Reconstructing Training Data from Document Understanding Models

Jérémie Dentan[1,2], *Arnaud Paran*[1], *Aymen Shabou*[1]

[1]*Crédit Agricole*, [2]*École Polytechnique Paris*

We train document understanding models on sensitive data

We train document understanding models on sensitive data

Document model

≈ BERT + 2D position encoding + visual features

(text) *(layout)* *(image)*

We train document understanding models on sensitive data

1. Name
2. Surname
3. Birth date
4. Document ID

Document model

≈ BERT + 2D position encoding + visual features

(text) *(layout)* *(image)*

Neural Language Models memorize some training data

Extracting Training Data from Large Language Models

Nicholas Carlini[1] Florian Tramèr[2] Eric Wallace[3] Matthew Jagielski[4]
Ariel Herbert-Voss[5,6] Katherine Lee[1] Adam Roberts[1] Tom Brown[5]
Dawn Song[3] Úlfar Erlingsson[7] Alina Oprea[4] Colin Raffel[1]
[1]*Google* [2]*Stanford* [3]*UC Berkeley* [4]*Northeastern University* [5]*OpenAI* [6]*Harvard* [7]*Apple*

Abstract

It has become common to publish large (billion parameter) language models that have been trained on private datasets. This paper demonstrates that in such settings, an adversary can perform a *training data extraction attack* to recover individual training examples by querying the language model.

We demonstrate our attack on GPT-2, a language model trained on scrapes of the public Internet, and are able to extract hundreds of verbatim text sequences from the model's training data. These extracted examples include (public) personally identifiable information (names, phone numbers, and email addresses), IRC conversations, code, and 128-bit UUIDs. Our attack is possible even though each of the above sequences are included in just *one* document in the training data.

We comprehensively evaluate our extraction attack to understand the factors that contribute to its success. Worryingly, we find that larger models are more vulnerable than smaller models. We conclude by drawing lessons and discussing possible safeguards for training large language models.

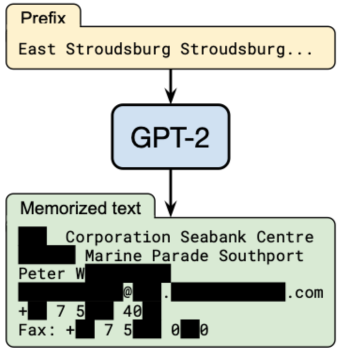


Figure 1: **Our extraction attack.** Given query access to a neural network language model, we extract an individual person's name, email address, phone number, fax number, and physical address. The example in this figure shows information that is all accurate so we redact it to protect privacy.

