Security risks with LLMs

CS 685, Spring 2023

Advanced Natural Language Processing

Mohit Iyyer

College of Information and Computer Sciences

University of Massachusetts Amherst

many slides from Kalpesh Krishna

We interact with LLMs mainly through blackbox APIs

- Generally no access to hidden states, next-word probability distributions, or even basic info like model size or architecture

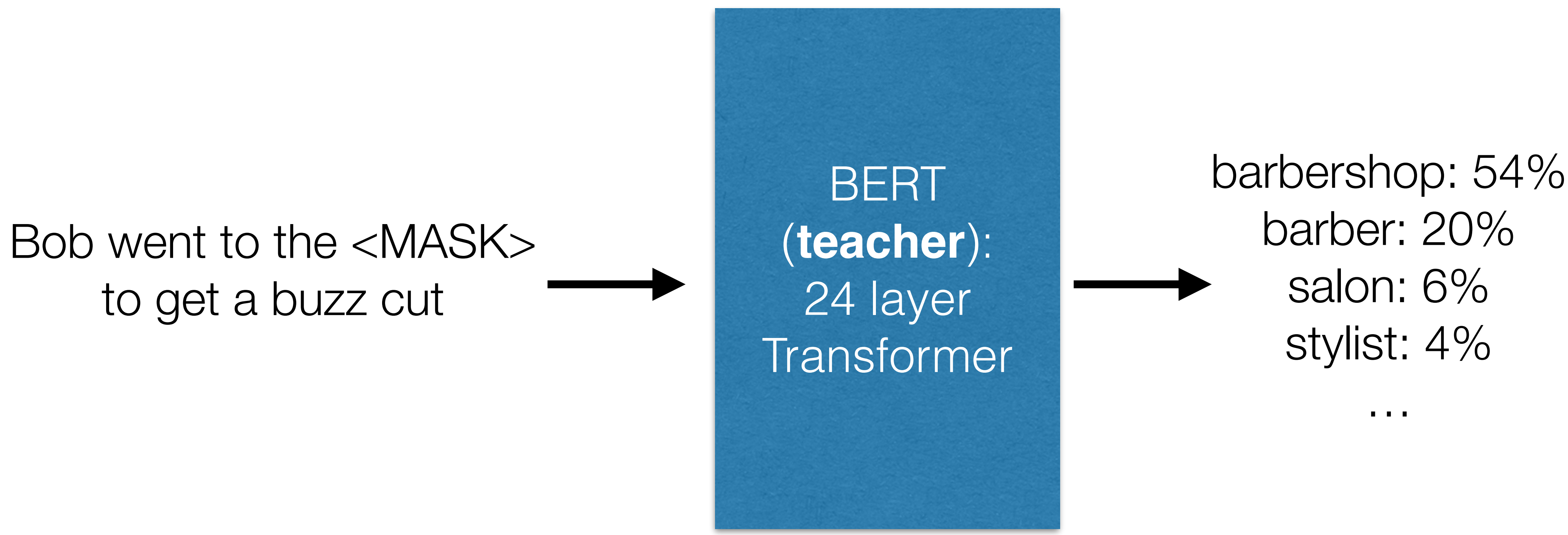
- In this setting, API providers should worry about their models being **extracted** or **distilled**

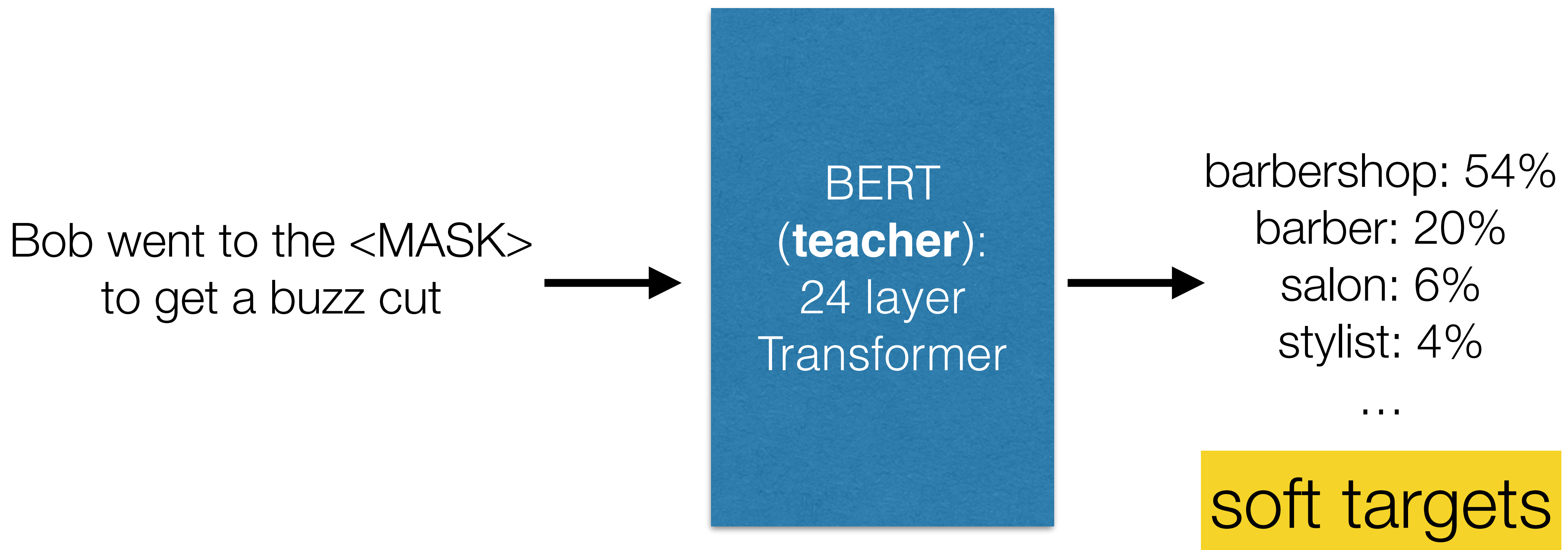
- Imagine you have a small LM. How can you use GPT-4 to improve its performance?

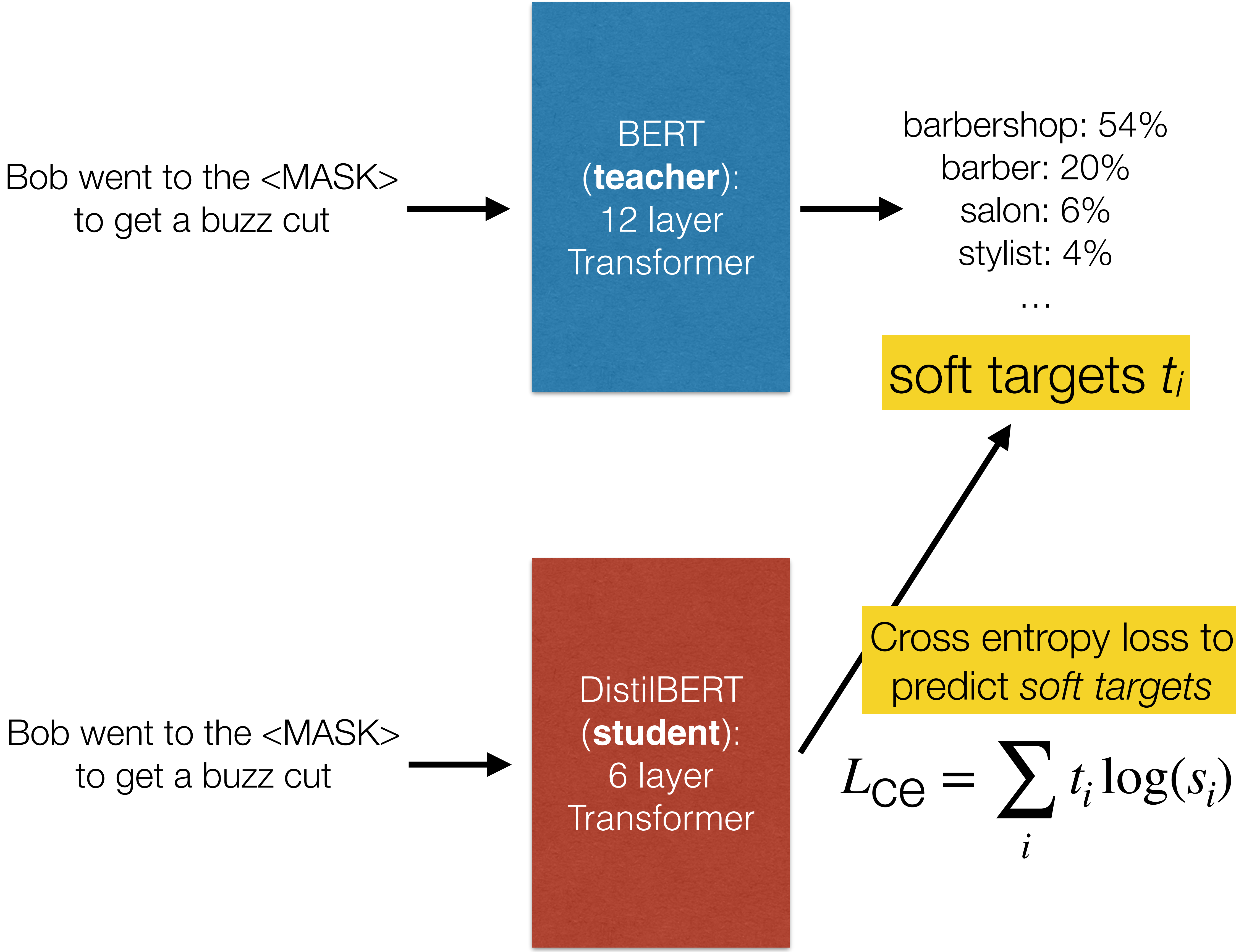
Knowledge distillation:

A small model (the **student**) is trained to mimic the predictions of a much larger pretrained model (the **teacher**)

Bucila et al., 2006; Hinton et al., 2015


Bob went to the <MASK>
to get a buzz cut
BERT
(**teacher**):
24 layer
Transformer
barbershop: 54%
barber: 20%
salon: 6%
stylist: 4%
...


