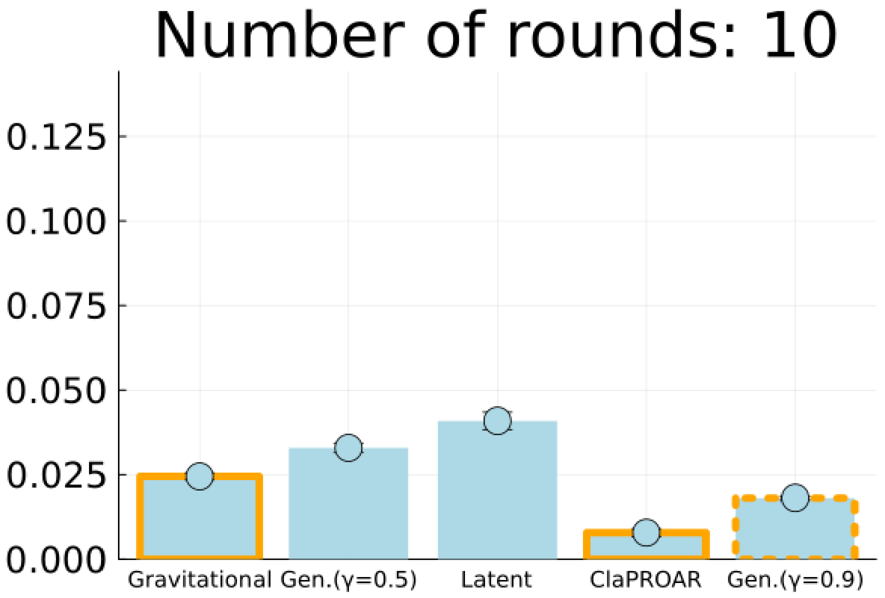
$$extcost(x') = dist(x', \bar{x}')$$

Additionally:

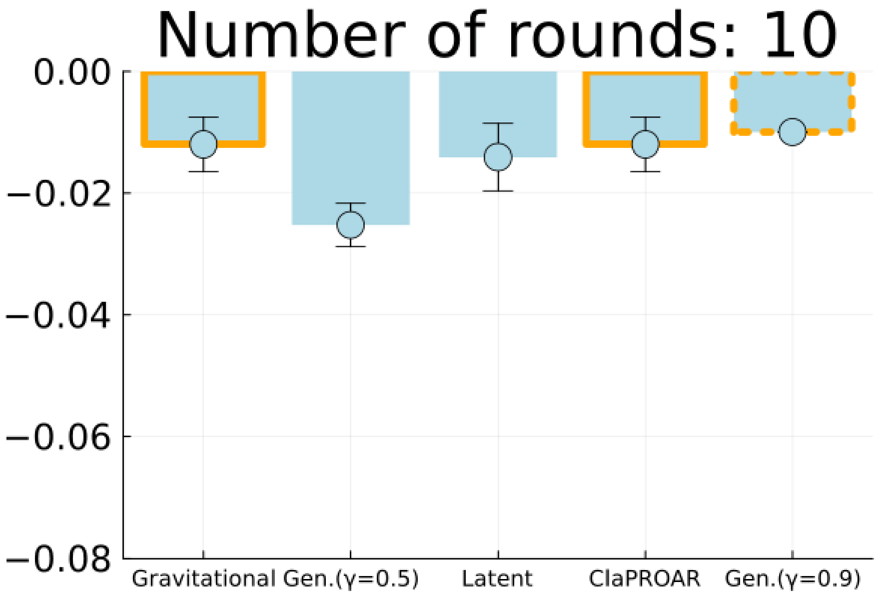
3. Generic CEs with more conservative decision thresholds ($\gamma = 0.9$)



Results – With Mitigation Strategies



Domain shift for overlapping data using deep ensemble model



Performance shift for overlapping data using deep ensemble model

Key Takeaways

- State-of-the-art approaches to AR induce substantial domain and model shifts.
- External costs of Individual Recourse should be shared across stakeholders.
- Our solution: penalize external costs in the counterfactual search objective function.

Natural Language Counterfactual Explanations in Financial Text Classification: A Comparison of Generators and Evaluation Metrics

Karol Dobiczek   Patrick Altmeyer   Cynthia C. S. Liem

Proceedings of the Fourth Workshop on Generation, Evaluation and Metrics (GEM^2 2025),
ACL 2025

Text CEs

Generate text that gets classified to a different class. How do we use our equation?

$$x' = \arg\min_{x'}\{yloss(M(x'), y') + \lambda cost(x')\}$$

We could:

1. Embed text
2. Search latent space
3. Select closest embedding

How do we maintain the desirable properties?

Text fluency/validity, grammatical correctness, reasonability of the explanation, content preservation

Text CEs Usage

- Data enhancement
- Causality assessments
- Explaining and analysing the model
- Help in text composition

Types of Text CEs

1. LLM-assisted

Prompt a LLM to modify a sentence such that it reflects the sentiment we want to have

