Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

PhD Thesis

Trinity College Dublin

The University of Dublin

School of Computer Science and Statistics

Text Privacy in the age of Large Language Models

Author:
Vaibhav Gusain

Supervisor:
Professor Douglas Leith

Submitted
February 2026

A thesis submitted in fulfillment of the requirements for the degree of
Doctor of Philosophy

Declaration

I, the undersigned, declare that this work has not previously been submitted as an exercise for a degree at this, or any other University, and that unless otherwise stated, is my own work.

I, the undersigned, agree to deposit this thesis in the University's open access institutional repository or allow the library to do so on my behalf, subject to Irish Copyright Legislation and Trinity College Library conditions of use and acknowledgement.

I consent to the examiner retaining a copy of the thesis beyond the examining period, should they so wish (EU GDPR May 2018).

Signed:  Date: 5 February 2026

Acknowledgement

I would like to express my deepest gratitude to my advisor, Prof. Douglas Leith, for his unwavering support throughout my PhD journey. His guidance not only helped me grow as a researcher but also shaped me into a more grounded and relaxed individual.

I am also especially thankful to my friends, colleagues, and table-tennis rivals—David, Dilina, Mohamed, and Dr. Sulthana. From endless banter to our ping-pong matches, they made every day in the lab brighter. Thank you for introducing me to Irish slang (even if I still sound like a confused tourist at times) and for filling this journey with laughter, energy, and friendship. You turned the challenges of a PhD into experiences I will always fondly remember. "Lab Rats" will always hold a special place in my heart.

Beyond the lab, I had the privilege of meeting some wonderful people in Ireland who remain very close to my heart. They stood by me during difficult times, and I owe special thanks to Salil, Vishal, Takshil, Neel, Sooraj, Prassana, Sachin, and Shivani for brightening my days and making life more bearable amid Dublin's often gloomy weather. I am also deeply grateful to Shyamalima and Sreeju, whose guidance and support helped me navigate life in Ireland when I was new to the country.

I would also like to thank my family. To my mother for not only bearing me but also bearing me, my brother for temporarily being the responsible sibling and his constant supply of computer games. I am infinitely indebted to my father, Anup Singh Gusain, who taught me to never underestimate myself. A heartfelt thanks goes to my fiancée Meenu Lodhi for waking me up every morning, making me a better person and teaching me patience. I am also grateful to my friends in India, Ashish, Gaurav, Ridhi, and Shubhita for their presence and constant support.

Finally, I gratefully acknowledge the generous financial support provided by Science Foundation Ireland through the CONNECT Research Centre for Future Networks under Grant No. 16/IA/4610, which made this research possible. I am also thankful to Trinity College Dublin and the CONNECT Centre for their continued support throughout this journey.

Contents

List of Figures

Chapter 1

Introduction

The improvements observed in large language models (LLMs) and modern machine learning systems are largely driven by access to extensive natural language text corpora. This training has allowed these models to achieve strong performance across tasks such as translation, summarization, and question answering (Abdou et al., 2021; B. Z. Li et al., 2021). However, these datasets often contain sensitive information, including personal identifiers, medical records, political opinions, and demographic details, which adversaries may extract either directly from the text or indirectly through models trained on it (Staab et al., 2024; Xu et al., 2020). As LLMs are increasingly deployed in real-world applications, protecting user privacy has become essential. This motivates the development of techniques to detect, remove, or obfuscate sensitive information in natural language corpora before it can be disclosed.

This thesis examines strategies to minimize the leakage of private information from natural language text and from models trained on such data. We approach this problem from two complementary directions. First, we investigate how to design more effective redaction techniques for natural language text and how to rigorously quantify their privacy benefits. Second, we explore whether large language models develop internal representations about the data from training sentences alone , and, if so, how such information can be controlled or removed.

