RedactedGPT: Final Report

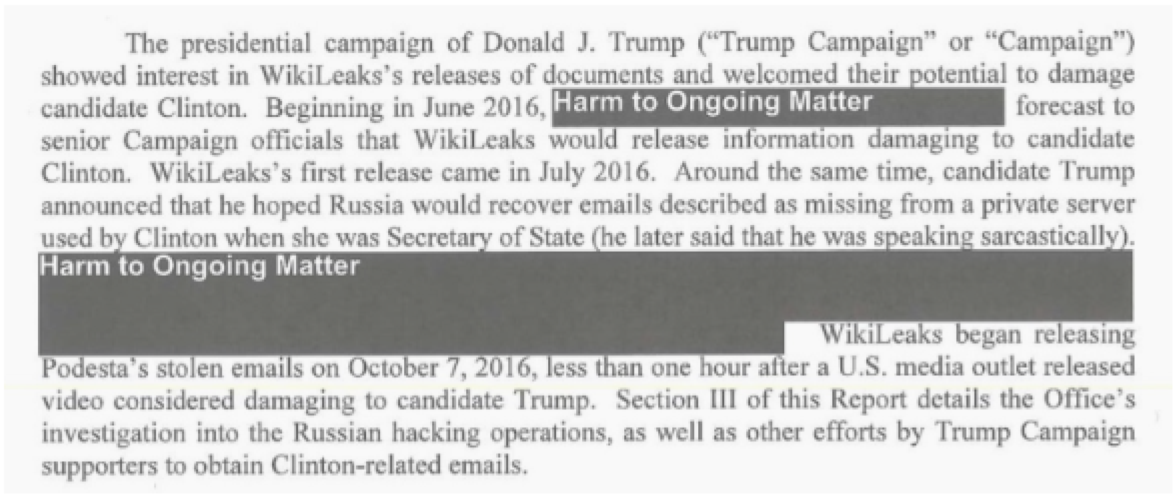
Aiden Rosebush 1005334997

Esmat Sahak 1004928126

Word Count: 1998, Penalty: 0

1. Introduction [1]

Redaction is the removal of sensitive information from private records (Figure 1). It has broad applications, including court cases [1], confidential emails [2], and medical records [3]. With the development of large language models (LLMs), there is increasing pressure to prevent these tools from revealing redacted text.



The presidential campaign of Donald J. Trump ("Trump Campaign" or "Campaign") showed interest in WikiLeaks's releases of documents and welcomed their potential to damage candidate Clinton. Beginning in June 2016, Harm to Ongoing Matter forecast to senior Campaign officials that WikiLeaks would release information damaging to candidate Clinton. WikiLeaks's first release came in July 2016. Around the same time, candidate Trump announced that he hoped Russia would recover emails described as missing from a private server used by Clinton when she was Secretary of State (he later said that he was speaking sarcastically). Harm to Ongoing Matter WikiLeaks began releasing Podesta's stolen emails on October 7, 2016, less than one hour after a U.S. media outlet released video considered damaging to candidate Trump. Section III of this Report details the Office's investigation into the Russian hacking operations, as well as other efforts by Trump Campaign supporters to obtain Clinton-related emails.

Figure 1: *Sample redacted excerpt from the Mueller Report [4]*

In this project, we intend to test the ability of transformer models to predict redacted text. We focus on declassified United States (US) government documents focusing on the 9/11 attacks. If we can train a language model to accurately predict censored information, we demonstrate proof of concept for a method of reversing redaction. Instead of directly working with redacted documents, we simulate this effect by masking tokens (section 3). Our model will be a toy example of one which, in the context of US federal elections, could have significant implications by predicting redacted information related to the modern candidates.

2. Illustration

Figure 2 shows the models we train to predict masked tokens, along with sample masked input and top 10 candidates to replace the masked token and their likelihoods.

3. Background and Related Work [1]

In most cases, models are trained to improve redaction, which entails (a) identifying text that should be redacted and (b) checking whether models can predict masked text. We only intend to work on (b).