Bob went to the <MASK> to get a buzz cut
BERT (**teacher**): 24 layer Transformer
barbershop: 54%
barber: 20%
salon: 6%
stylist: 4%
…
soft targets



Bob went to the <MASK> to get a buzz cut
BERT (**teacher**): 12 layer Transformer
barbershop: 54%
barber: 20%
salon: 6%
stylist: 4%
...
soft targets t_i
Bob went to the <MASK> to get a buzz cut
DistilBERT (**student**): 6 layer Transformer
Cross entropy loss to predict *soft targets*
$L_{ce} = \sum_i t_i \log(s_i)$

Instead of “one-hot” ground-truth, we have a full predicted distribution

- More information encoded in the target prediction than just the “correct” word
- Relative order of even low probability words (e.g., “church” vs “and” in the previous example) tells us some information
  - e.g., that the <MASK> is likely to be a noun and refer to a location, not a function word

Table 1: **DistilBERT retains 97% of BERT performance.** Comparison on the dev sets of the GLUE benchmark. ELMo results as reported by the authors. BERT and DistilBERT results are the medians of 5 runs with different seeds.

Model	**Score**	CoLA	MNLI	MRPC	QNLI	QQP	RTE	SST-2	STS-B	WNLI
ELMo	68.7	44.1	68.6	76.6	71.1	86.2	53.4	91.5	70.4	56.3
BERT-base	79.5	56.3	86.7	88.6	91.8	89.6	69.3	92.7	89.0	53.5
DistilBERT	77.0	51.3	82.2	87.5	89.2	88.5	59.9	91.3	86.9	56.3

Can also distill other parts of the teacher, not just its final predictions!

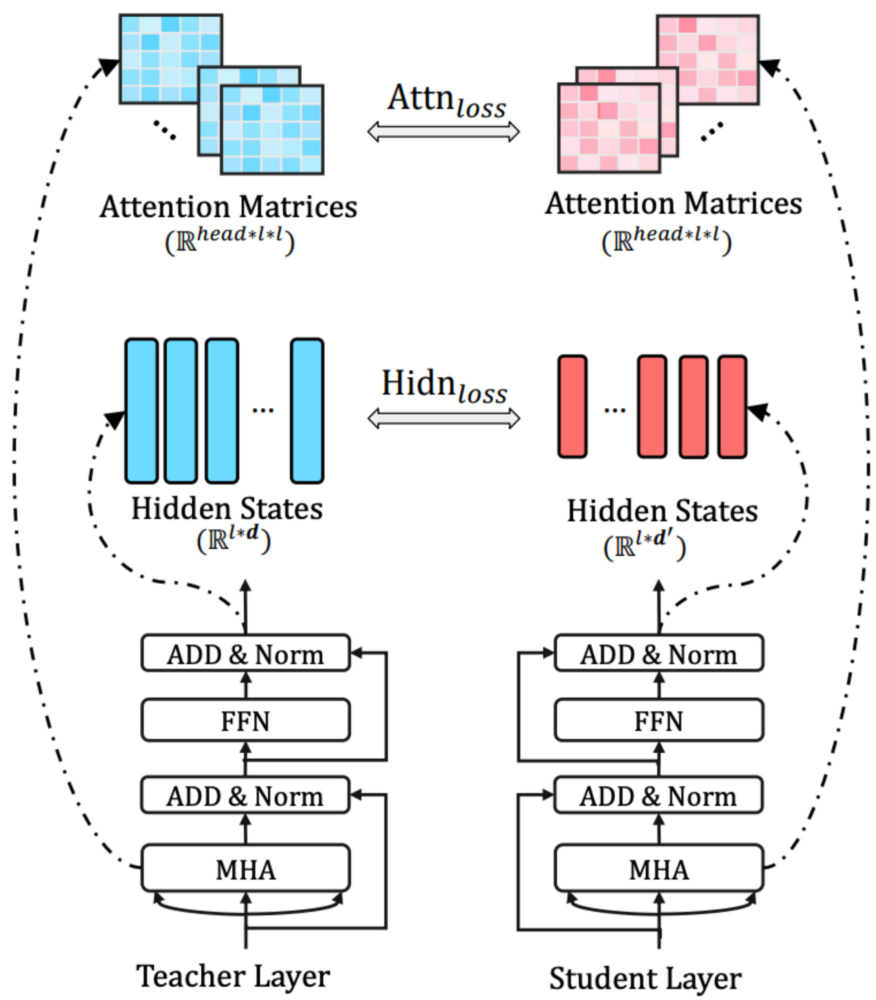
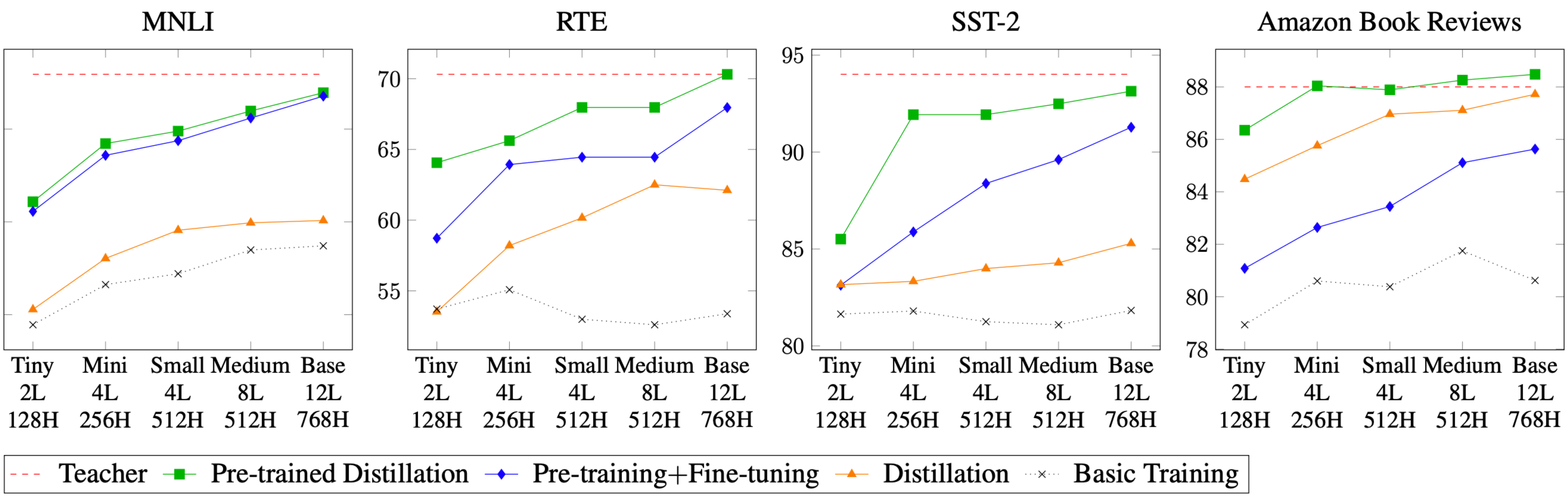


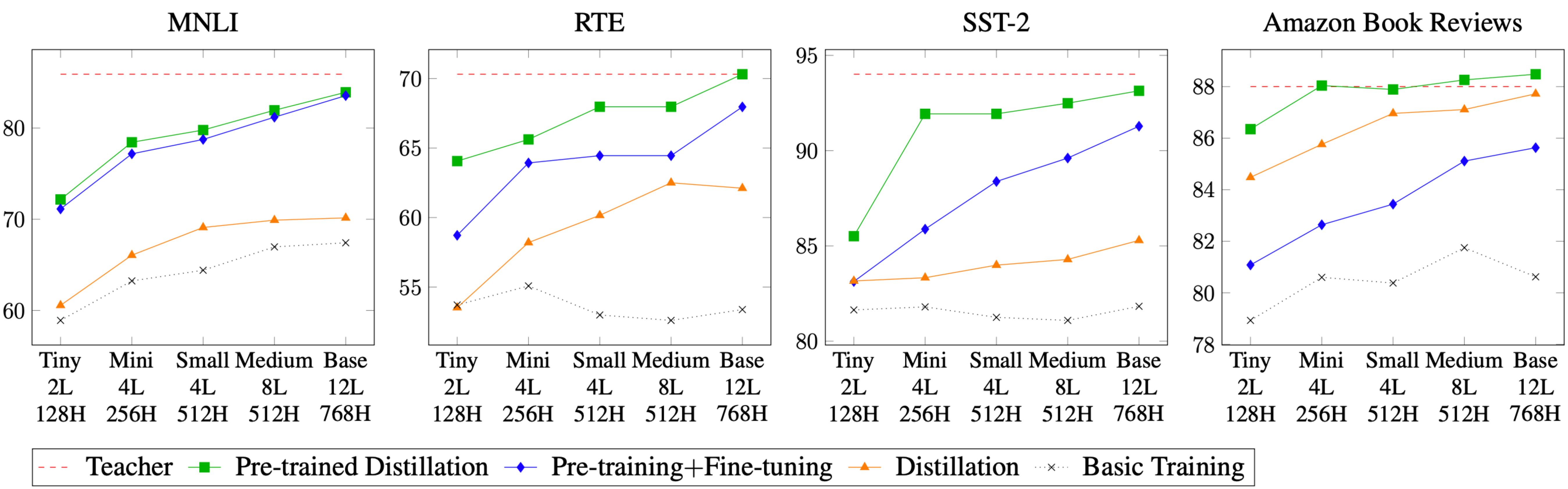
Figure 2: The details of Transformer-layer distillation consisting of Attn$_{loss}$(attention based distillation) and Hidn$_{loss}$(hidden states based distillation).