The first part of this thesis studies redaction as a direct approach to text sanitisation. Three main strategies have been proposed for privacy-preserving natural language processing. The first applies noise to model gradients during training, for example through Differentially Private Stochastic Gradient Descent (DP-SGD) (Abadi et al., 2016). While these methods provide formal guarantees, they often require noise levels that substantially degrade model performance (Bagdasaryan et al., 2019; Jaiswal & Provost, 2019). The second strategy sanitises the text itself, for example by applying word-level differential privacy, where each word is mapped to an embedding, perturbed with noise, and then mapped back to text (Chen et al.,

2023; Feyisetan et al., 2020; Yue et al., 2021). However, this process is poorly understood for modern neural embeddings, and treating words independently without considering correlations often overestimates the privacy gains (Mattern et al., 2022). The third strategy, synthetic text generation, trains a language model on sensitive data with DP-SGD to produce privacy-preserving text. This suffers from several difficulties (Libbi et al., 2021). DP only ensures that the addition or removal of a single sentence does not substantially affect the output, offering no protection against classification attacks such as detecting cancer-related text. Achieving stronger privacy requires heavy noise, which either degrades text quality or demands much larger datasets, increasing cost.

Redaction replaces selected words with an uninformative mask token, concealing sensitive content while maintaining sentence structure. We develop methods to quantify the privacy benefits of redaction using Rényi-divergence and propose algorithms to efficiently select words for masking in order to maximise privacy while preserving utility. Experiments across diverse datasets show that redaction can offer strong privacy protection at relatively low masking levels, providing a practical and interpretable solution.

The second part of this thesis investigates whether LLMs develop internal representations about the data from training sentences. This work is motivated by the world-representation hypothesis, which suggests that LLMs may construct internal models of the world from their training data (Karvonen, 2024; K. Li et al., 2024). If such representations exist, they could also include sensitive information, which adversaries might exploit. It is therefore important to assess the extent to which such information is present and to consider possible strategies for its removal.

To this end, we evaluate whether graphs can be reconstructed from the predictions of trained LLMs and whether their internal activations capture graph structure. These experiments are conducted under both in-context and zero-shot learning approaches.[1] Our results show that LLMs are able to reconstruct graphs and achieve higher probe accuracy when explicit contextual information is provided. In contrast, under zero-shot conditions, such graph-like structures are not observed. This indicates that LLMs do not independently form robust internal graph representations, but instead rely heavily on contextual information supplied during inference.

1.1 Research Questions

This thesis addresses the following overarching research question:

How can privacy risks arising from natural language text and from models

[1] In in-context learning, the model receives both queries and examples as input; in zero-shot learning, it only receives the query.

trained on such data be mitigated, while preserving the utility of the data and the resulting models?

To address this overarching question, the thesis investigates the following two sub-research questions:

- **RQ1:** How can efficient text redaction techniques be developed to remove sensitive words from natural language text, including information revealed through surrounding context, while minimising the amount of text removed?
- **RQ2:** To what degree do large language models develop internal representation about the data from training data alone, and how can such representations be removed if they arise?

1.2 Contributions

The main contributions of this thesis are as follows:

- **Chapter 2: Plausible Deniability of Redacted Text.**

  We propose an empirical framework to quantify privacy benefits from redacted datasets. Viewing a corpus as a probability distribution over word sequences, we measure differences between datasets using Rényi-divergence. We show that when sufficient words are redacted, sensitive and non-sensitive text can be made statistically indistinguishable.

  This chapter is based on the following publication

  Publication: Gusain, V., & Leith, D. (2025, April). Plausible Deniability of Redacted Text. In Computer Security. ESORICS 2024 International Workshops: DPM, CBT, and CyberICPS, Bydgoszcz, Poland, September 16–20, 2024, Revised Selected Papers, Part I (Vol. 15263, p. 50). Springer Nature.

- **Chapter 3: Text Redaction for Privacy Preserving Natural Language Processing.**

  We demonstrate that models trained on redacted corpora achieve a superior privacy-utility trade-off compared to differential privacy approaches. By directly modifying text, redaction avoids the performance degradation associated with DP-SGD.