[1] Modified version from project proposal

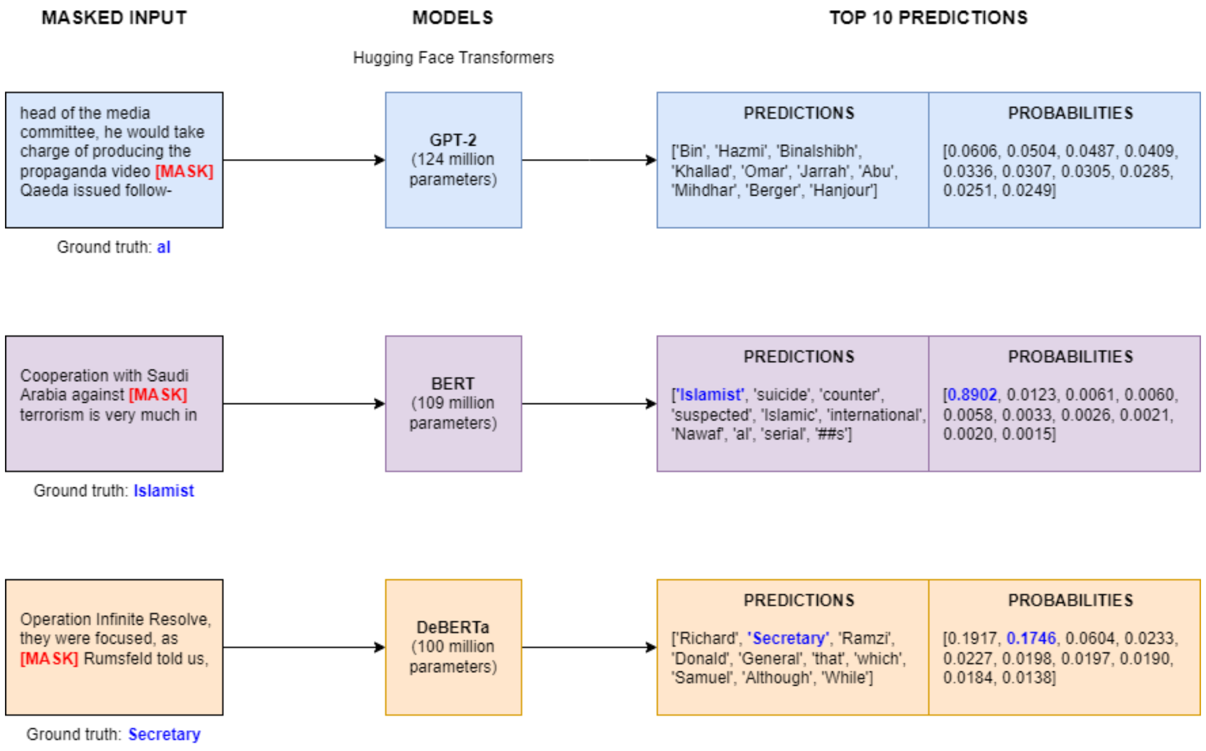

MASKED INPUT
MODELS
Hugging Face Transformers
TOP 10 PREDICTIONS
head of the media committee, he would take charge of producing the propaganda video [MASK] Qaeda issued follow-
Ground truth: al
GPT-2 (124 million parameters)
PREDICTIONS
['Bin', 'Hazmi', 'Binalshibh', 'Khallad', 'Omar', 'Jarrah', 'Abu', 'Mihdhar', 'Berger', 'Hanjour']
PROBABILITIES
[0.0606, 0.0504, 0.0487, 0.0409, 0.0336, 0.0307, 0.0305, 0.0285, 0.0251, 0.0249]
Cooperation with Saudi Arabia against [MASK] terrorism is very much in
Ground truth: Islamist
BERT (109 million parameters)
PREDICTIONS
['Islamist', 'suicide', 'counter', 'suspected', 'Islamic', 'international', 'Nawaf', 'al', 'serial', '##s']
PROBABILITIES
[0.8902, 0.0123, 0.0061, 0.0060, 0.0058, 0.0033, 0.0026, 0.0021, 0.0020, 0.0015]
Operation Infinite Resolve, they were focused, as [MASK] Rumsfeld told us,
Ground truth: Secretary
DeBERTa (100 million parameters)
PREDICTIONS
['Richard', 'Secretary', 'Ramzi', 'Donald', 'General', 'that', 'which', 'Samuel', 'Although', 'While']
PROBABILITIES
[0.1917, 0.1746, 0.0604, 0.0233, 0.0227, 0.0198, 0.0197, 0.0190, 0.0184, 0.0138]