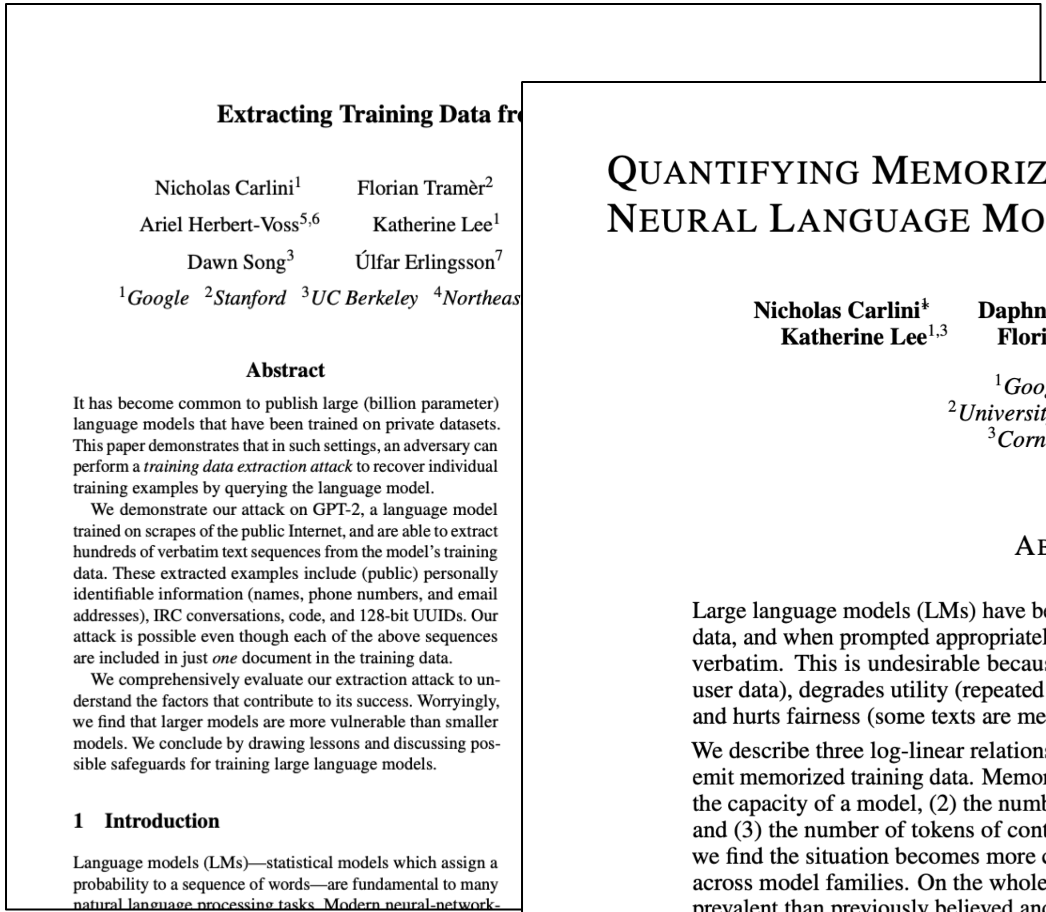
1 Introduction

Language models (LMs)—statistical models which assign a probability to a sequence of words—are fundamental to many natural language processing tasks. Modern neural-network-

Such privacy leakage is typically associated with *overfitting* [75]—when a model's training error is significantly lower

[1] Nicholas Carlini, Florian Tramèr, *et al.* Extracting Training Data From Large Language Models. Usenix Security. 2021.

Neural Language Models memorize ~~*some training data*~~ > 1% of training data



Extracting Training Data fr

Nicholas Carlini[1] Florian Tramèr[2]
Ariel Herbert-Voss[5,6] Katherine Lee[1]
Dawn Song[3] Úlfar Erlingsson[7]
[1]*Google* [2]*Stanford* [3]*UC Berkeley* [4]*Northeas*