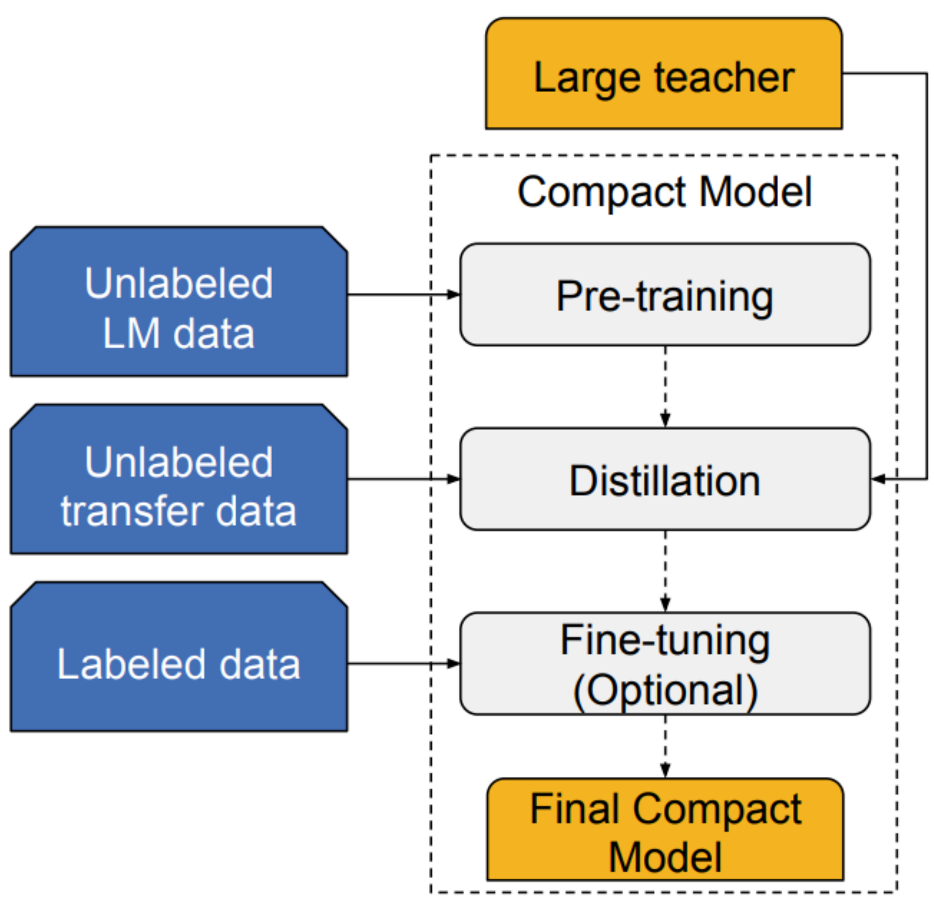
Jiao et al., 2020 ("TinyBERT")

Distillation helps significantly over just training the small model from scratch



Turc et al., 2019 ("Well-read students learn better")



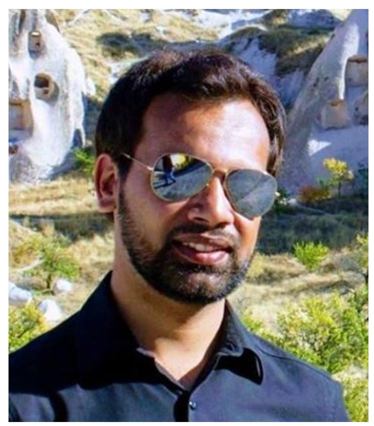


Turc et al., 2019 ("Well-read students learn better")

What if you only have access to the model's argmax prediction, and you also don't have access to its training data?

Thieves on Sesame Street!
Model Extraction of BERT-based APIs


Kalpesh Krishna[1]


Gaurav S. Tomar[2]


Ankur P. Parikh[2]


Nicolas Papernot[2]


Mohit Iyyer[1]

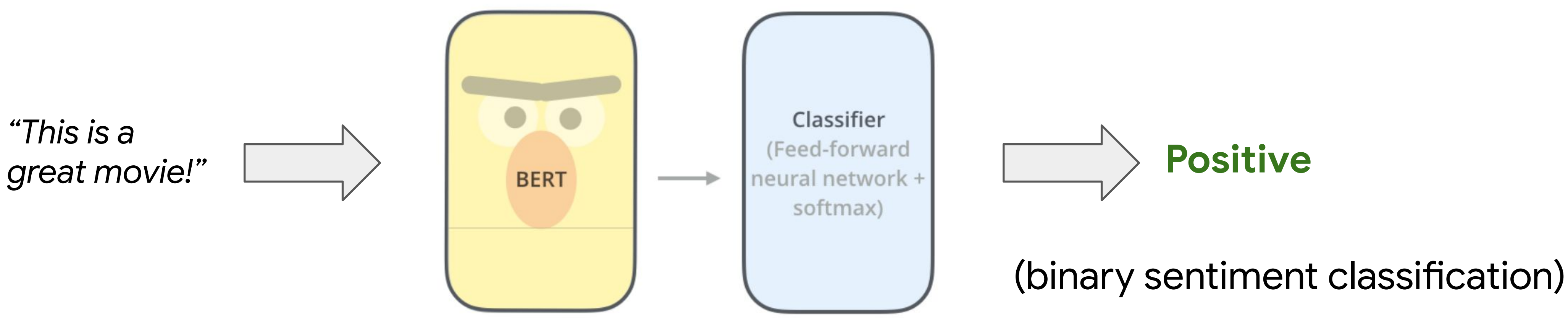
[1] UMass Amherst

[2] Google AI

Work done during an internship at Google AI Language.

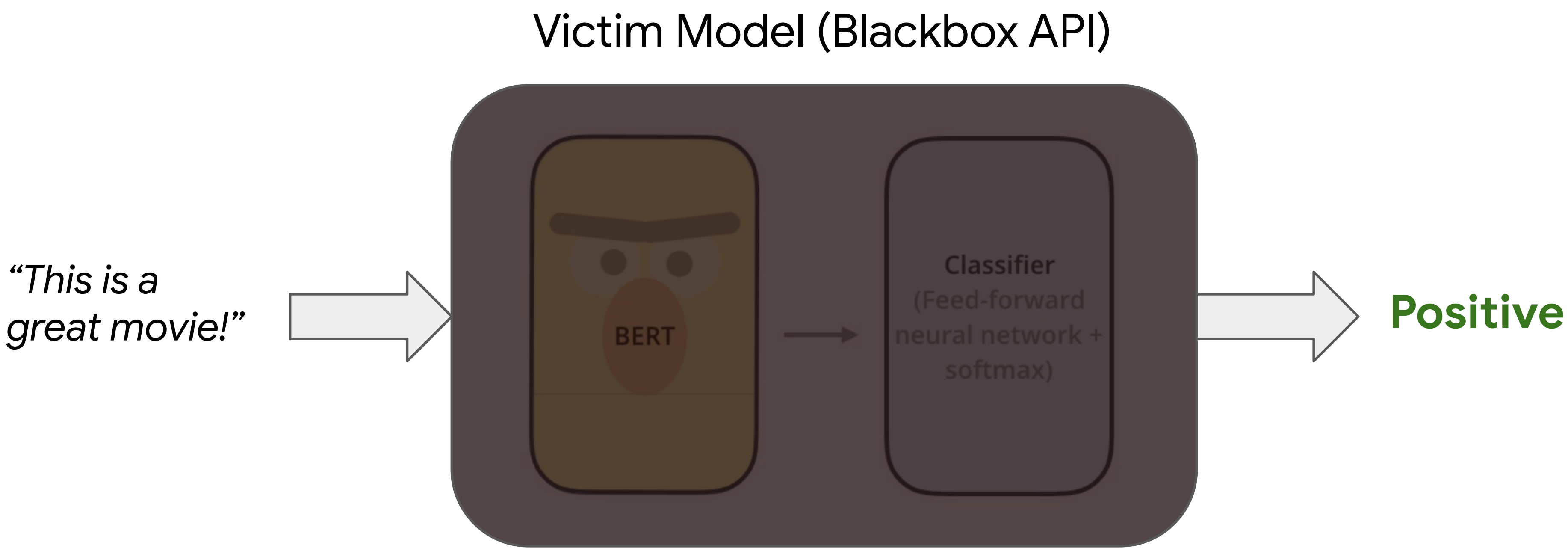
What are model extraction attacks?

Victim Model (Blackbox API)



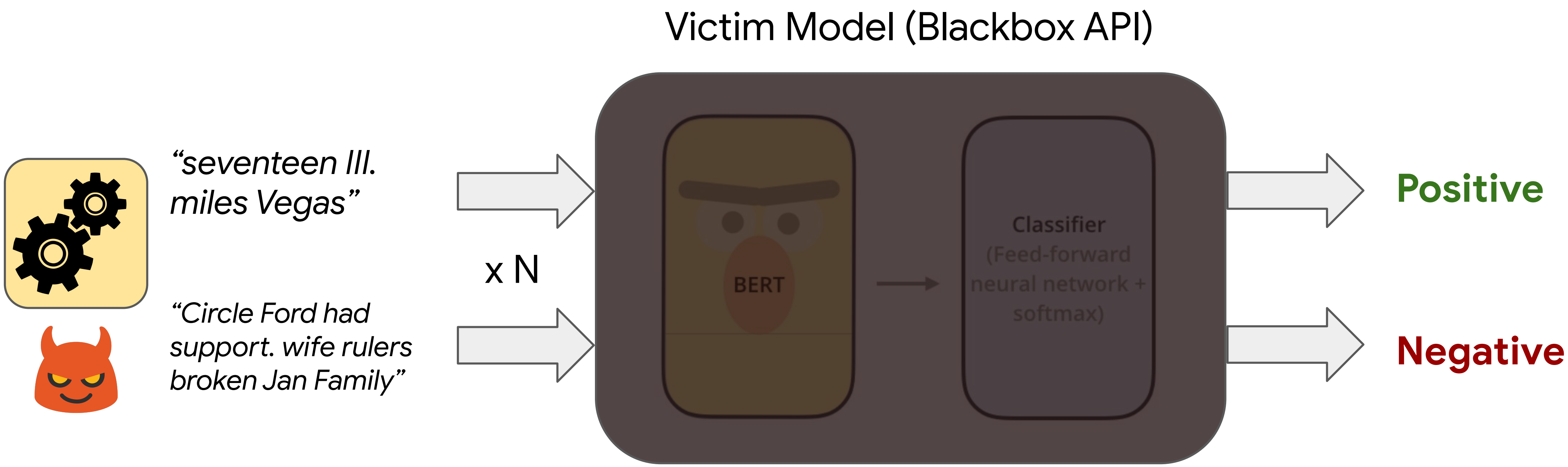
A company trains a binary sentiment classifier based on BERT