2. Latent decoding

Perturb the latent embedding such that we get a valid CE

3. Sequential infilling

Mask certain tokens and generate new ones

Examples

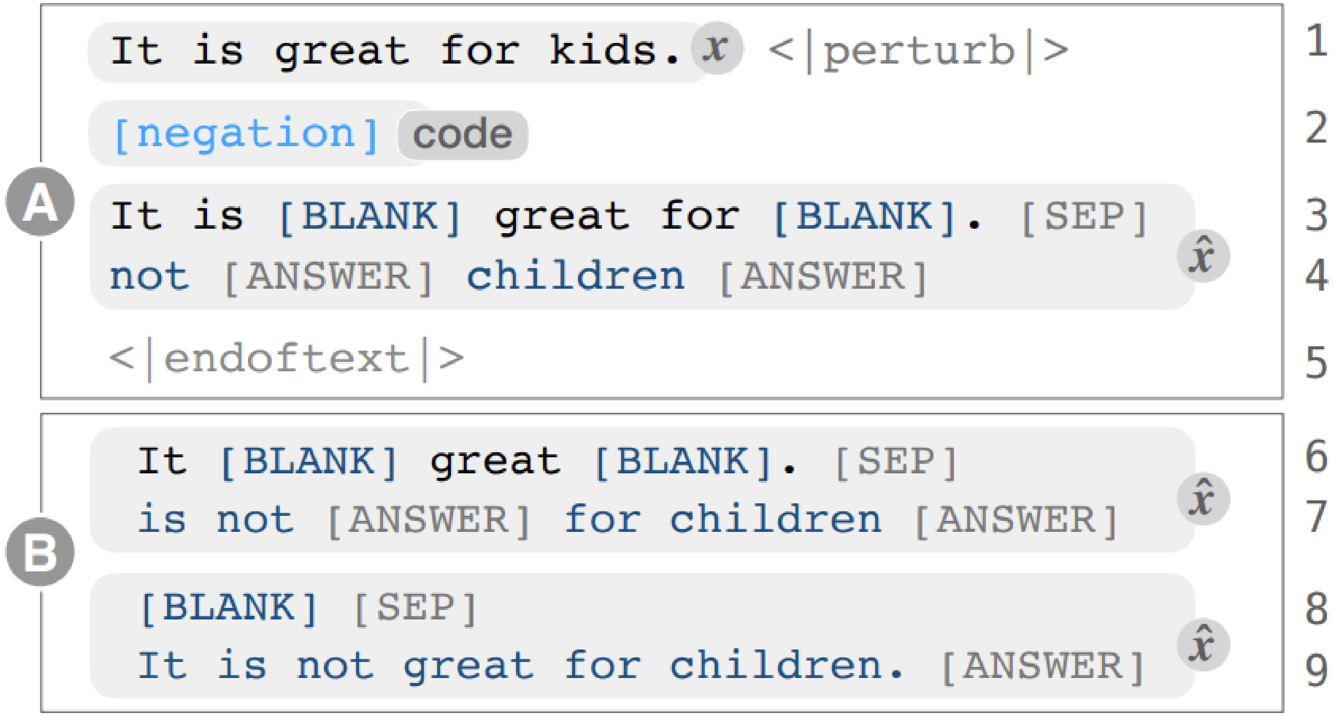
LLM-assisted	Latent decoding	Sequential infilling
Polyjuice, CheckList	PPLM, GYC, CounterfactualGAN	MiCE, RELITC, Polyjuice*

POLYJUICE: Generating Counterfactuals for Explaining, Evaluating, and Improving Models

Tongshuang Wu[1] **Marco Tulio Ribeiro**[2] **Jeffrey Heer**[1] **Daniel S. Weld**[1,3]
[1]University of Washington [2]Microsoft Research [3]Allen Institute for Artificial Intelligence
wtshuang@cs.uw.edu marcotcr@microsoft.com {jheer,weld}@cs.uw.edu

Polyjuice

Fine-tuning GPT-2 for infilling:



Polyjuice – Motivation

- Manual rewrites for counterfactuals costly

  (4-5 minutes per CE (Kaushik et al., 2020))

- Human annotators might miss certain structures e.g.

  It is great for kids.

  great -> not great, but miss kids -> no one

- Automatic generators might neglect prediction-preserving CEs

Polyjuice – Control Codes

Control code	Definitions and POLYJUICE-generated Examples	Training Datasets
`negation`	A dog is not embraced by the woman.	(Kaushik et al., 2020)
`quantifier`	A dog is → Three dogs are embraced by the woman.	(Gardner et al., 2020)
`shuffle`	*To move (or swap) key phrases or entities around the sentence.* A dog → woman is embraced by the woman → dog.	(Zhang et al., 2019b)
`lexical`	*To change just one word or noun chunk without altering the POS tags.* A dog is embraced → attacked by the woman.	(Sakaguchi et al., 2020)
`resemantic`	*To replace short phrases without altering the remaining dependency tree.* A dog is embraced by the woman → wrapped in a blanket.	(Wieting and Gimpel, 2018)
`insert`	*To add short phrases without altering the remaining dependency tree.* A dog is embraced by the little woman.	(McCoy et al., 2019)
`delete`	*To remove short phrases without altering the remaining dependency tree.* A dog is embraced by the woman.	(McCoy et al., 2019)
`restructure`	*To alter the dependency tree structure,* e.g., *changing from passive to active.* A dog is embraced by → hugging the woman.	(Wieting and Gimpel, 2018)

Table 1: We design a list of control codes to guide generation. We show POLYJUICE-*generated* counterfactual examples, and the representative training datasets for each corresponding pattern. Details are in Appendix A.

Polyjuice – Discussion

Pros:

- High fluency due to use of a LLM
- Content and structure preservation

Cons:

- High pre-training or fine-tuning cost
- Only fine-tuned on “simple” tasks

Plug and Play Language Models: a Simple Approach to Controlled Text Generation

Sumanth Dathathri *
CMS, Caltech

Andrea Madotto *
HKUST

Janice Lan
Uber AI

Jane Hung
Uber AI

Eric Frank
Uber AI

Piero Molino
Uber AI

Jason Yosinski †
Uber AI

Rosanne Liu †
Uber AI

dathathris@gmail.com, amadotto@connect.ust.hk
{janlan, jane.hung, mysterefrank, piero, yosinski, rosanne}@uber.com

Published as a conference paper at ICLR 2020

PPLM - Motivation

- Motivated by Plug & Play Generative Networks (PPGN)
- Modelling $p(x|a)$, where a is a controllable attribute (class)
- Estimate $p(x|a)$ as $p(a|x)p(x)$
  - $p(a|x)$ – attribute model
  - $p(x)$ – generative model

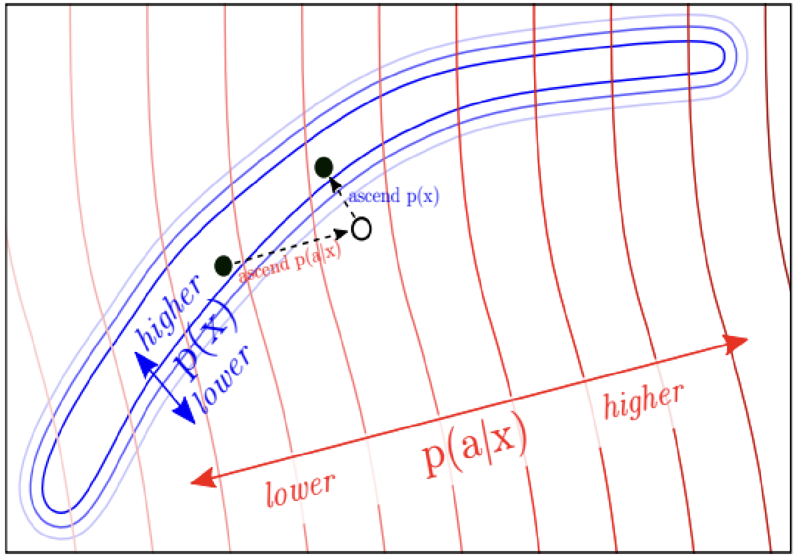
PPLM – Motivation

In case of Language Models:

- $p(a|x)$ – discriminator model (PPLM: 1 FC layer) or Bag of Words
- $p(x)$ – unconditional GPT-2

How do we use those models?