Abstract

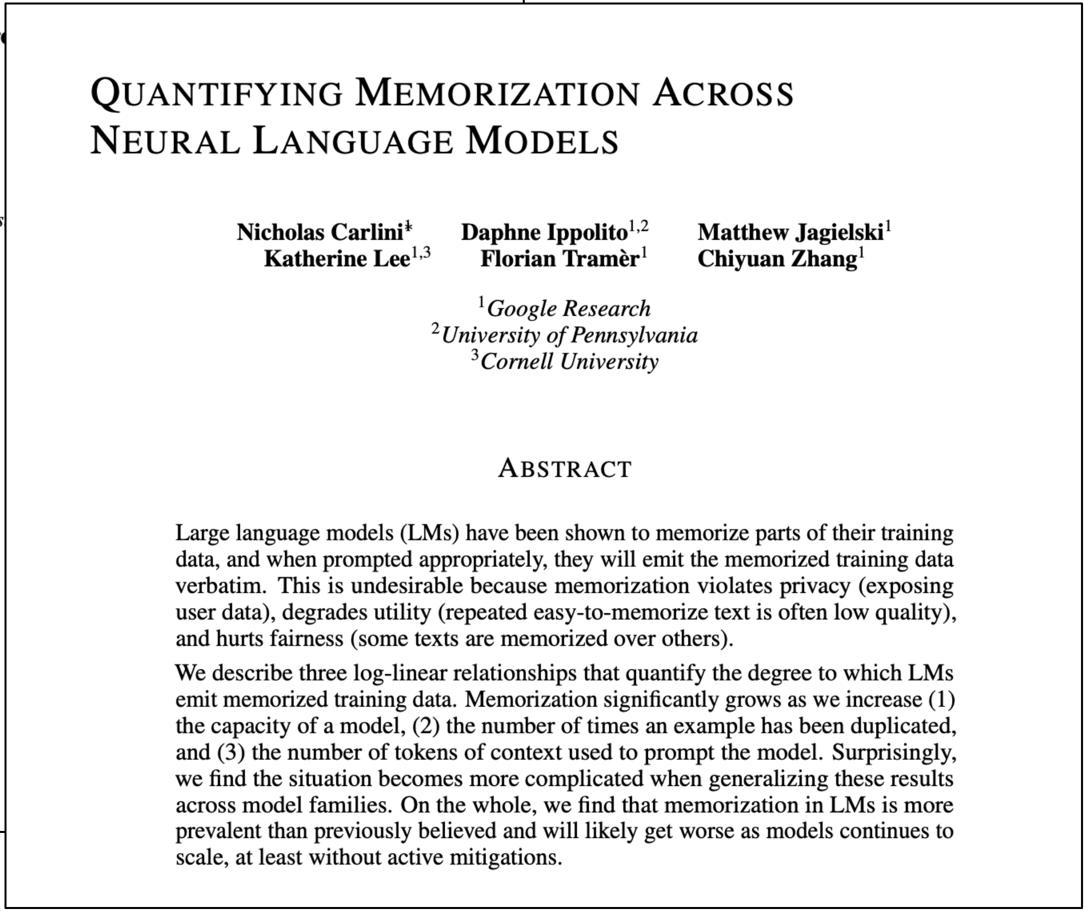
It has become common to publish large (billion parameter) language models that have been trained on private datasets. This paper demonstrates that in such settings, an adversary can perform a *training data extraction attack* to recover individual training examples by querying the language model.

We demonstrate our attack on GPT-2, a language model trained on scrapes of the public Internet, and are able to extract hundreds of verbatim text sequences from the model's training data. These extracted examples include (public) personally identifiable information (names, phone numbers, and email addresses), IRC conversations, code, and 128-bit UUIDs. Our attack is possible even though each of the above sequences are included in just *one* document in the training data.

We comprehensively evaluate our extraction attack to understand the factors that contribute to its success. Worryingly, we find that larger models are more vulnerable than smaller models. We conclude by drawing lessons and discussing possible safeguards for training large language models.

1 Introduction

Language models (LMs)—statistical models which assign a probability to a sequence of words—are fundamental to many natural language processing tasks. Modern neural-network-



QUANTIFYING MEMORIZATION ACROSS NEURAL LANGUAGE MODELS

Nicholas Carlini[1] **Daphne Ippolito**[1,2] **Matthew Jagielski**[1]
Katherine Lee[1,3] **Florian Tramèr**[1] **Chiyuan Zhang**[1]
[1]*Google Research*
[2]*University of Pennsylvania*
[3]*Cornell University*

ABSTRACT

Large language models (LMs) have been shown to memorize parts of their training data, and when prompted appropriately, they will emit the memorized training data verbatim. This is undesirable because memorization violates privacy (exposing user data), degrades utility (repeated easy-to-memorize text is often low quality), and hurts fairness (some texts are memorized over others).

We describe three log-linear relationships that quantify the degree to which LMs emit memorized training data. Memorization significantly grows as we increase (1) the capacity of a model, (2) the number of times an example has been duplicated, and (3) the number of tokens of context used to prompt the model. Surprisingly, we find the situation becomes more complicated when generalizing these results across model families. On the whole, we find that memorization in LMs is more prevalent than previously believed and will likely get worse as models continues to scale, at least without active mitigations.