Figure 2: *Overview of models and sample input-output*

A paper published in the Massachusetts Institute of Technology's Computational Linguistics journal curated the Text Anonymization Benchmark (TAB), an open-source corpus for text anonymization based on court proceedings [1]. They propose metrics for assessing the quality of redaction (i.e. whether it masks the correct information), and compare the performance of a Longformer model fine-tuned on TAB to several other baseline models on out-of-domain data, finding that it performs better.

The ability of transformer models to predict masked tokens raises privacy concerns for documents containing redacted text. Researchers from Trinity College Dublin used BART to generate the top 5 most likely representations of redacted text [5]. They then propose a privacy evaluation metric by computing the similarity of generated outputs - the higher the similarity, the lower the privacy retention score.

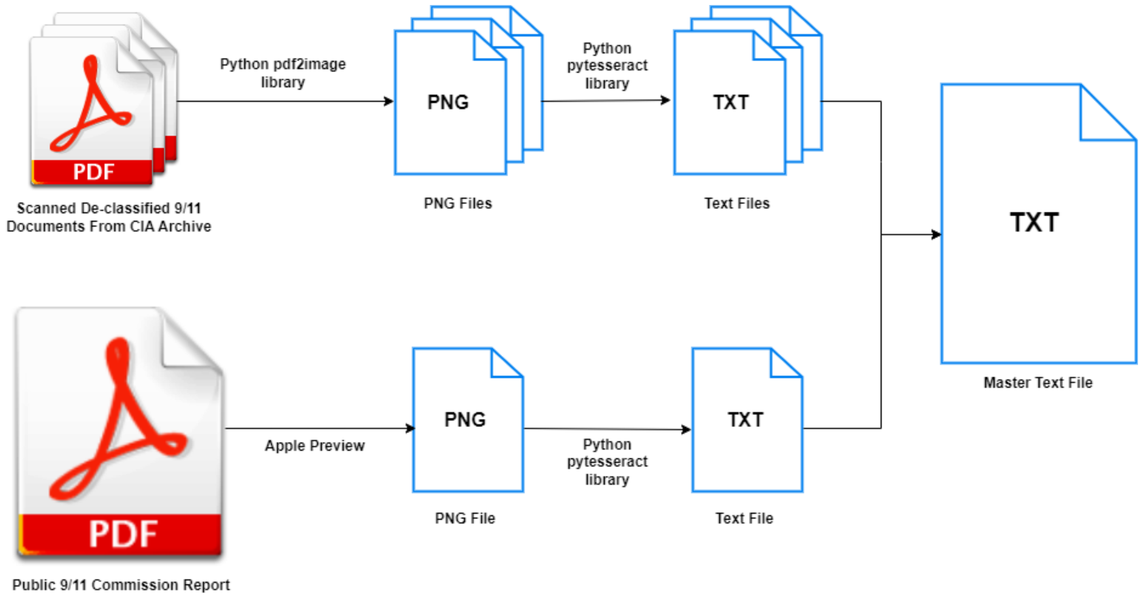
3. Data and Data Processing

The Central Intelligence Agency (CIA) Reading Room website [6] provides access to scanned government documents as PDF files. We retrieved six declassified documents related to the 9/11 attacks from this site along with the PDF of the 9/11 commission report [7].