  This chapter is based on the following publication

  Publication: Gusain, V., & Leith, D. Text Redaction for Privacy Preserving Natural Language Processing. Under Review.

- **Chapter 4: Improving Privacy Benefits of Redaction.**

  We present a novel redaction method that uses a neural network ranker trained with a KL-divergence loss to efficiently select words for masking. This approach achieves stronger privacy guarantees than prior methods at much lower masking levels, reducing the number of words removed while preserving textual coherence.

  This chapter is based on the following publication

  Publication: Gusain, V., & Leith, D. Improving Privacy Benefits of Redaction. ESANN 2025.

- **Chapter 5: Do LLMs Learn Graph Representations Without Context?**

  We investigate whether LLMs develop internal representations of underlying graph structures from training data alone. Using graph-based tasks under both in-context and zero-shot learning conditions, we find that LLMs do not form robust internal graph representations in zero-shot settings. Instead, their predictions and internal activations are strongly influenced by the contextual information provided during inference. These findings indicate that the models' performance is primarily driven by context utilization rather than independent internal modeling of structural knowledge, which has implications for assessing potential privacy risks.

  This chapter is based on the following publication

  Publication: Gusain, V., & Leith, D. Do LLMs Learn Graph Representations Without Context? Under Review.

Chapters 2–4 examine text redaction as a practical mechanism for improving privacy protection in natural language data, while Chapter 5 complements this analysis by investigating whether large language models learn latent structural information from training data alone, thereby assessing the broader limits of text-level sanitisation.

Chapter 2

Plausible Deniability of Redacted Text

In this chapter we propose viewing a corpus of text as a probability distribution over sequences of words. A sentence is then one realization from this distribution and redacting words changes the probability distribution. We use the Renyi-divergence divergence as a measure of the distance between two redacted datasets. We show that if enough words are redacted then sensitive redacted text can be made be statistically indistinguishable from non-sensitive redacted text. This can be used to develop efficient redaction strategies, that minimise the amount of redaction while meeting a privacy target.

2.1 Introduction

Training of neural nets such as large language models requires the availability of natural language text training data. In this chapter we revisit the question of how to sanitise sensitive text data so that it can be used for model training while preserving privacy. We introduce a new approach for quantitatively estimating the privacy gain from text redaction and demonstrate its usefulness on a wide range of datasets. This can be used to develop efficient redaction strategies, that minimise the amount of redaction while meeting a privacy target. The approach is closely related to differential privacy, but differs in several respects that are important for text data.

There are two main approaches to enhancing privacy when training machine learning models. These approaches are complementary, and both can be used together. One is to add noise to the gradient updates used during training, e.g. see DP-SGD (Abadi et al., 2016) (Shokri & Shmatikov, 2015) and related work. The other is to sanitise the training data itself. Here we focus on the latter.

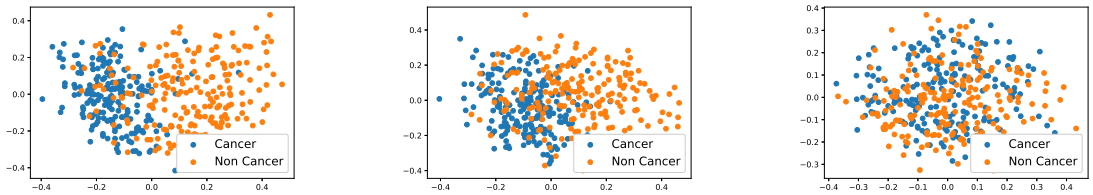
When the training data is numeric then the addition of appropriate noise, e.g.