[1] Nicholas Carlini, Florian Tramèr, *et al.* Extracting Training Data From Large Language Models. Usenix Security. 2021.
[2] Nicholas Carlini, Daphne Ippolito, *et al.* Quantifying Memorization Across Neural Language Models. ICLR. 2023.

Are document models prone to privacy attacks ?

Are document models prone to privacy attacks ?

1. **No privacy attacks against document models were documented in the literature**

Are document models prone to privacy attacks ?

1. **No privacy attacks against document models were documented in the literature**

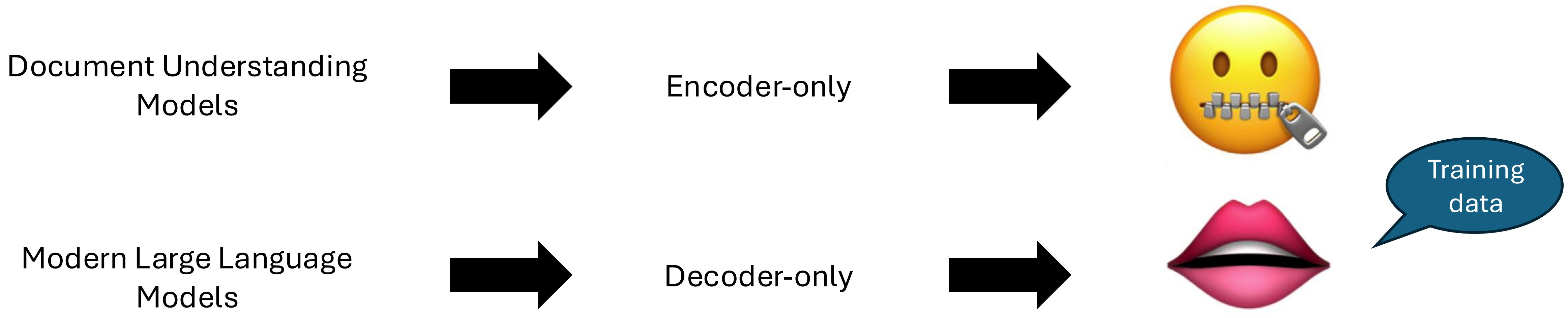
2. **Document models are multimodal. Does it increase / decrease robustness ?**

Are document models prone to privacy attacks ?

1. **No privacy attacks against document models were documented in the literature**

2. **Document models are multimodal. Does it increase / decrease robustness ?**

3. **It’s harder to do reconstruction attacks on encoder-only models**



Are document models prone to privacy attacks ?

Short answer: YES

How do we reconstruct training data ?

Target field in a
target document

How do we reconstruct training data ?

Auxiliary MLM
(Trained on public data)

Target field in a target document

- 14
- 31
- 19
- The
- 01
- 13
- Yes

Candidate tokens

How do we reconstruct training data ?

Auxiliary MLM
(Trained on public data)

Target model
(Trained on the private data)

Target field in a target document

- 14
- 31
- 19
- The
- 01
- 13
- Yes

Candidate tokens

- 14
- 31
- 19
- The
- 01
- <u>**13**</u>
- Yes

Token that minimizes the loss

How do we reconstruct training data ?

Target field in a target document

Auxiliary MLM
(Trained on public data)

- 14
- 31
- 19
- The
- 01
- 13
- Yes

Candidate tokens

Target model
(Trained on the private data)

- 14
- 31
- 19
- The
- 01
- **13**
- Yes

Token that minimizes the loss

How do we reconstruct training data ?

Target field in a
target document

Etc.

How do we reconstruct training data ?

Target field in a
target document

How do we reconstruct training data ?

Target field in a
target document

In practice, it required many heuristics
to work on real data…

Did it work in practice ?

Data	Archi	Task	Reconstruction	Len	Occ
			restaurant jiawei jiawei house		
			guardian health and beauty sdn bhd		
			m. a. peterson		
			r. g. ryan		
			lim seng tho hardware trading		
			101. 75		
			april 13, 1984		
			1. 500. 00		
			dr. a. w. spears		

In the best setting, we perfectly reconstructed
4.1% of the fields

Did it work in practice ?