1. Take the latent representation of x
2. Perturb it so that we get higher $p(a|x)$
    1. Increase $p(a|x)$
    2. Increase $p(x)$



PPLM – Latent Representation

Given token sequence $X = \{x_0, x_1, \dots x_n\}$ let history matrix H_t

$$H_t = \left[\left(K_t^{(1)}, V_t^{(1)}\right), \left(K_t^{(2)}, V_t^{(2)}\right), \dots, \left(K_t^{(l)}, V_t^{(l)}\right)\right]$$

Where $\left(K_t^{(n)}, V_t^{(n)}\right)$ is the Key-Value pair of the self-attention layer n at time step t

For latent perturbations initialize a $\Delta H_t = 0$

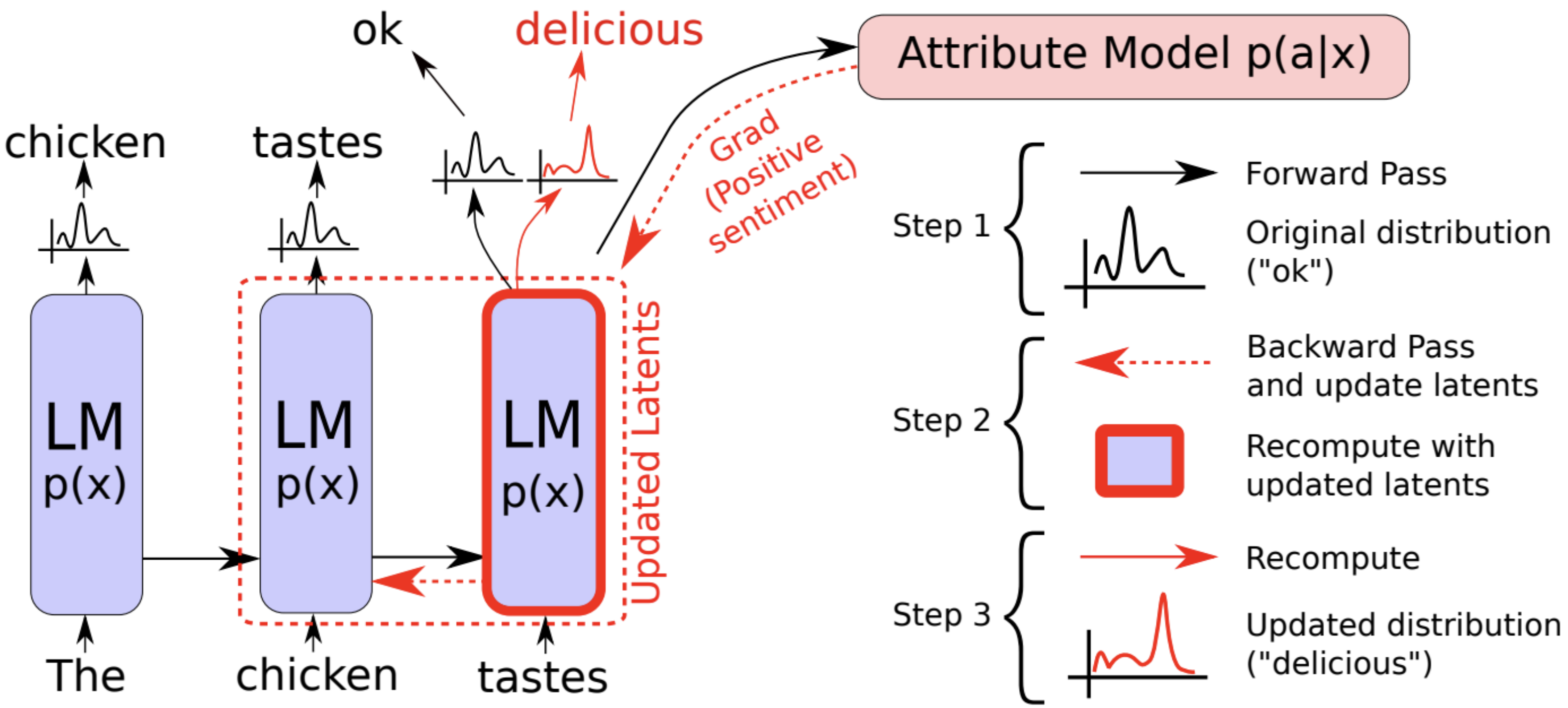
PPLM – Ascending p(x|a)

Calculate log likelihood gradient $\nabla_{\Delta H_t} \log p(a|H_t + \Delta H_t)$ and update H_t:

$$\Delta H_t \leftarrow \Delta H_t + \alpha \frac{\nabla_{\Delta H_t} \log p(a|H_t + \Delta H_t)}{||\nabla_{\Delta H_t} \log p(a|H_t + \Delta H_t)||}$$

Minimalize KL-Divergence between the original $LM(x_t, H_t)$ and $LM(x_t, H_t + \Delta H_t)$

PPLM – "In Practice"



PPLM – Examples

Table 3: Comparison of different samples generated by (top row) baseline GPT-2 and (other rows) PPLM with different BoW corresponding to different topics (e.g. **[Military]**), all conditioned on a single prefix: "The issue focused". Both directly optimized (in red) and related words (in soft red) are highlighted, showing how the optimization takes effect.