To extract text from the scanned PDFs, we convert them to image format using Python's pdf2image library [8]. We also convert the commission report PDF to a PNG file using Apple

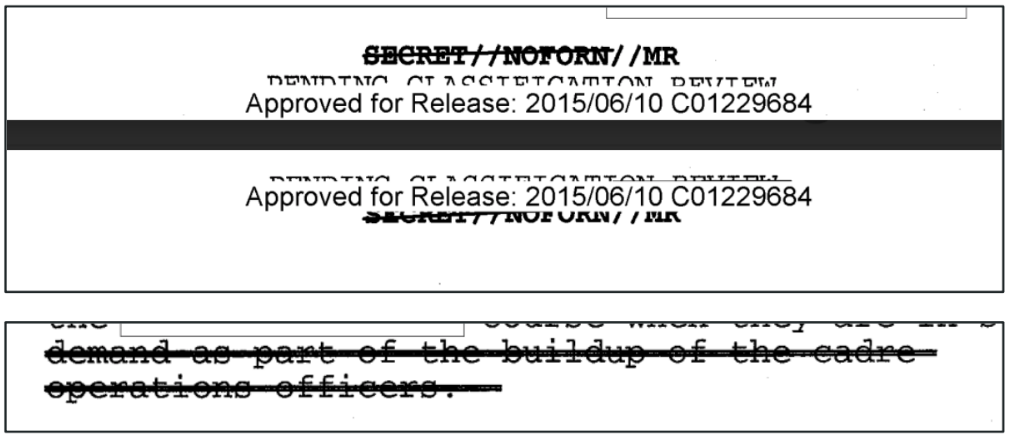
Preview. We then use Python's pytesseract library [9] to extract text contained within the images, save results to text files, and consolidate the text files into a master text file. Figure 3 summarizes the data processing pipeline. Due to some anomalies in the scanned PDFs (Figure 4) and pytesseract misinterpreting some words, manual cleaning was necessary to ensure the cleaned master text file contained coherent words and sentences.

Next, we use a Python dictionary to rank the most frequent words and identify 72 domain-specific keywords relevant for training a model. They include names such as Bin Laden, acronyms like CIA and FBI, or organizations like Al Qaeda. These keywords were chosen arbitrarily based on prior knowledge of the subject matter. We acknowledge that by studying the documents in more detail, better keywords could have been chosen. Nonetheless, we believe these keywords are adequate for a proof of concept.



Python pdf2image library
Python pytesseract library
PNG
TXT
TXT
Scanned De-classified 9/11 Documents From CIA Archive
PNG Files
Text Files
Master Text File
Apple Preview
PNG
Python pytesseract library
TXT
PNG File
Text File
Public 9/11 Commission Report

Figure 3: *Data processing pipeline*



SECRET//NOFORN//MR

Approved for Release: 2015/06/10 C01229684

Approved for Release: 2015/06/10 C01229684

~~demand as part of the buildup of the cadre operations officers.~~

Figure 4: *Anomalies in scanned PDF documents*

To generate training examples, we take the keywords and one word before and after as labels, and the surrounding line of text as context. We only set a target if it is a keyword or is already in the model tokenizer's vocabulary (Figure 5). We add every keyword to each model's tokenizer and make a custom training dataset for each model. We originally attempted to use four lines of context and five target word candidates on each side of the keyword, adding all possible target words to each model tokenizer's vocabulary, but this proved infeasible with the computational resources available on Colab Pro (section 8).

Example: For a history of the DCI's authority over the intelligence community, **see** **CIA** **report**

- For a history of the DCI's authority over the intelligence community, **[MASK]** **CIA** **report**
    - ✅ **see** is part of the model tokenizer's vocabulary
- For a history of the DCI's authority over the intelligence community, **see** **[MASK]** **report**
    - ✅ **CIA** is part of the model tokenizer's vocabulary
- For a history of the DCI's authority over the intelligence community, **see** **CIA** **[MASK]**
    - ❌ **report** is not part of the model tokenizer's vocabulary

Figure 5: *Example of training sample generation*

We tokenize the datasets, replacing the target word with a [MASK] token. Without knowing the text in the redacted segments, we couldn't consider any redacted sentences for training. Due to our limited set of target words and context in training, we couldn't generate plausible predictions for redacted segments after training. Our training-validation-test splits are shown in Table 1.