2 datasets: FUNSD & SROIE

2 architectures: LayoutLM v1 & BROS

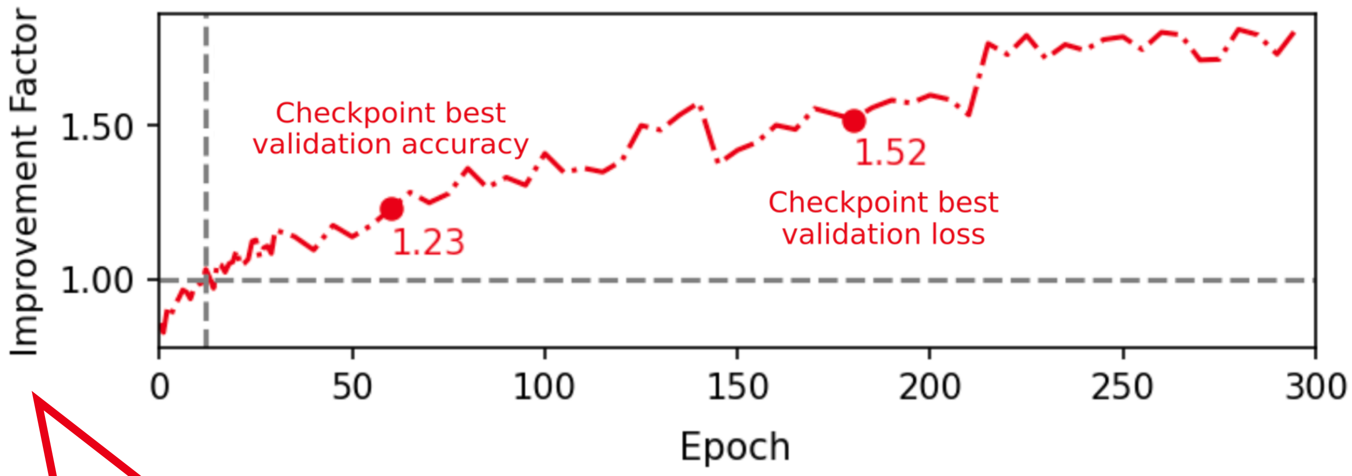
Four fine-tuning tasks: MLM, EL, EE-SPD, EE-BIO

Data	Archi	Task	Reconstruction	Len	Occ
SRO	LayoutLM	EE-SPD	restaurant jiawei jiawei house	6	1
SRO	LayoutLM	EE-BIO	guardian health and beauty sdn bhd	8	1
FUN	LayoutLM	EL	m. a. peterson	5	1
FUN	LayoutLM	EE-SPD	r. g. ryan	5	1
SRO	LayoutLM	MLM	lim seng tho hardware trading	6	1
SRO	LayoutLM	MLM	101. 75	3	2
FUN	LayoutLM	MLM	april 13, 1984	4	1
FUN	BROS	MLM	1. 500. 00	6	1
FUN	BROS	MLM	dr. a. w. spears	7	1

In the best setting, we perfectly reconstructed **4.1% of the fields**

Ablation #1 : Does memorization require overfitting ?

Ablation #1 : Does memorization require overfitting ?



Improvement Factor = performance of the attack

No, it does not.

- Memorization starts well before overfitting
- Overfitting increases memorization
- Consistent with other works such as [3]

[3] Chiyuan Zhang, Samy Bengio, *et al.* Understanding deep learning requires rethinking generalization. ICLR. 2017.

Ablation #2 : Does the visual modality memorize data ?