Photo credits - http://jalammar.github.io/illustrated-bert/

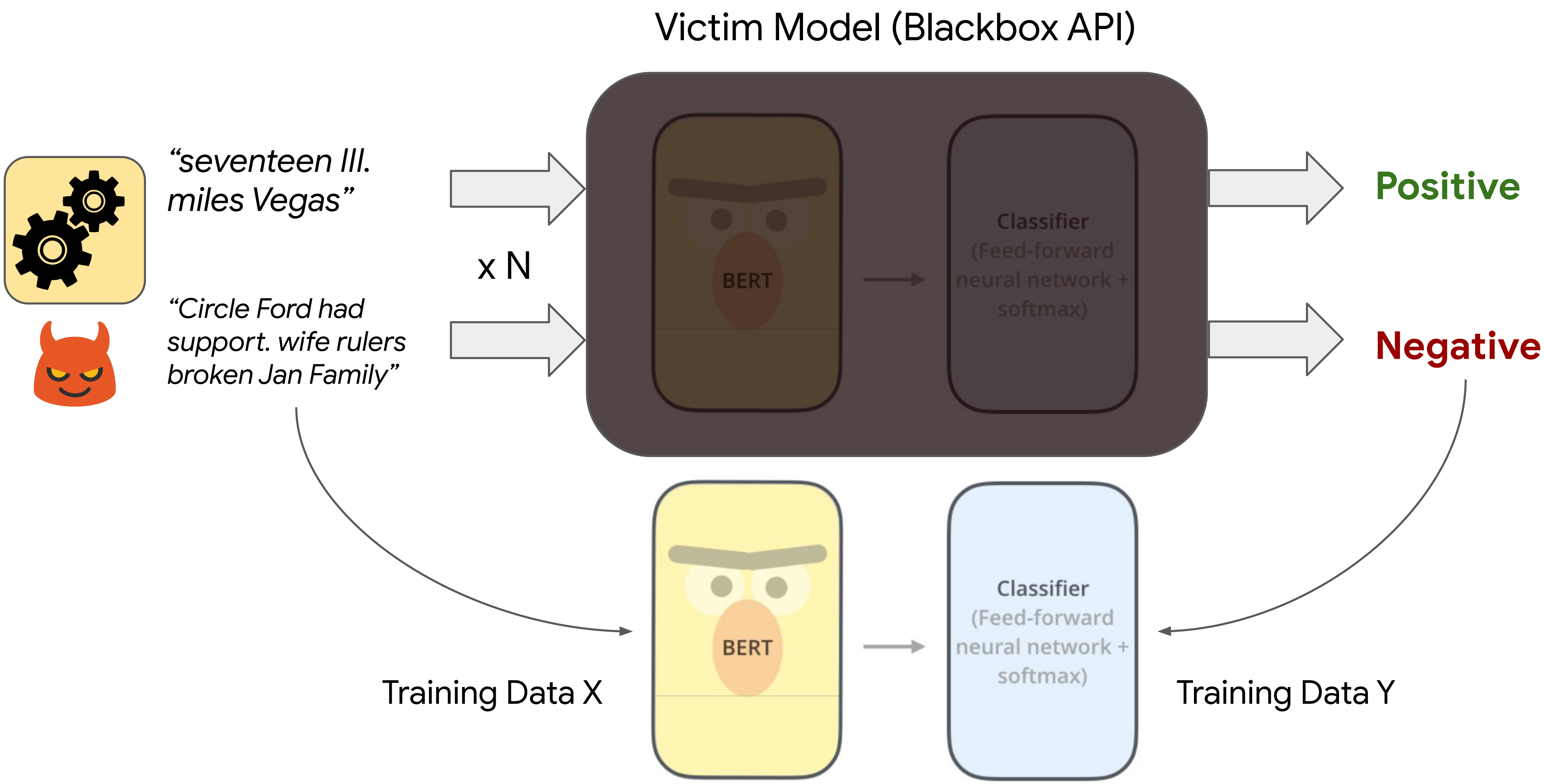


It is released as a black-box API (the “victim model”)

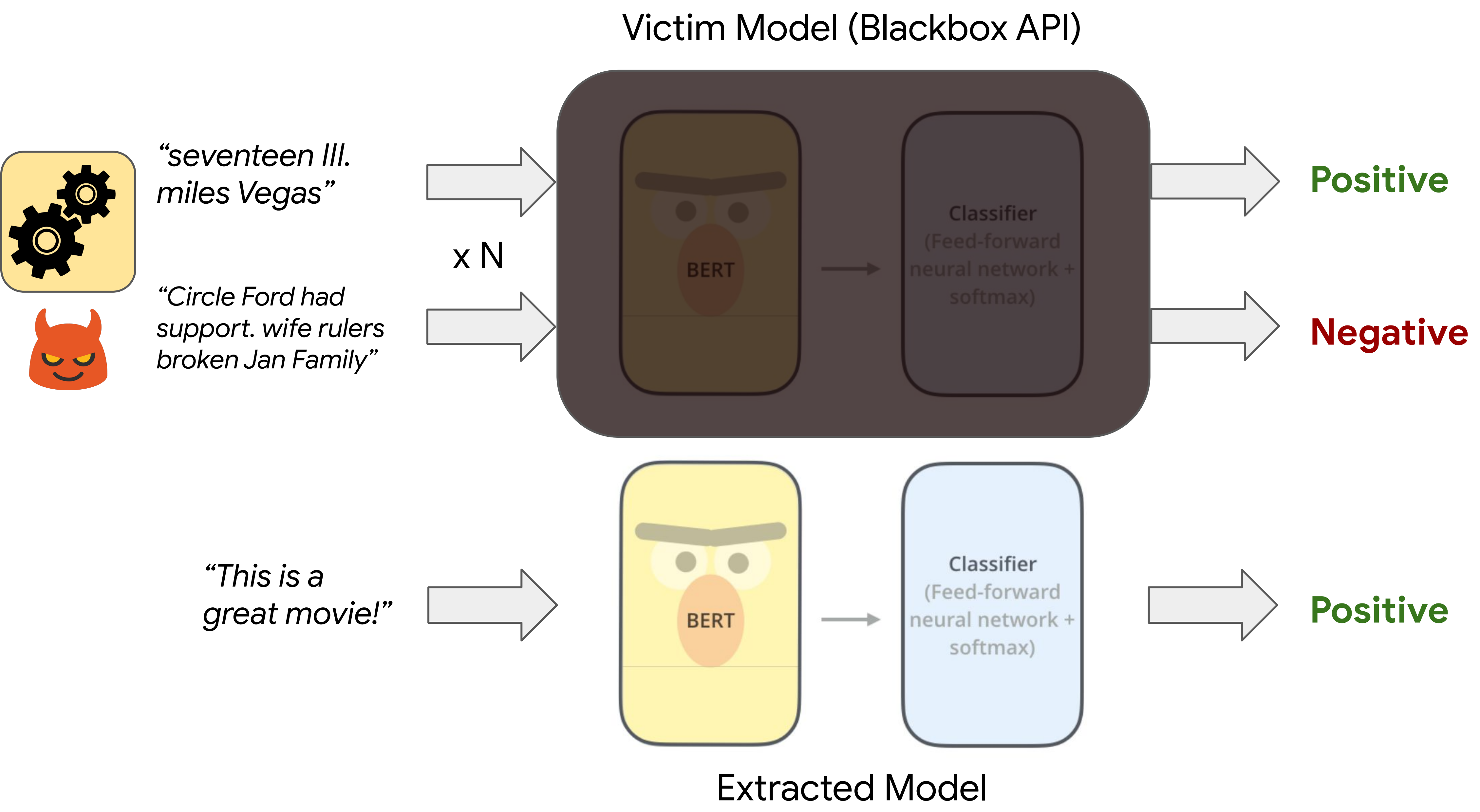
A malicious user generates many queries
(in this work, **random gibberish sequences of words**)



The attacker queries the API with the generated inputs and collects the labels



The collected data is used to train a “copy” of the model

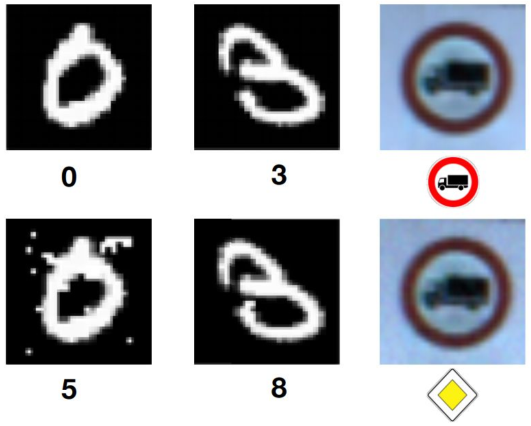


The stolen copy ("extracted model") works well on real data

Why is model extraction a problem?

Theft of intellectual property

Leakage of original training data



Adversarial example generation

Image credits - [Papernot et al. 2017](), [Fredrikson et al. 2015]()

These attacks are economically practical

Google Cloud Natural Language API cost <= $1.00 per 1000 API calls.

Dataset	**Size**	**Upperbound Price**
SST2 (sentiment classify)	67349 sentences	$62.35
Switchboard (speech)	300 hours	$430.56
Translation	1 million sentences (100 characters each)	$2000.00

Smart attackers can scrape APIs like Google Translate for free

https://cloud.google.com/products/calculator/

How is this different from distillation?

No training data

Goal is theft, not compression

We attack BERT models for,

1) sentiment classification (SST2)
2) natural language inference (MNLI)
3) question answering (SQuAD, BoolQ)

We use two query generators - RANDOM & WIKI

RANDOM
(gibberish sequences of words sampled from a fixed vocabulary)

1. cent 1977, preparation (120 remote Program finance add broader protection
2. Mike zone fights Woods Second State known, defined come

WIKI
(sentences from Wikipedia)

1. The unique glass chapel made public and press viewing of the wedding easy.
2. Wrapped in Red was first released internationally on October 25, 2013.

For multi-input tasks (like question answering) we ensure inputs are related to each other

RANDOM Paragraph: as and conditions Toxostoma storm, The interpreted. Glowworm separation Leading killed Papps wall upcoming Michael Highway that of on other Engine On to Washington Kazim of consisted the ” further and into touchdown(AADT), Territory fourth of h; advocacy its Jade woman ”lit that spin. Orange the EP season her General of the

For multi-input tasks (like question answering) we ensure inputs are related to each other

RANDOM Paragraph: as and conditions Toxostoma storm, The interpreted. Glowworm separation Leading killed Papps wall upcoming Michael Highway that of on other Engine On to Washington Kazim of consisted the ” further and into touchdown(AADT), Territory fourth of h; advocacy its Jade woman ”lit that spin. Orange the EP season her General of the

RANDOM Question: Kazim Kazim further as and Glowworm upcoming interpreted. its spin. Michael as

Results - attacks are effective

	# of Queries	SST2 (%)	MNLI (%)	SQUAD (F1)
API / Victim Model	1x	93.1	85.8	90.6
RANDOM	1x	90.1	76.3	79.1
RANDOM	upto 10x	90.5	78.5	85.8
WIKI	1x	91.4	77.8	86.1
WIKI	upto 10x	91.7	79.3	89.4

A BERT model trained on the real SQuAD data gets 90.6 F1

Results - attacks are effective

	# of Queries	SST2 (%)	MNLI (%)	SQUAD (F1)
API / Victim Model	1x	93.1	85.8	90.6
RANDOM	1x	90.1	76.3	79.1
RANDOM	upto 10x	90.5	78.5	85.8
WIKI	1x	91.4	77.8	86.1
WIKI	upto 10x	91.7	79.3	89.4

RANDOM achieves 85.8 F1 (**~95%** performance) **without seeing a single grammatically valid paragraph or question** during training

Results - attacks are effective

	# of Queries	SST2 (%)	MNLI (%)	SQUAD (F1)
API / Victim Model	1x	93.1	85.8	90.6
RANDOM	1x	90.1	76.3	79.1
RANDOM	upto 10x	90.5	78.5	85.8
WIKI	1x	91.4	77.8	86.1
WIKI	upto 10x	91.7	79.3	89.4

WIKI achieves 89.4 F1 (**~99%** performance) without seeing a single grammatically valid question during training

Key findings from experimental analysis

- better pretraining ⇒ better model extraction
- WIKI / RANDOM queries closer to the victim model's learnt distribution are more effective

What about large language models?