[–] The issue focused on the way that the city's police officers have reacted in recent years to the deaths of Michael Brown in Ferguson, Mo., Eric Garner in New York City and Sandra Bland in Texas, as well as the shooting of unarmed teen Michael Brown by a white police officer in Ferguson, Mo. . . .	
[Military] The issue focused on the fact that the government had spent billions on the military and that it could not deploy the troops in time. The prime minister said that the country would take back control of its airspace over Syria in the next 48 hours. \n The military is investigating why. . .	
[Space] The issue focused on a series of incidents that occurred in the past few months, which included an alleged attack by Islamic State fighters on a Kurdish checkpoint, the use of drones in combat, space technology research by Russian and American space companies, and more. \n The world. . .	
[Science] The issue focused on a single piece: the question "What is the meaning of life?" This question has puzzled many philosophers, who have attempted to solve it by using some of the concepts of quantum mechanics, but they have to solve it by the laws of nature themselves.. . .	
[Politics] The issue focused on a single section of the legislation. It's unclear whether the committee will vote to extend the law, but the debate could have wider implications. \n "The issue of the law's applicability to the United Kingdom's referendum campaign has been one of. . .	
[Computers] The issue focused on the role of social media as a catalyst for political and corporate engagement in the digital economy, with the aim of encouraging companies to use the power of social media and the Internet to reach out to their target market. \n . . .	

PPLM – Discussion

Pros:

- High fluency of the unconditional LM
- Low training overhead

Cons:

- Does not retain the structure of the original sentence
- Prone to hallucinations and runaways:

  "[Space] The potato has been successfully cultivated in our solar system. The potato stars stars stars stars stars stars..."

Relevance-based Infilling for Natural Language Counterfactuals

Lorenzo Betti
ISI Foundation
Turin, Italy
Department of Network and Data Science, Central European University
Vienna, Austria
lrn.betti@gmail.com

Carlo Abrate
CENTAI
Turin, Italy
Sapienza University
Rome, Italy
carlo.abrate@centai.eu

Francesco Bonchi
CENTAI
Turin, Italy
Eurecat
Barcelona, Spain
bonchi@centai.eu

Andreas Kaltenbrunner
ISI Foundation
Turin, Italy
Universitat Oberta de Catalunya
Barcelona, Spain
kaltenbrunner@gmail.com

In Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM ’23), October 21–25, 2023, Birmingham, United Kingdom.

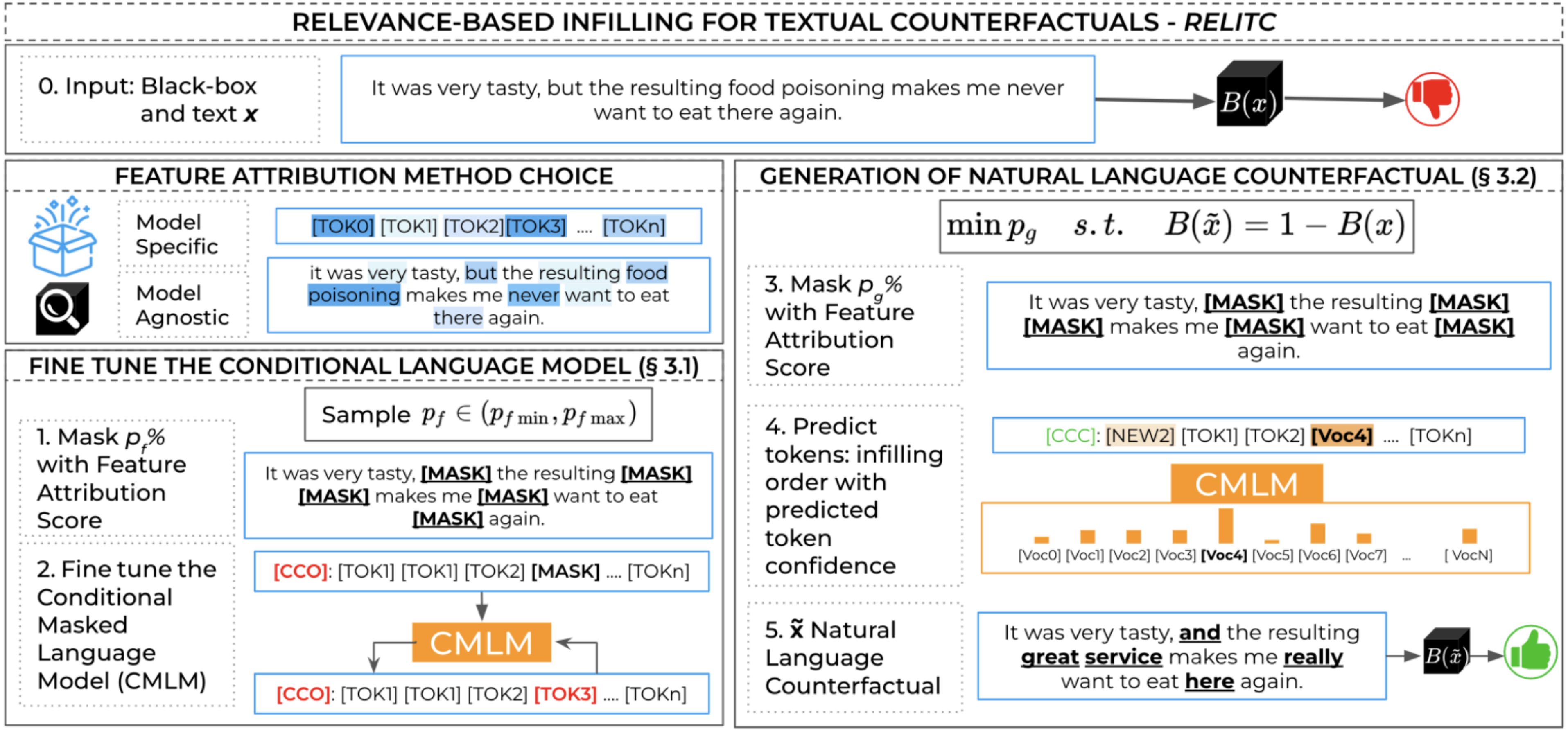
RELITC – Motivation

- Instead of generating texts that might be counterfactuals, **use the classifier itself in the process**
- Desiderata: closeness, feasibility, content preservation
- Conditional Masked LM

  Use CMLM's uncertainty to guide the infilling

- Based on MiCE

RELITC – Overview



RELITC – CMLM

Conditional Masked Language Model – BERT

Task: `The quick [MASK] jumps over…` → predicted token logits

Condition on label: add to the fine-tuning text:

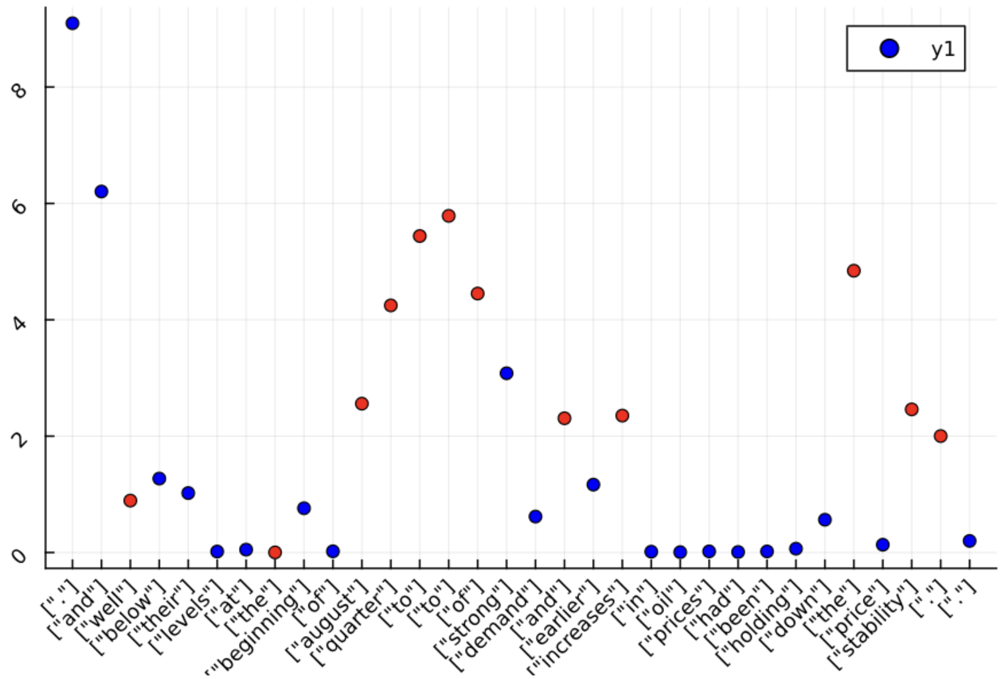
`[Negative] The bad [MASK] jumps over…`

`[Positive] The good [MASK] jumps over…`

RELITC – Infilling

Logit entropy as proxy for uncertainty

Choose lowest entropy first



RELITC – Examples

Method	CallMe Example Text
Input text	the husband is responsible for the family so the wife must obey him.
HUMAN	the parents are responsible for the family so the children must obey them.
RELITC$_{L2R}$	the **president** is responsible for the family **and the people who** obey him.
RELITC$_{Conf}$	the **lord** is responsible for the family **and the family must** obey him.
MiCe	the **grandpa** is responsible for the family so the **fam** must **protect** him.

Metrics	CF. label	NED	Fluen.	Cont. Pres.	Mask Frac.
HUMAN	Non-sexist	0.286	0.981	0.703	-
RELITC$_{L2R}$	Non-sexist	0.286	1.245	0.512	0.313
RELITC$_{Conf}$	Non-sexist	0.214	0.641	0.743	0.250
MiCe	Non-sexist	0.214	2.389	0.303	0.172

RELITC – Discussion

Pros:

- Explanations closer to the classifier
- Conditioning on label with CMLM
- Content and structure preservation

Cons:

- Might lack fluency
- Attribution calculation adds overhead

Our work - Motivation

Central banks moderate the public expectations by emitting communications. Text CEs could help them know if they convey certain sentiment (hawkish or dovish):

- Provide a new perspective how a sentence might be understood.
- Provide ways to better convey the message.

Evaluation of previous text CE methods focused on simple tasks.

How should we evaluate these methods?

Methods

Evaluating text counterfactual explanations

Quantitatively:

- Edit distances
- Embedding distance
- Label flip
- Faithfulness
- Implausibility
- Perplexity
- ...

Qualitatively:

- Fluency
- Minimality
- Grammar correctness
- Plausibility
- Naturalness
- ...

Methods

Evaluating text counterfactual explanations

Quantitatively:

- Edit distances
- Embedding distance
- Label flip
- Faithfulness
- Implausibility
- Perplexity
- ...

Qualitatively:

- Fluency
- Minimality
- Grammar correctness
- Plausibility
- Naturalness
- ...

Methods

Specifying qualitative metrics

Fluency:

A fluent segment is one that is grammatically well-formed; contains correct spellings; adheres to the common use of terms, titles and names; contains properly capitalized letters; and is intuitively acceptable. Unfinished sentences also impact the fluency of a segment.

Extending definition from Ma and Cieri (2006)

Methods

Specifying qualitative metrics

Plausibility:

A plausible counterfactual segment adheres well to samples seen in the real data distribution, and the target sentiment of the target class. The changes made to the factual, considering the meaning and context of the edited words, should also fit the target domain.

Applying the definition from Altmeyer et al. (2024) to texts.

Experiments

Dataset:

Trillion Dollar Words (Shah et al., 2023) – excerpts from central bank communications.

Counterfactuals generated by each of three methods.

Qualitative assessment:

- 8 central bank employees (including Federal Reserve, Bank of England) judging the fluency and plausibility
- Crowdsourced evaluations of fluency from native speakers on Prolific

Results

Quantitative

Generator	Perplexity ↓	Perpl. ratio	Edit dist. ↓	Tree dist. ↓	Emb. dist. ↓	Implausib. ↓	Faithful. ↑	Succ. rate ↑
Polyjuice	90.98 (172.1)	1.80 (4.6)	0.31 (0.3)	19.67 (24.0)	**20.32** (3.7)	33.64 (4.6)	0.18 (0.4)	0.34 (0.5)
PPLM	**36.97** (16.9)	**0.78** (0.5)	0.69 (0.5)	36.94 (10.3)	20.88 (3.7)	**32.18** (4.0)	0.34 (0.6)	0.51 (0.5)
RELITC	100.94 (125.2)	1.67 (1.2)	**0.14** (0.1)	**10.72** (12.2)	21.96 (3.9)	33.30 (3.9)	**0.54** (0.6)	**0.74** (0.4)

Qualitative

	Annotators			
	Non-exp.	N-e. 5 CE	Expert	
Generator	Fluency	Fluency	Fluency	Plausibility
PPLM	2.86 (0.7)	2.48 (0.5)	2.26 (0.5)	1.83 (0.3)
Polyjuice	3.40 (0.9)	3.44 (0.7)	3.45 (0.9)	**2.45** (0.7)
RELITC	**3.43** (0.8)	**3.96** (0.5)	**3.90** (0.6)	2.12 (0.3)

Results

Do they align?