Document model

≈ BERT + 2D position encoding + **visual features**

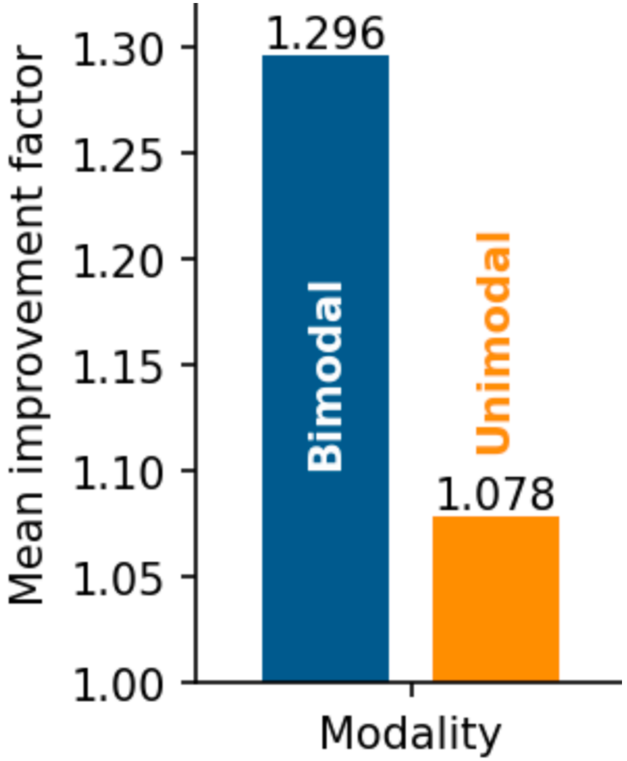
(text) *(layout)* *(image)*

Ablation #2 : Does the visual modality memorize data ?

Document model

≈ <u>BERT + 2D position encoding</u> + <u>visual features</u>

(text) *(layout)* *(image)*



Yes, it does.

➢ Pixel/token associations are memorized.

Ablation #3 : Does the layout memorize data ?

Document model

≈ BERT + **2D position encoding**

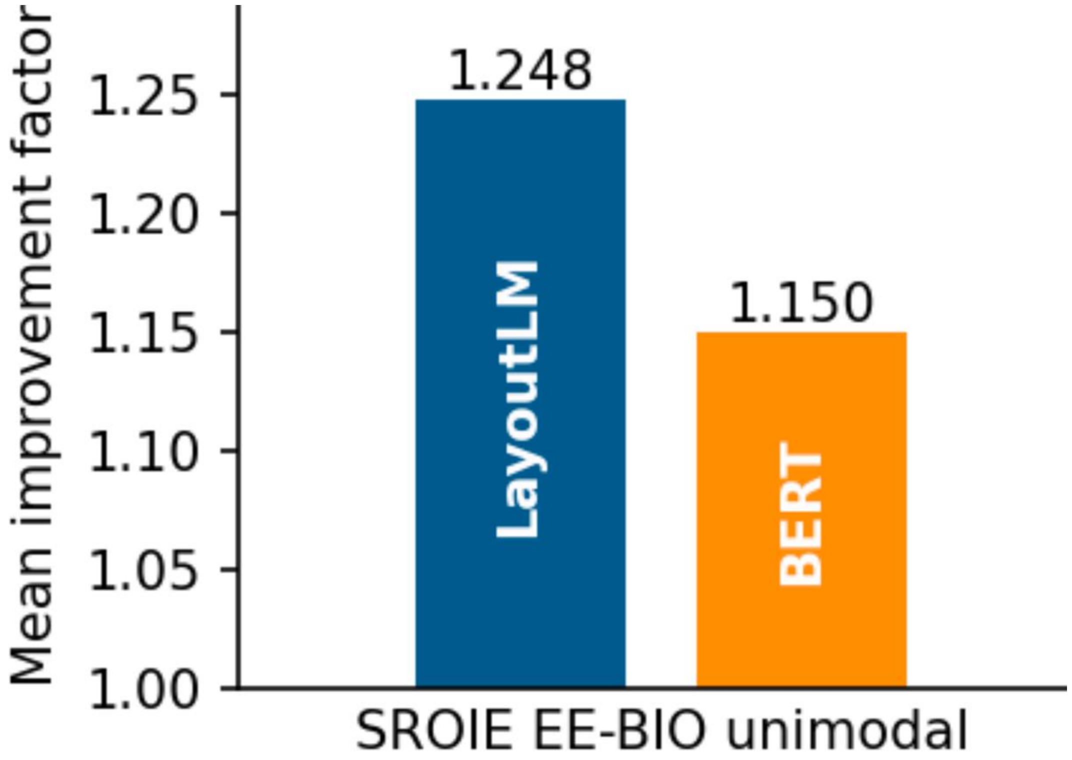
(text) *(layout)*

Ablation #3 : Does the layout memorize data ?