How to extract an LLM served via a blackbox API:

1. Acquire a small open-source pretrained language model (e.g., Meta's LLaMA)
2. Extract fine-tuning data from API via e.g., self-instruct (Wang et al., 2022)
3. Fine-tune the pretrained model from step 1 with the data from step 2

Proof of concept: Alpaca from Stanford, Vicuna (fine-tuned on ChatGPT interactions)

Self-instruct demo

Paraphrasing evades detectors of AI-generated text, but retrieval is an effective defense

Kalpesh Krishna[♠,*] **Yixiao Song**[♠] **Marzena Karpinska**[♠]
John Wieting[◇,†] **Mohit Iyyer**[♠,†]

[♠]University of Massachusetts Amherst, [◇]Google Research

{kalpesh,mkarpinska,miyyer}@cs.umass.edu

yixiaosong@umass.edu jwieting@google.com



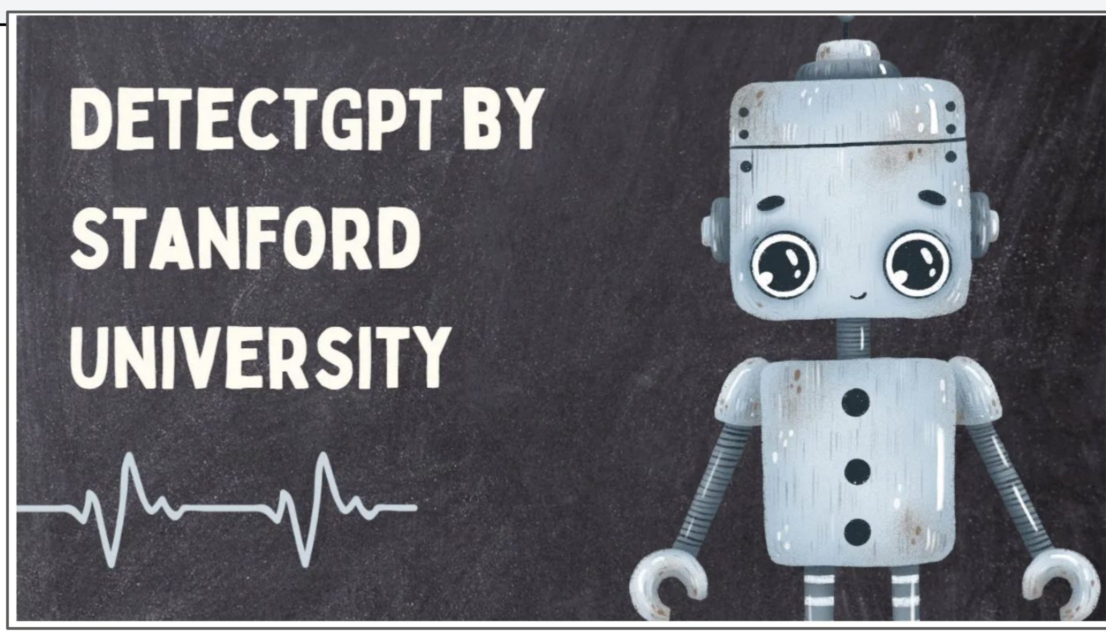

Turnitin's ChatGPT and AI wri
detection capabilities go live
98pc confidence rating (Austr
New Zealand)

New capabilities in the existing Turnitin workflow give educators highly accurate insights into t
text for more than 62 million students.

Wednesday 5 April 2023

New AI classifier for indicating AI-written text

We're launching a classifier trained to distinguish
ext.



Try GPTZero 👇

Pre-fill with examples:

HUMAN | AI | MIXED CONTENT

particularly the emission of greenhouse gases into the atmosphere.
The most significant greenhouse gas is carbon dioxide, which is primarily produced by burning fossil fuels such as coal, oil, and gas.
The consequences of climate change are already visible in the form of rising temperatures, melting glaciers and ice caps, and more frequent extreme weather events such as hurricanes, droughts, and floods.

or, choose a file to upload

CHOOSE FILE | No file chosen

Accepted file types: pdf, docx, txt

☑ I agree to the terms of service

GET RESULTS

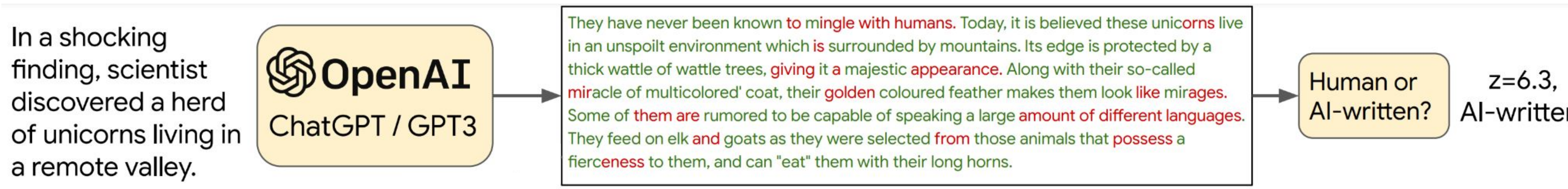
Your text is likely to be written entirely by AI

Watermarking

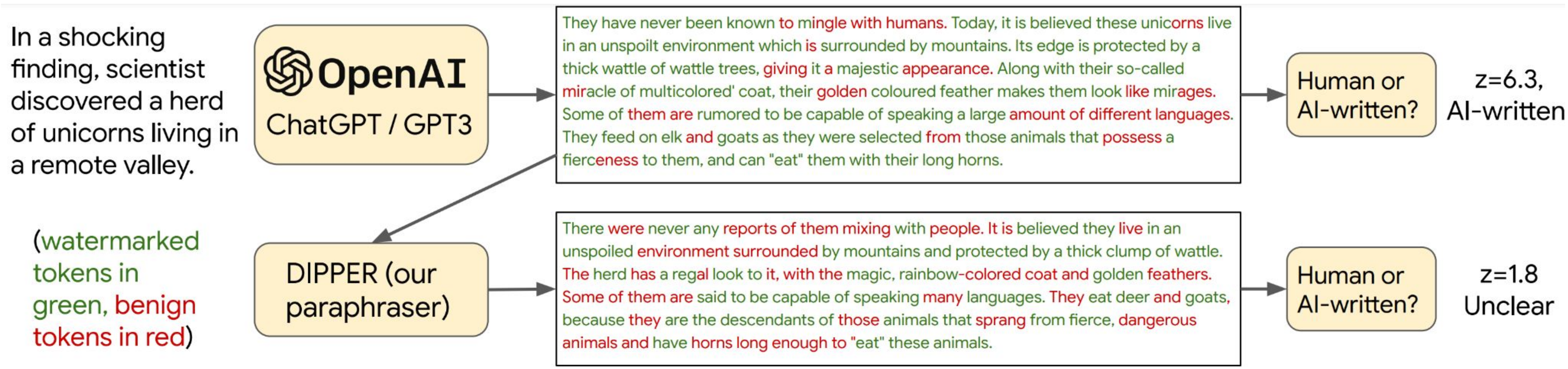
Prompt	Num tokens	Z-score	p-value
...The watermark detection algorithm can be made public, enabling third parties (e.g., social media platforms) to run it themselves, or it can be kept private and run behind an API. We seek a watermark with the following properties:			
No watermark Extremely efficient on average term lengths and word frequencies on synthetic, microamount text (as little as 25 words) Very small and low-resource key/hash (e.g., 140 bits per key is sufficient for 99.999999999% of the Synthetic Internet	56	.31	.38
With watermark - minimal marginal probability for a detection attempt. - Good speech frequency and energy rate reduction. - messages indiscernible to humans. - easy for humans to verify.	36	7.4	6e-14

Kirchenbauer et al., 2023

Do these methods work?