	Perplexity	Perp. ratio	Edit Dist.	Tree edit dist.	Emb. dist.	Implausib.
Fluency (non exp.)	-0.06 (0.2)	-0.03 (0.5)	**-0.21 (0.0002)**	**-0.21 (0.0003)**	0.03 (0.7)	0.06 (0.3)
Fluency (exp.)	0.12 (0.6)	0.14 (0.6)	**-0.56 (0.016)**	**-0.56 (0.015)**	-0.25 (0.3)	0.13 (0.3)
Plausibility	0.32 (0.2)	0.02 (0.9)	-0.12 (0.6)	-0.28 (0.3)	-0.12 (0.6)	0.28 (0.3)

Table 3: Pearson correlation coefficients and p-values between the quantitative and qualitative metric results.

Results

Expert comments:

- Polyjuice:
  - switches the subject entirely
  - lack of relevance in changes
- PPLM:
  - uses domain-specific words, but does it incorrectly
  - Makes the tone too conversational
- RELITC:
  - Can make the tone unclear or conversational
  - Can introduce domain-specific words that are incorrectly used

Conclusions

- Which counterfactual generator should be used?
- Trade-off between plausibility and faithfulness

Generator	Perplexity ↓	Perpl. ratio	Edit dist. ↓	Tree dist. ↓	Emb. dist. ↓	Implausib. ↓	Faithful. ↑	Succ. rate ↑
Pseudo-RAG	74.00 (38.8)	1.37 (0.5)	0.29 (0.1)	19.40 (11.5)	24.86 (4.0)	32.39 (2.9)	**0.36** (0.5)	**0.88**
Polyjuice	86.49 (79.9)	1.58 (1.3)	0.26 (0.3)	17.36 (15.3)	**24.78** (3.5)	**31.56** (2.7)	0.00 (0.4)	0.36
PPLM	**37.11** (15.2)	**0.76** (0.4)	0.56 (0.2)	37.48 (7.3)	24.97 (4.4)	32.09 (4.5)	0.04 (0.7)	0.52
RELITC	86.72 (71.6)	1.54 (1.0)	**0.13** (0.1)	**11.00** (7.0)	25.83 (3.7)	32.18 (3.1)	0.32 (0.6)	0.80

Table 5: Results for the quantitative metrics including the Pseudo-RAG method. Averaged over 25 factual-counterfactual rows.

- The need for using human evaluations
- The need for engaging classifiers in the process

Red Teaming for Large Language Models At Scale: Tackling Hallucinations on Mathematics Tasks

Aleksander Buszydlik[1,*] Karol Dobiczek[1,*] Michał Teodor Okoń[1,*]
Konrad Skublicki[1] Philip Lippmann[2] Jie Yang[2]

Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology, The Netherlands

ART of Safety: Workshop on Adversarial testing and Red-Teaming for generative AI,
AACL 2023

Motivation

- Red-Teaming, a practice in AI safety that aims to systematically find backdoors in Large Language Models to elicit irresponsible responses (Microsoft, 2023).
- Multiple methods for red-teaming using prompt engineering exist (Derczynski, 2023)
- Are we able to reduce hallucinations by using those methods?

Setup

Mathematical tasks:

- Likely to exist in the training data
- Not trivial for the LLM (tokenization)
- Easy to parse and evaluate

Two types of tasks:

- Elementary mathematics – addition and multiplication
- Algebraic reasoning – puzzles describing systems of equations

Two difficulties: products of numbers from 1 to 100 and 100 to 10 000

Red-Teaming by Prompting

Code – produce code, “simulate” its execution to provide the final answer.

Explanation – explain the understanding of the problem or rephrase it, and then provide the answer. Encourage the model to attempt step-by-step reasoning about the problem in natural language.

Impersonation – request that the model impersonates a celebrated mathematician to provide the answer in their words.

Restorying – frame the problem differently, ex.: write a manual, poem, or a theatre play with two actors.

Default – baseline.

Setup – Metrics

- Accuracy – how often is the model totally correct
- Levenshtein edit distance – edits, insertions, deletions
- Relative edit distance – relative to answer length
- Relative distance

Experiments

For each context, for each difficulty 200 queries on GPT-3.5-turbo and GPT-4 using OpenAI API.

OpenAI

For the Algebraic Reasoning task, we additionally add a variant where we provide a solved example.

Results – Examples

`DefaultContext:`

Prompt: What is the product of 9163 and 3584?

Model Response: [FINAL ANSWER] 32843072

Prediction: 32843072
Ground Truth: 32840192

`ExplanationContext:`

Prompt: Explain how you understand the following problem: Provide the product of 4263 and 3224. Then provide an answer to this problem.

Model Response: This problem is asking for the result of multiplying 4263 by 3224. To solve this, you simply need to multiply these two numbers together. [FINAL ANSWER] 13744232

Prediction: 13744232
Ground Truth: 13743912

`CodeContext:`

Prompt: Write code in Python to solve the following problem: Add 5002 and 6231. Then simulate the execution of this code to provide an answer to this problem.

Model Response:
Here is the Python code to solve the problem:

```
#Python code to add 5002 and 6231
def add_numbers(num1, num2):
    return num1 + num2

result = add_numbers(5002, 6231)
print(result)
```

Simulating the execution of this code:

```
#Python code to add 5002 and 6231
def add_numbers(num1, num2):
    return num1 + num2

result = add_numbers(5002, 6231)
print(result)
```

This will output: 11233 [FINAL ANSWER] The sum of 5002 and 6231 is 11233.

Prediction: 11233
Ground Truth: 11233

Results – Examples

`DefaultContext:`

Prompt: *You will now receive an example puzzle with an explanation how to solve it. Example puzzle: There are three people named Daniel, Anne and Elise. If Daniel had 3 times the pencils, together they would all have 147 pencils. If Anne had 3 times the pencils, together they would all have 179 pencils. If Elise had double the pencils, together with Daniel they would have 143 pencils. How many pencils does Elise have? Explanation: You can transform this puzzle into a system of equations: 3a + b + c = 147; a + 3b + c = 179; 2c + a = 143; You can find the answer by solving the equation for one variable in terms of the other variables, then substituting found expression into the other equations. Using this method you will arrive at the answer 63. This was the end of the example, use it to solve the following puzzle.* There are three people named Anne, Daniel and Charlie. Anne and Daniel together have 72 dollars. Anne and Charlie together have 114 dollars. If everyone had three times the amount of dollars, together they would have 381 dollars. How many dollars does Anne have?