Model	Mask Labels	Dataset Examples	Train Examples	Validation Examples	Test Examples
gpt2 (124M params)	404	15947 (39.4 examples per label)	11162 (70%)	1674 (10.5%)	3111 (19.5%)
bert-base-cased (109M params)	590	16343 (27.7 examples per label)	11440 (70%)	2451 (15%)	2452 (15%)
deberta-base (100M params)	404	15947 (39.4 examples per label)	11162 (70%)	1914 (12%)	2871 (18%)

Table 1: *Training, validation, and test splits, splits vary due to Colab RAM limits (section 8)*

4. Architecture and Software

This is a Class 1 project, meaning we fine-tune pretrained models. We also train embeddings for the new tokens representing keywords identified in section 3. We use: (1) GPT-2: autoregressive transformer using only words preceding the target as context (2) BERT: bidirectional transformer which uses words before and after the target as context (3) DeBERTa: a smaller version of BERT which separates position and word embeddings. These pre-trained transformer models are available on Hugging Face [10], under the names *gpt2*, *bert-base-cased*, and *microsoft/deberta-base*.

To measure accuracy, we use top-k accuracy with k = 10, meaning if the model predicts that the label is among the ten most likely candidates, we consider it to have made a correct prediction. With smaller values of k, accuracy decreases, but we want the model to generate several possible answers to simulate prediction using decoding methods such as beam search and sampling.

5. Baseline Model or Comparison

First, we want to see if BERT and DeBERTa outperforms GPT-2, as they are bidirectional so they have access to the entire line of context, whereas GPT-2 only has access to words preceding the target. Second, we want to compare model sizes, namely between BERT and DeBERTa. We want to see which model has a higher accuracy, and if BERT displays more overfitting due to being a larger model. Finally, we want to see if DeBERTa's separate positional and word embeddings outperforms BERT.

6. Quantitative Results

We explore batch size and learning rate as hyperparameters to optimize training. Due to long runtimes and previous issues with larger training sets, we explore only four hyperparameter combinations for each model. The best test accuracy was achieved by BERT (93%), while DeBERTa was close (91.7%) and GPT-2 lagged behind (62.2%). Tables 2a, 2b, and 2c summarize the test accuracy of all experiments

gpt2		
	BS: 32	BS: 64
LR: 5e-5	35.42%	37.61%
LR: 5e-4	**62.20%**	52.04%

(a)

deberta-base		
	BS: 32	BS: 64
LR: 5e-5	89.72%	87.01%
LR: 5e-4	48.66%	**91.68%**

(b)

bert-base-cased		
	BS: 50	BS: 100
LR: 5e-6	71.73%	70.00%
LR: 5e-5	**92.99%**	90.62%

(c)

Tables 2a, 2b, 2c: *Experiments - LR denotes learning rate and BS denotes training batch size.*

We found that only BERT, despite its accuracy, exhibited some overfitting, as the validation loss lags behind the training loss. This suggests that BERT might perform even better with different hyperparameters, or on a larger dataset. This supports our hypothesis that BERT can overfit when DeBERTa doesn't, given it is a larger model. The accuracy and loss curves of the best of each model are shown below in Figures 6a, 6b, and 6c.