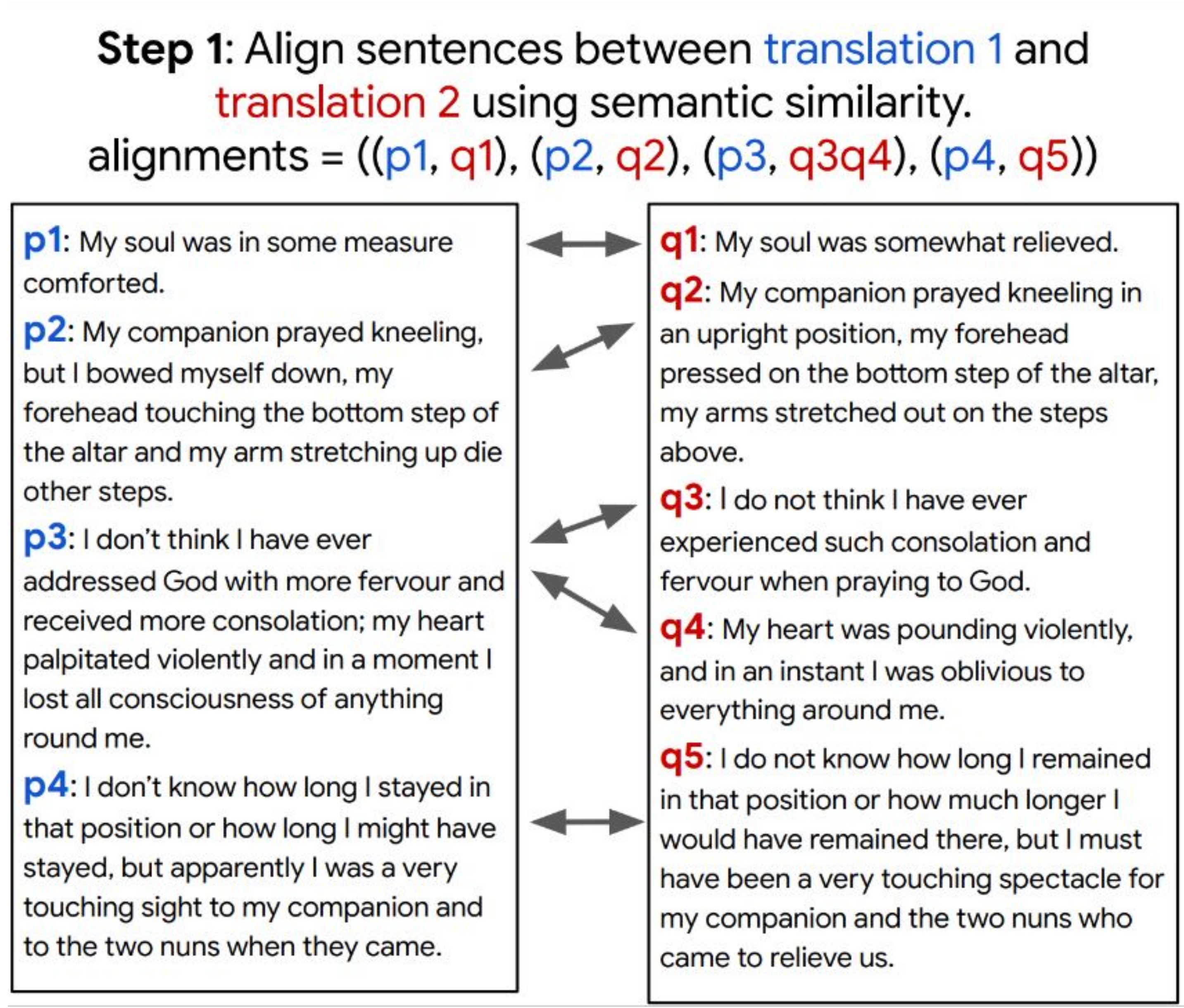
Paraphrasing easily evades them!

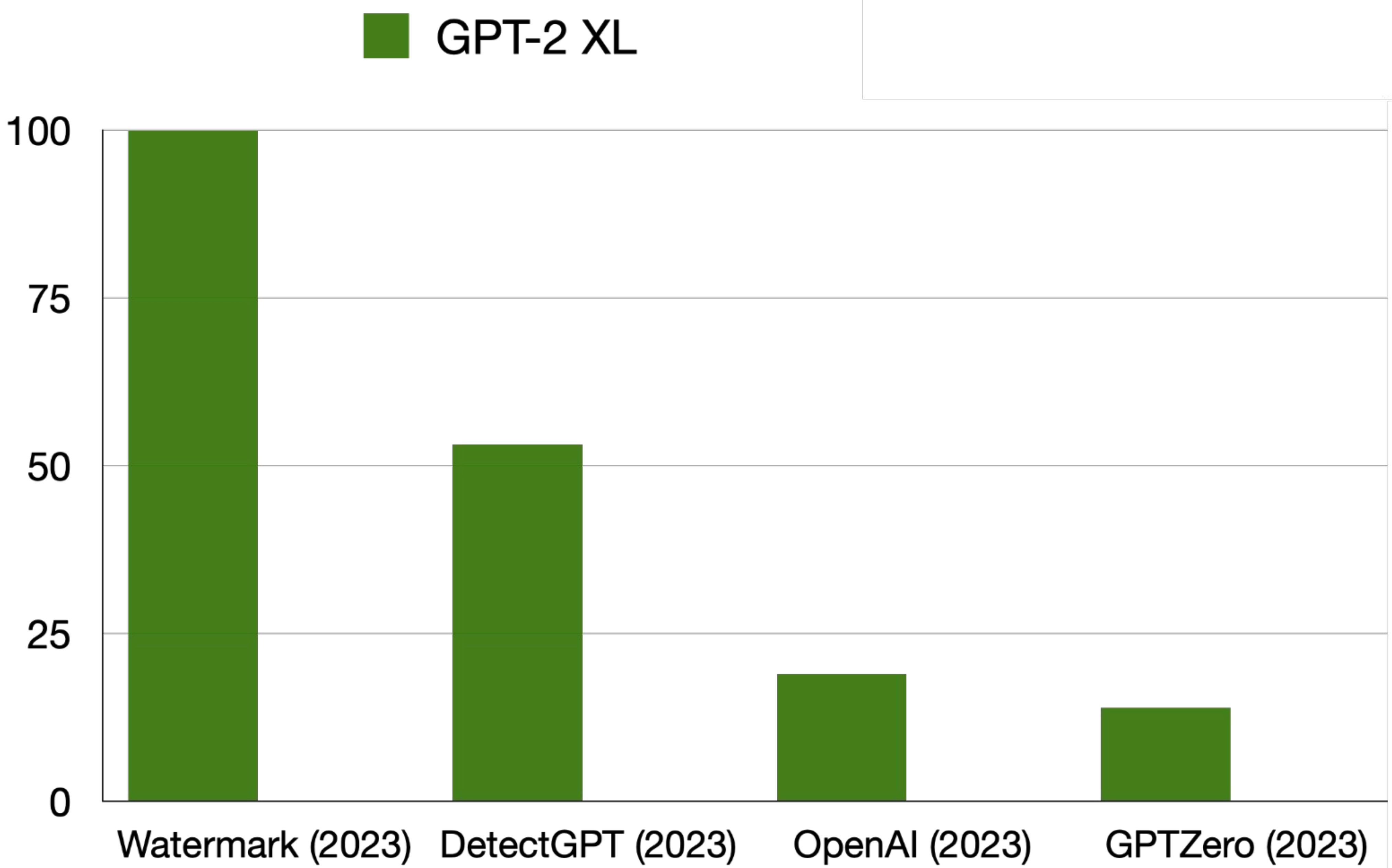


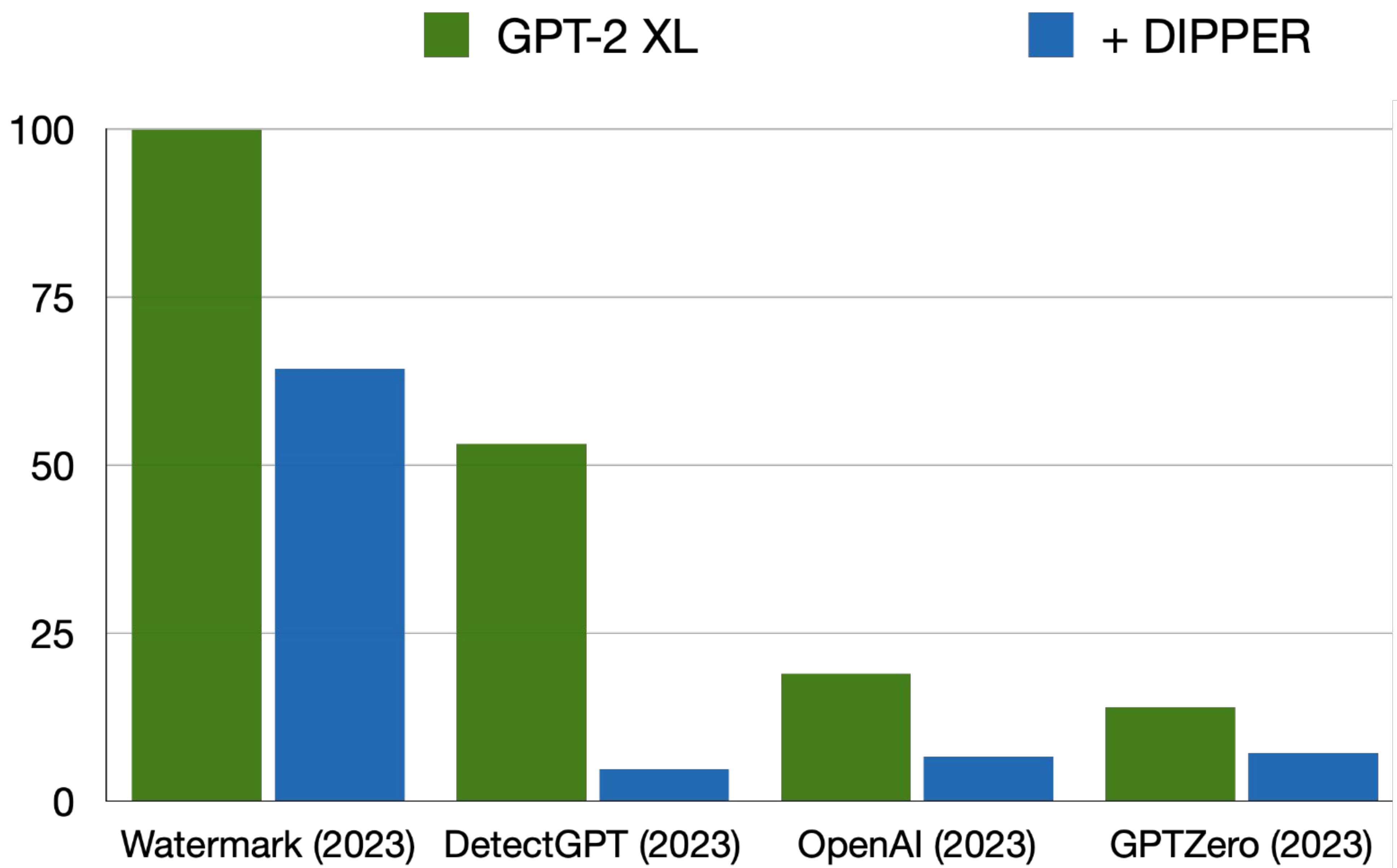
Paraphrasing attacks

- Let’s say an adversary wants to evade one of these detectors
- They can take the output of ChatGPT/GPT-4, and then pass it through an *external* paraphrasing model
    - Useful for paraphraser to be **controllable**, as adversary may want to make minimal changes needed to fool detector (e.g., lexical swaps, or content reordering)
    - Useful for paraphraser to be **context-aware**, so it can condition paraphrases on discourse-level information (e.g., prompts)