Model response:
You can transform this puzzle into a system of equations:
a + d = 72; a + c = 114; 3a + 3d + 3c = 381;
From the first equation, we can express d in terms of a: d = 72 - a.
Substitute d into the third equation: 3a + 3(72 - a) + 3c = 381;
Simplify the equation: 3a + 216 - 3a + 3c = 381;
Simplify further: 216 + 3c = 381;
Solve for c: c = (381 - 216) / 3 = 55.
Substitute c into the second equation: a + 55 = 114;
Solve for a: a = 114 - 55 = 59.
So, Anne has 59 dollars.
[FINAL ANSWER] Anne has 59 dollars.

Prediction: 59
Ground Truth: 59

Results – Elementary mathematics GPT-3.5

Technique	Difficulty	Red teaming	Edit distance (characters) ↓	Relative edit distance (%) ↓	Relative distance (%) ↓	Accuracy (%) ↑
Code	Easy	✓	**0.000 (0.000)**	**0.000 (0.000)**	**0.000 (0.000)**	**100.0**
			0.000 (0.000)	**0.000 (0.000)**	**0.000 (0.000)**	**100.0**
	Hard	✓	1.490 (1.596)	19.1 (20.3)	0.0 (0.1)	**49.5**
			1.350 (1.410)	**17.6 (18.1)**	**0.0 (0.0)**	46.0
Explanation	Easy	✓	0.180 (0.740)	4.5 (18.5)	5.5 (22.0)	94.0
			0.000 (0.000)	**0.0 (0.0)**	**0.0 (0.0)**	**100.0**
	Hard	✓	1.565 (2.047)	20.9 (26.8)	14.1 (33.5)	54.1
			1.010 (1.300)	**12.8 (16.4)**	**0.0 (0.0)**	**59.0**
Impersonation	Easy	✓	0.265 (0.903)	6.6 (22.6)	5.3 (21.3)	91.5
			0.025 (0.211)	**0.6 (5.3)**	**0.0 (0.0)**	**98.5**
	Hard	✓	1.540 (1.928)	20.0 (24.6)	6.6 (22.9)	52.0
			1.175 (1.387)	**15.3 (17.9)**	**0.0 (0.1)**	**54.0**
Re-storying	Easy	✓	0.926 (1.382)	28.4 (41.2)	23.8 (53.1)	65.0
			0.000 (0.000)	**0.0 (0.0)**	**0.0 (0.0)**	**100.0**
	Hard	✓	3.827 (2.539)	56.1 (35.3)	55.0 (85.0)	18.8
			1.410 (1.429)	**17.9 (18.1)**	**0.0 (0.0)**	**46.0**

Table 1: Elementary mathematics experiment on `gpt-turbo-3.5` at `top_p=0.2`. SD is given in parentheses.

Results – Elementary mathematics GPT-4

Technique	Difficulty	Red teaming	Edit distance (characters) ↓	Relative edit distance (%) ↓	Relative distance (%) ↓	Accuracy (%) ↑
Code	Easy	✓	0.030 (0.263)	0.8 (6.6)	**0.0 (0.0)**	98.5
			0.015 (0.157)	**0.4 (3.9)**	**0.0 (0.0)**	**99.0**
	Hard	✓	1.310 (1.541)	16.7 (19.5)	4.5 (63.5)	**53.0**
			1.300 (1.520)	**16.6 (19.4)**	**0.0 (0.0)**	**53.0**
Explanation	Easy	✓	**0.005 (0.071)**	**0.1 (1.8)**	**0.0 (0.0)**	**99.5**
			0.005 (0.071)	**0.1 (1.8)**	**0.0 (0.0)**	**99.5**
	Hard	✓	**1.060 (1.465)**	**13.5 (18.6)**	**0.0 (0.0)**	**62.5**
			1.135 (1.545)	14.4 (19.6)	4.5 (63.5)	61.5
Impersonation	Easy	✓	**0.005 (0.071)**	**0.1 (1.8)**	**0.0 (0.0)**	**99.5**
			0.005 (0.071)	**0.1 (1.8)**	**0.0 (0.0)**	**99.5**
	Hard	✓	**1.345 (1.472)**	**17.1 (18.7)**	**0.0 (0.0)**	50.0
			1.360 (1.520)	17.3 (19.3)	4.5 (63.5)	**50.5**
Re-storying	Easy	✓	0.121 (0.580)	3.2 (15.6)	2.5 (15.4)	95.1
			0.000 (0.000)	**0.0 (0.0)**	**0.0 (0.0)**	**100.0**
	Hard	✓	2.165 (2.249)	30.0 (31.2)	76.4 (740.4)	42.8
			1.265 (1.531)	**16.1 (19.6)**	**0.0 (0.0)**	**55.0**

Table 3: Elementary mathematics experiment on `gpt-4` at `top_p=0.2`. SD is given in parentheses. Exact matches for *Easy* problems in "Explanation" and "Impersonation" are caused by the model making identical mistakes.

Results – Algebraic Reasoning GPT-3.5

Technique	Example	Edit distance (characters) ↓	Relative edit distance (%) ↓	Relative distance (%) ↓	Accuracy (%) ↑
Default	✓	**1.530 (1.367)**	**78.9 (72.7)**	**381.5 (1423.8)**	**38.3**
		1.758 (1.130)	95.2 (71.5)	386.8 (1261.1)	22.2
Code	✓	**1.815 (1.325)**	**97.9 (57.4)**	**182.5 (858.2)**	**6.6**
		1.848 (0.704)	99.3 (43.6)	205.4 (1445.1)	3.8
Explanation	✓	1.726 (1.443)	99.2 (92.3)	2073.6 (14097.8)	**32.0**
		1.710 (1.291)	**95.5 (79.1)**	**426.9 (1249.7)**	27.4
Impersonation	✓	**1.619 (1.188)**	**94.1 (78.8)**	**576.1 (1712.0)**	**27.1**
		2.131 (0.718)	120.3 (58.0)	717.7 (2183.3)	3.0
Re-storying	✓	**2.171 (0.990)**	**119.1 (67.5)**	739.1 (1830.0)	**10.9**
		2.215 (0.928)	119.2 (58.4)	**672.1 (2031.4)**	7.9

Table 2: Algebraic reasoning experiment on `gpt-turbo-3.5` at `top_p=0.2`. SD is given in parentheses.