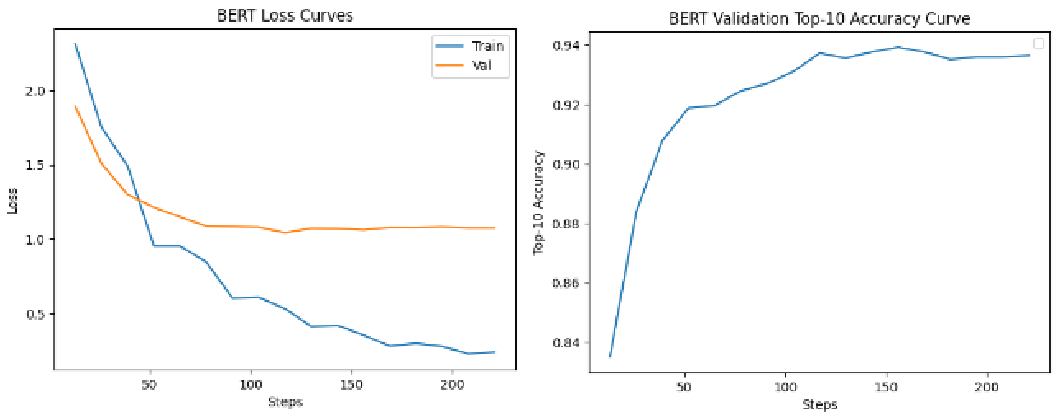


Figure 6a: *Loss and accuracy curves of the BERT model*

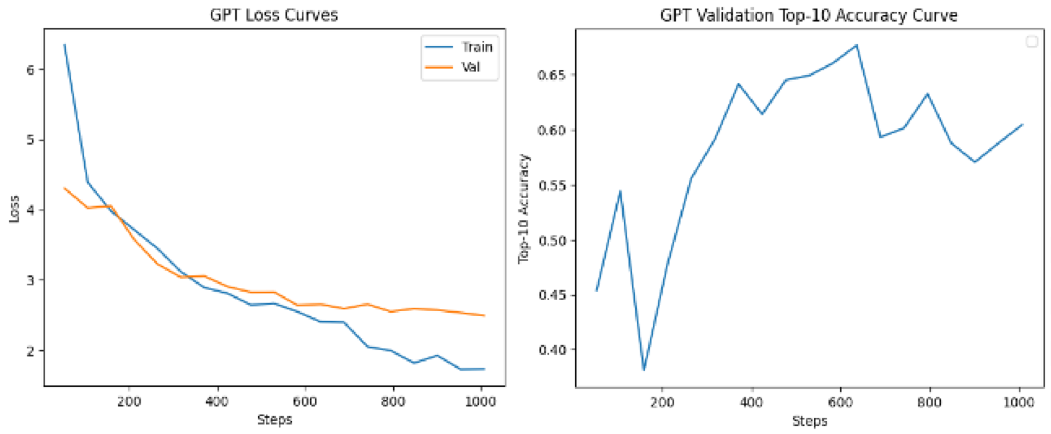


Figure 6b: *Loss and accuracy curves of the GPT-2 model*

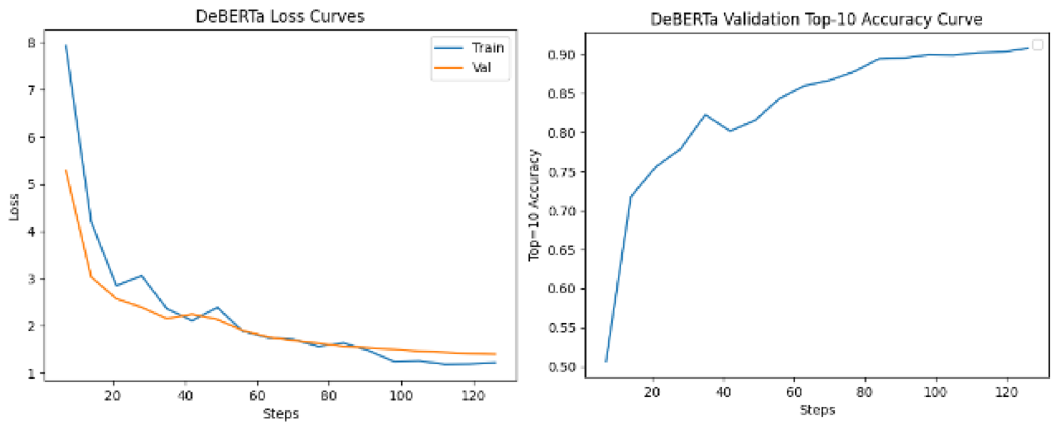


Figure 6c: *Loss and accuracy curves of the DeBERTa model*

7. Qualitative Results

We found many examples supporting our hypothesis that GPT-2 performs worse than BERT due to BERT's bidirectionality. For example, on a prompt which reads *"head of the media committee, he would take charge of producing the propaganda video [MASK] Qaeda issued follow-"*, GPT-2 can't guess the masked label, since it can't learn that *"al"* frequently precedes *"Qaeda"* but BERT can. BERT also correctly identifies the mask in the input *"to [MASK] Moussaoui from obtaining any further training that he could use to"* - here GPT-2 only has one word of context so it fails to guess correctly, but BERT uses the entire line. DeBERTa performed similarly to BERT with such examples.