Building DIPPER



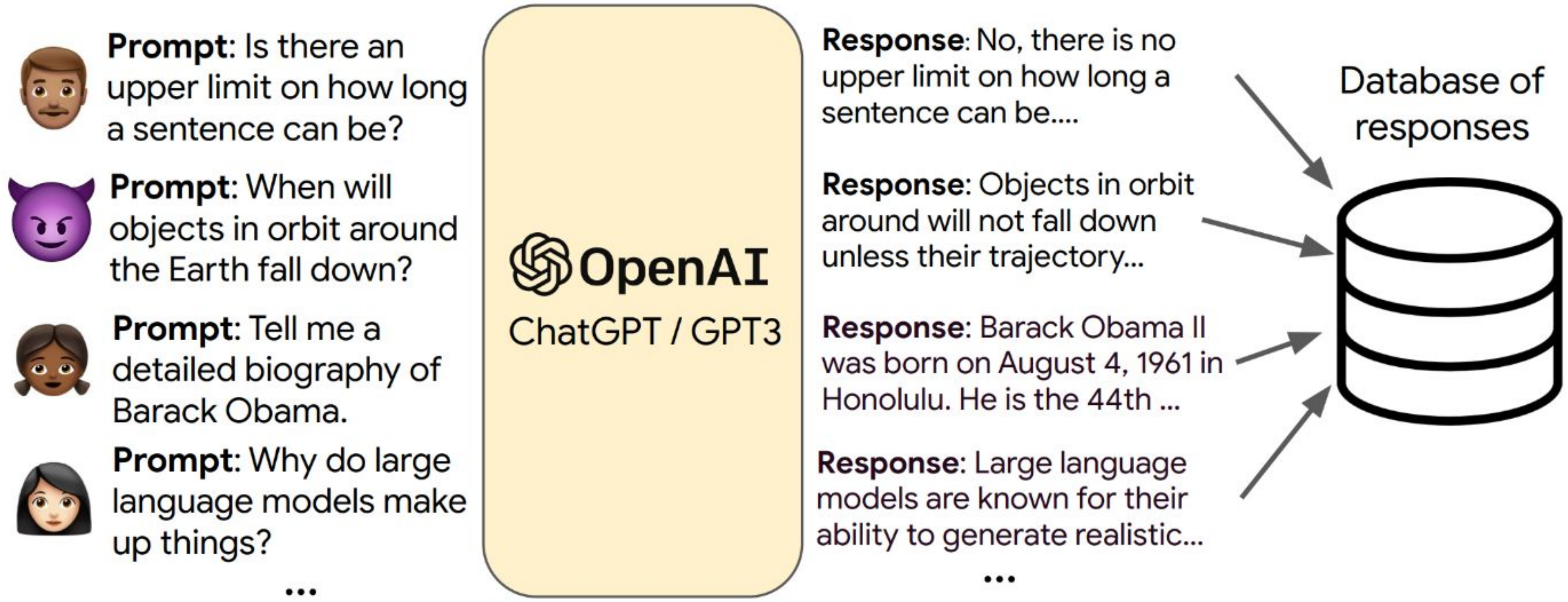

GPT-2 XL
100
75
50
25
0
Watermark (2023)
DetectGPT (2023)
OpenAI (2023)
GPTZero (2023)


GPT-2 XL
+ DIPPER
100
75
50
25
0
Watermark (2023)
DetectGPT (2023)
OpenAI (2023)
GPTZero (2023)

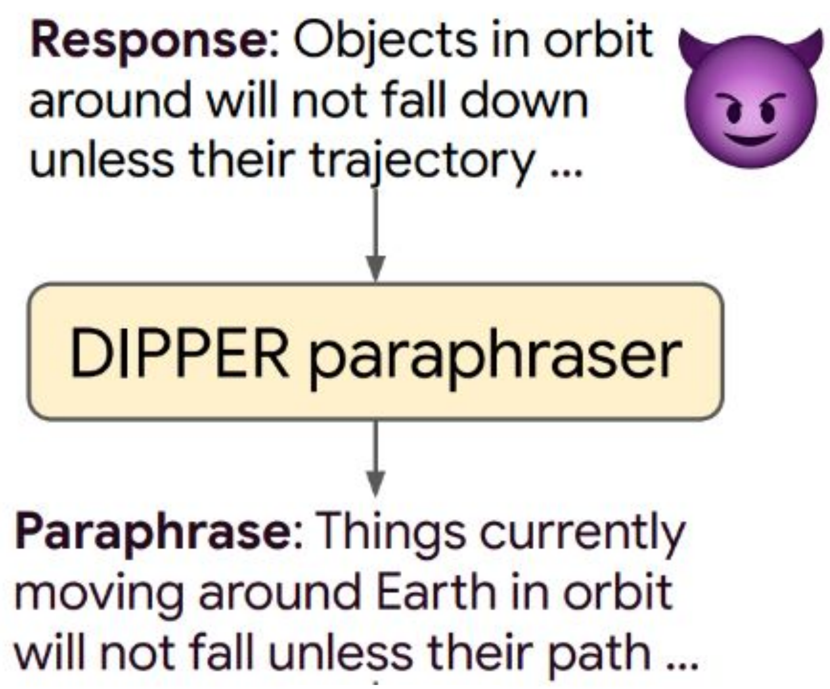
Defending against paraphrasing attacks?

- We propose a simple *retrieval-based* defense that must be maintained by an LLM API provider (e.g., OpenAI)
- Given a candidate text, it will retrieve semantically-similar generations from a database of all the text it has ever generated before
- A candidate is detected as AI-generated if it scores above some similarity threshold