Results – Algebraic Reasoning GPT-4

Technique	Example	Edit distance (characters) ↓	Relative edit distance (%) ↓	Relative distance (%) ↓	Accuracy (%) ↑
Default	✓	0.960 (1.127)	50.2 (59.8)	**70.7 (210.8)**	51.0
		0.897 (1.184)	**49.8 (69.1)**	128.0 (444.7)	**59.3**
Code	✓	1.645 (0.744)	87.1 (45.3)	60.9 (112.4)	10.7
		1.576 (0.776)	**82.6 (44.4)**	**58.9 (116.8)**	**12.8**
Explanation	✓	**0.851 (1.074)**	**44.6 (58.3)**	93.2 (539.0)	55.9
		0.901 (1.194)	46.9 (61.9)	**92.4 (278.2)**	**58.2**
Impersonation	✓	**1.056 (1.200)**	**57.6 (68.1)**	**148.3 (486.8)**	51.7
		1.108 (1.296)	58.6 (70.6)	191.7 (639.6)	**51.8**
Re-storying	✓	**1.331 (1.182)**	**75.7 (72.6)**	**261.9 (920.4)**	**37.5**
		1.773 (1.116)	98.2 (69.7)	418.8 (891.6)	21.7

Table 4: Algebraic reasoning experiment on `gpt-4` at `top_p=0.2`. SD is given in parentheses.

Key Takeaways

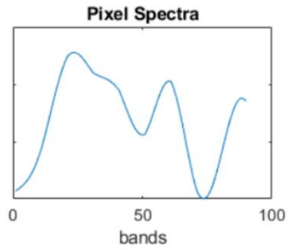
- Red-Teaming techniques do not necessarily help with hallucinations
- Structuring the responses of GPT models might help
- Giving the model examples improves performance
    - Might suggest that GPT models have some capacity to transfer knowledge

Current and Future Directions

Future

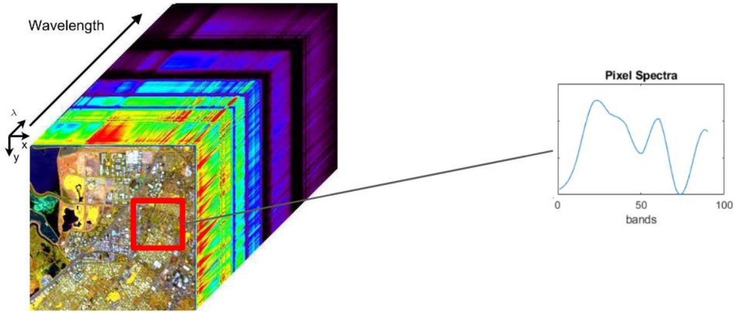
Using prototype models for EO

- Model attributions might not make the user "see" the whole picture



???

- Make the model itself "tell" the user what it considers



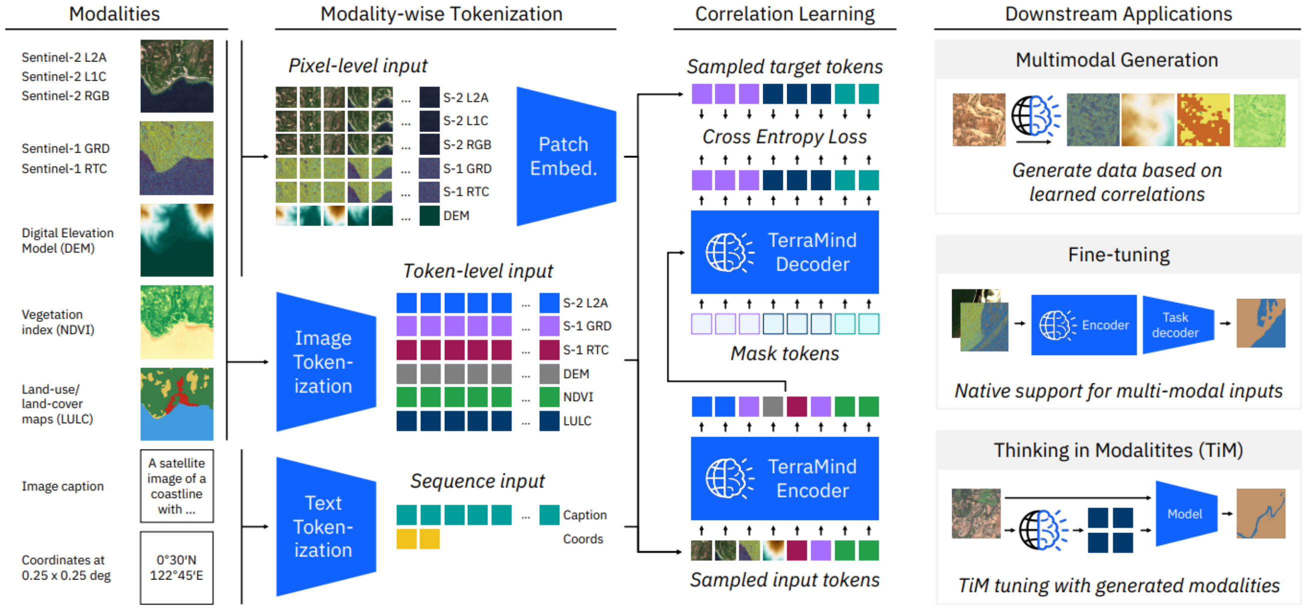
Future

"Red-teaming" foundational EO models with synthetic CEs

- Perturbations
- Noise
- Translation of certain features

TerraMind: Large-Scale Generative Multimodality for Earth Observation

Johannes Jakubik[1,*] Felix Yang[1,2*] Benedikt Blumenstiel[1,*] Erik Scheurer[3]
Rocco Sedona[3] Stefano Maurogiovanni[3,6] Jente Bosmans[4] Nikolaos Dionelis[4]
Valerio Marsocci[4] Niklas Kopp[1] Rahul Ramachandran[5] Paolo Fraccaro[1,†]
Thomas Brunschwiler[1,†] Gabriele Cavallaro[3,6,†] Juan Bernabe-Moreno[1,†] Nicolas Longépé[4,†]
[1]IBM Research – Europe [2]ETH Zurich [3]Forschungszentrum Jülich
[4]European Space Agency Φ-Lab [5]NASA IMPACT [6]University of Iceland
johannes.jakubik1@ibm.com



Future

Expert domain knowledge in XAI

- Human (expert)-in-the-loop optimization for XAI processes like counterfactual generation
- Embedding expert knowledge in models like GNNs

Thank you for your attention!