We found that BERT outperforms DeBERTa in cases of frequently used words which were already in the tokenizer's vocabulary. For example, DeBERTa could not identify *"among"* as the label for the input *"Jan. 31, 1997; Intelligence report, Cooperation [MASK] Usama Bin Laden's Islamic Army, Iran, and the NIF Jan. 31"*, but BERT can. This performance difference could be due to the different sizes of the models, or is a limitation of decoupling the positional and contextual embeddings in DeBERTa. Finally, all models performed poorly on infrequently used or new tokens, which is to be expected when embeddings starting in a randomized state are not sufficiently trained.

8. Discussion and Learnings

Balancing the scope of the training data with available computational resources was challenging. We had to scale back the number of labels, the number of new embeddings added to each tokenizer, and the context for each training example (section 3). The GPU RAM would overflow during evaluation, even with a small validation set. This was because the entire validation set is passed to the evaluation function in Hugging Face's Trainer class [10]. A large validation batch size forces the GPU to manage each example's interaction with each model parameter, and the resulting probability tensors, spanning the entire vocabulary, are kept in GPU RAM for the evaluation process.

To mitigate these issues, we scaled down the dataset, reduced the batch size to prevent overflow while still allowing fast evaluation, and added a parameter in the training arguments which passes the validation results to CPU memory after every evaluation step. We also added a custom callback for each trainer which set the model to evaluation mode so gradients don't accumulate during evaluation. Although Hugging Face implements this in their source code [10] for their evaluation function, it nonetheless allowed a tenfold increase in validation sizes. In the future, we will start with a conservative scope and gently scale up, instead of being too ambitious then scaling down.

We also learned how difficult data processing can be when working with scanned PDFs. In the future, we will carefully clean all document images, automate image processing further and run tests on known documents to improve the pipeline. Despite these issues, we demonstrated that bidirectional LLMs can bypass redaction provided sufficient context.

9. Individual Contributions

Contributions to the project were equal. Aiden found, processed, and cleaned the raw data, then generated custom training sets for each model, while Esmat generated initial results for the GPT and BERT models. Aiden found how to mitigate RAM overflow and trained the BERT model to completion. Esmat adapted these methods for DeBERTa and GPT-2 and collected all results. Many of the choices about scaling of the project was a collaborative effort.

10. References

[1] I. Pilán, P. Lison, L. Øvrelid, A. Papadopoulou, D. Sánchez, and M. Batet, "The Text Anonymization Benchmark (TAB): A dedicated corpus and evaluation framework for text anonymization," *Comput. Linguist. Assoc. Comput. Linguist.*, vol. 48, no. 4, pp. 1053–1101, 2022.

[2] E. Eder, U. Krieg-Holz, and U. Hahn, "CodE Alltag 2.0 --- A Pseudonymized German-Language Email Corpus," in *Proceedings of the Twelfth Language Resources and Evaluation Conference*, 2020, pp. 4466–4477.

[3] M. Friedrich, A. Köhn, G. Wiedemann, and C. Biemann, "Adversarial learning of privacy-preserving text representations for DE-identification of medical records," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 5829–5839.

[4] "Read the Mueller report: The full redacted version, annotated," *Washington post (Washington, D.C.: 1974)*, The Washington Post.

[5] V. Gusain and D. Leith, "Towards quantifying the privacy of redacted text," in *Lecture Notes in Computer Science*, Cham: Springer Nature Switzerland, 2023, pp. 423–429.

[6] "Freedom of information act electronic reading room," *Cia.gov*. [Online]. Available: https://www.cia.gov/readingroom/.

[7] The 9/11 Commission Report, https://www.9-11commission.gov/report/911Report.pdf

[8] "Pdf2image," *PyPI*. [Online]. Available: https://pypi.org/project/pdf2image/.

[9] "Pytesseract," *PyPI*. [Online]. Available: https://pypi.org/project/pytesseract/.

[10] *Transformers: 🤗 transformers: State-of-the-art Machine Learning for Pytorch, TensorFlow, and JAX.*