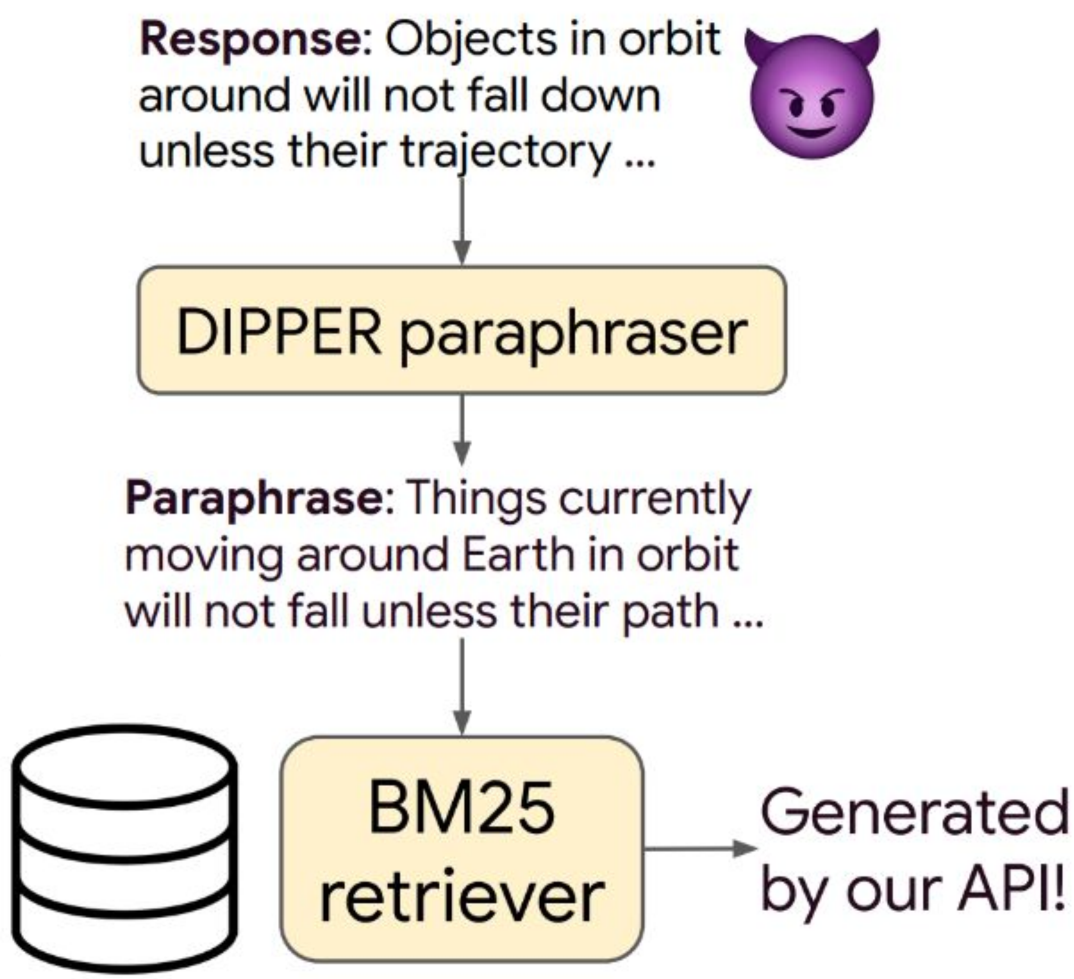
A retrieval-based detector

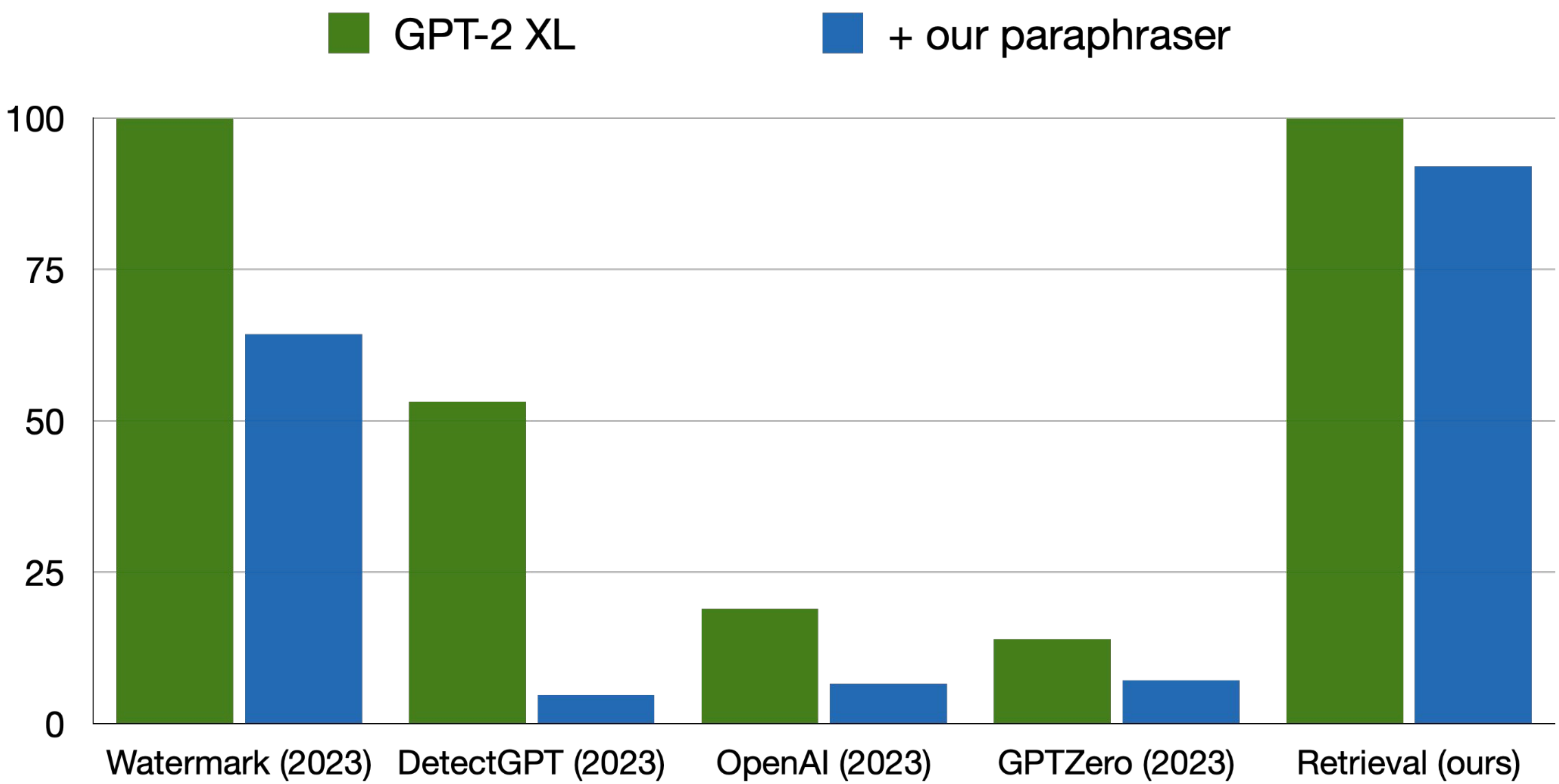


A retrieval-based detector



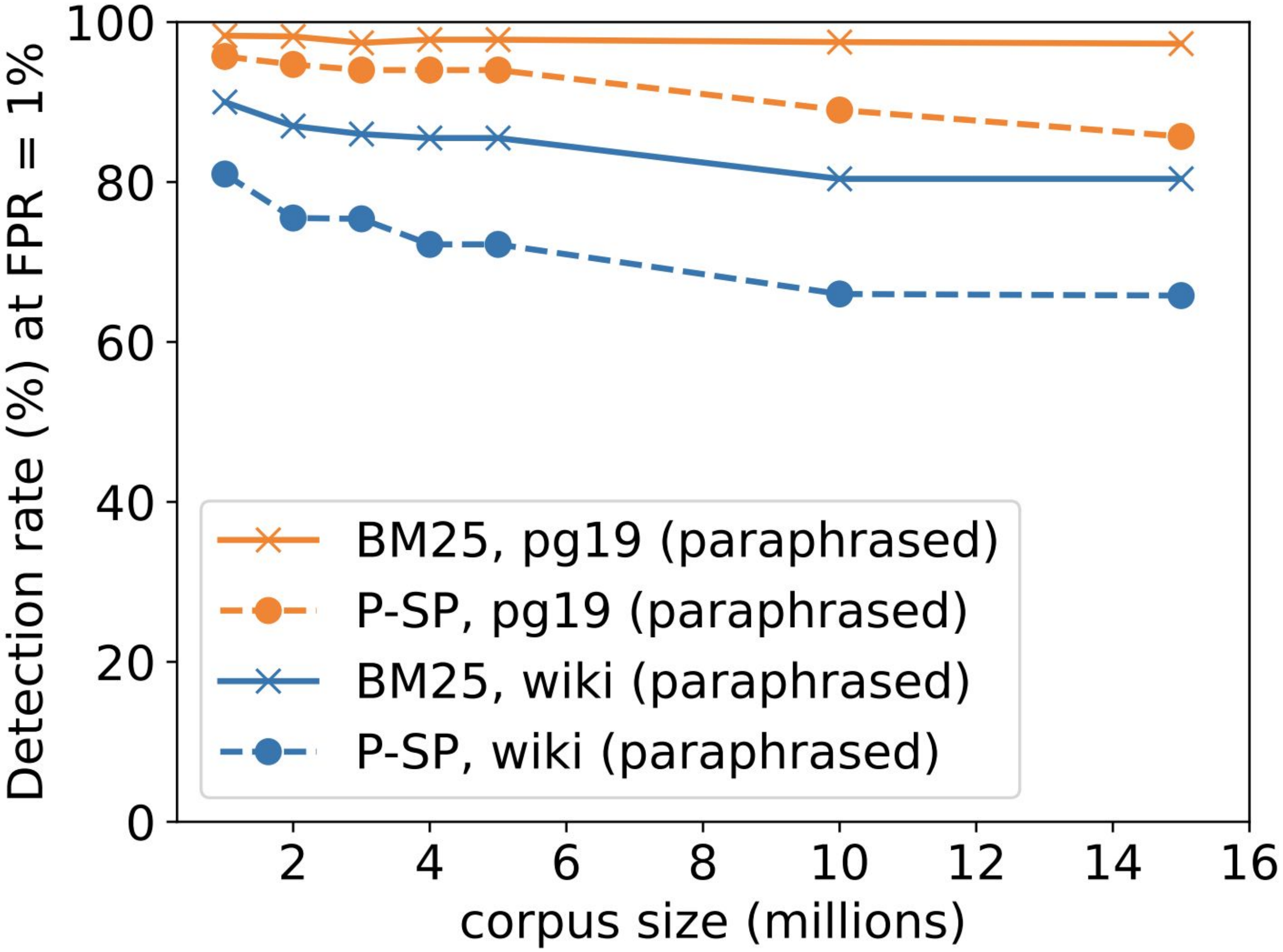
A retrieval-based detector



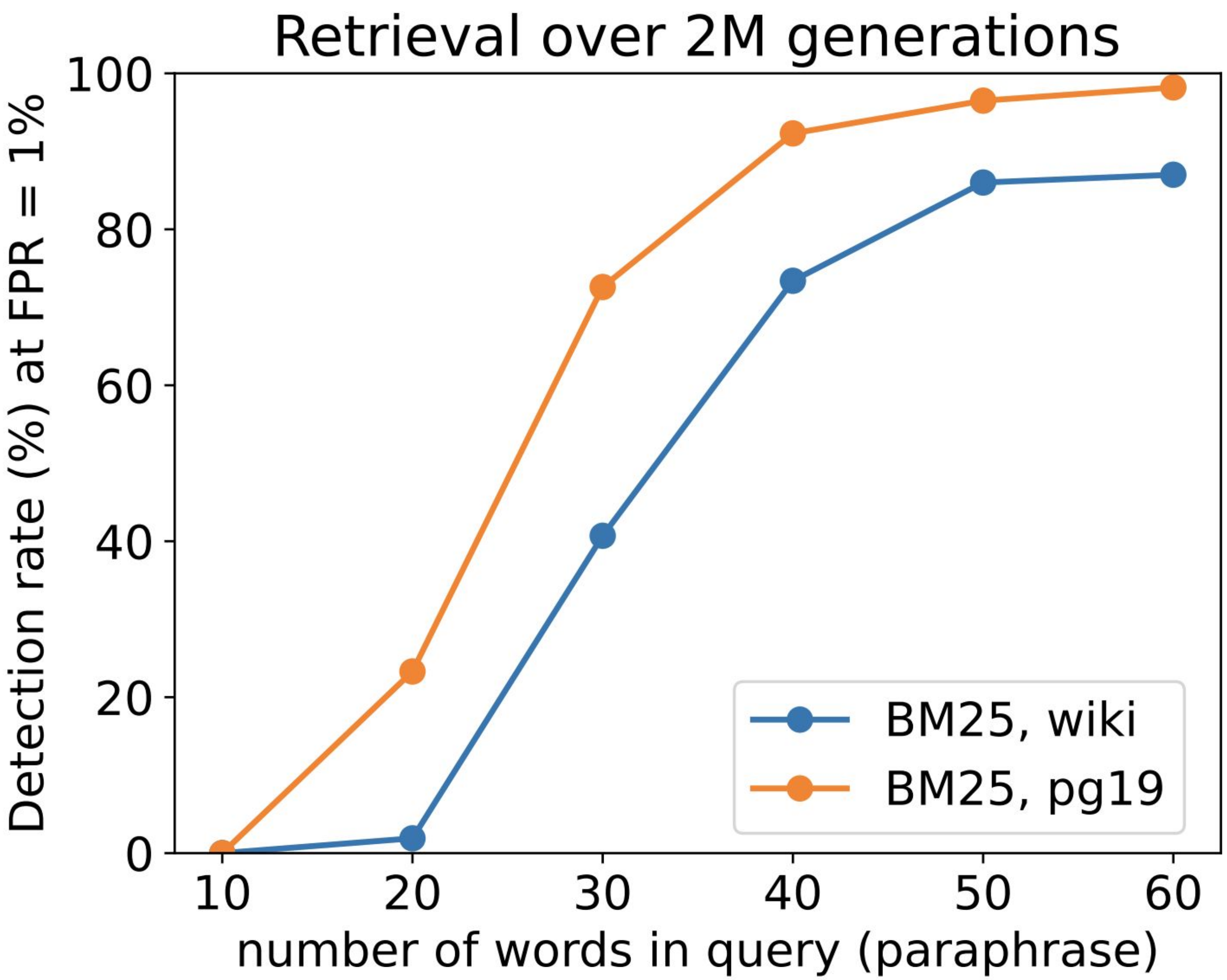


GPT-2 XL
+ our paraphraser
100
75
50
25
0
Watermark (2023)
DetectGPT (2023)
OpenAI (2023)
GPTZero (2023)
Retrieval (ours)

Slightly worse as database size increases



Requires long-form generations



Limitations of retrieval

- Detection is specific only to a single API
- API provider needs to enable low-latency retrieval over a huge-scale database
- False positives due to training data memorization
  - *Possible solution*: retrieving over training data as well
- Vulnerability to membership inference attacks
  - *Possible solution*: redact private info, rate limiting
- If detector is public, attackers can iteratively improve their perturbation model
  - *Possible solution*: give detector access to verified users only (e.g., teachers), rate limiting