Document model

≈ **BERT** + **2D position encoding**

(text) *(layout)*



Yes, it does.

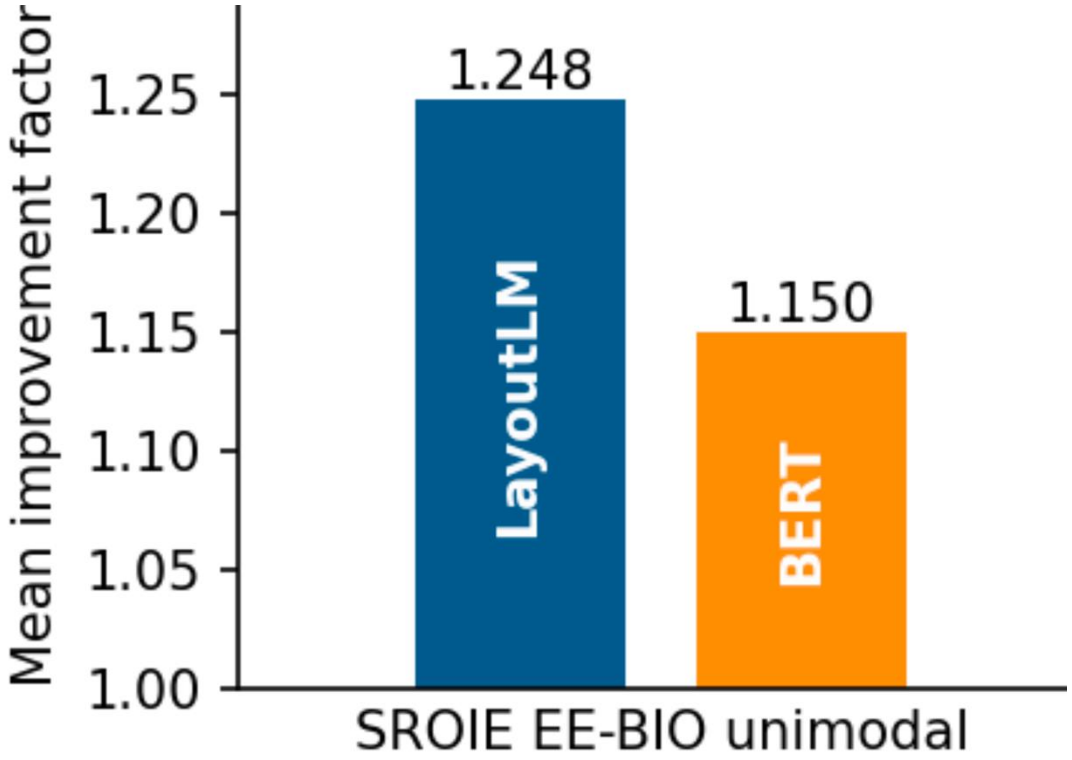
➢ Layout/token associations are memorized.

Ablation #3 : Does the layout memorize data ?

Document model

≈ **BERT** + **2D position encoding**

(text) *(layout)*



Yes, it does.

- Layout/token associations are memorized.

And other ablations in the paper…

Conclusions

1. Document understanding models memorize training data

2. Reconstruction attacks are realistic, even without overfitting / duplication

3. Document models are more vulnerable than pure-text models for the same task

Future research directions

Improvements :

- Implement the same attack strategy with pre-training rather than fine-tuning

On a broader scale :

- Deepen our understanding of the nature of memorization and why it happens

Questions ?