Laplacian noise scaled proportionally to the differential privacy (DP) "sensitivity" of the data, can be used to enforce differential privacy guarantees(Dwork et al., 2006). While it is tempting to apply this approach to text by mapping the text to a numeric embedding vector, adding noise, and then mapping back to text (Feyisetan et al., 2020) (Chen et al., 2023) (Yue et al., 2021), this creates a host of unpleasant issues. For example, the nature of the mapping from text to vectors (which texts are mapped to vectors near or far away from one another) directly affects the impact of added noise, and so privacy. However, this is poorly understood, especially for modern embedding approaches based on neural networks. Another deeply problematic issue is that the words in a sentence tend to be correlated in complex ways, making any privacy approach based on individual words tend to overestimate the privacy gained.

In practice, the most popular approach for sanitising text data is redaction i.e. replacing selected words with an uninformative mask token. This is, for example, already widely used to remove personally identifying information (PII), e.g. names and addresses, from documents (Murugadoss et al., 2021). However, other aspects of the text data can also be sensitive. For example, the text data may reveal sexual/gender traits, political preferences, health-related information/concerns, social and racial characteristics, etc of the user population from which the data was gathered. Textual style can also act as a personal fingerprint facilitating de-anonymisation and membership attacks against neural nets trained on this data.

Protecting privacy is especially challenging with text data because simply redacting specified keywords is rarely enough: the surrounding context can easily continue to reveal sensitive information (H. Brown et al., 2022). For example, in the sentence "I am diagnosed with cancer. I have to go to St Lukes for chemotherapy and will probably lose my hair" redacting "cancer" and "chemo" is not sufficient to conceal the cancer diagnosis if St Lukes is known to be a cancer care hospital. If "St Lukes" is also redacted, the combination of "diagnosis" and "lose my hair" is still enough to indicate a cancer diagnosis with high probability.

In summary, there is an urgent need more effective methods for improving the privacy of text data. Given the challenging nature of natural language text, it is probably too much to hope for theoretical guarantees but that should not stop us from trying to develop useful methods motivated by theoretical analysis. In this chapter we take a step in that direction. We use redaction to add "noise" to text and by staying within original text domain thereby avoid most of the issues with numerical embeddings, and by working with text corpuses rather than individual words or sentences we can accommodate the word correlations within sentences.



(a) No redaction (b) 20% redaction (c) 40% redaction

Figure 2.1: Illustrating the increasing overlap between the semtence embeddings for cancer and non-cancer text from the Medal dataset as the level of redaction is increased. SentenceBERT embeddings are projected to two dimensions using PCA, random redaction is used.

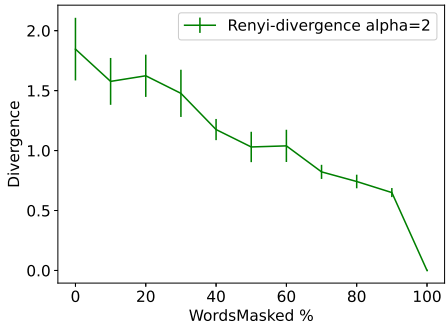


Figure 2.2: Measured Renyi vs random redaction level for Medal dataset.

2.1.1 Our approach

A dataset $\mathcal{D}$ is a collection of items. Each item is a sequence $x = (x_1, ..., x_{|x|})$ of words x_t belonging to a fixed vocabulary and with length $|x| \leq N$, N being the maximum admissible length. A redaction policy $\pi_p(x)$ maps sequence x to a new sequence where some words have been redacted i.e. replaced by an uninformative mask token `MASK`. We will assume that every redaction policy is parameterised by a parameter p taking a value between 0 and 1 such that when $p = 0$ then no words are redacted, when $p = 1$ then every word is redacted. For example, the uniform random $\pi_{rand,p}(x)$ redaction policy redacts each word in sequence x with probability p. Alternatively, we might rank the words in our vocabulary by their sensitivity and redact the top p fraction of these.

Each item x in a dataset is a random draw from a probability distribution $P(x)$ over sequences of words. After redaction, each element x is mapped to a new sequence $redact(x)$ and the redacted dataset becomes a sample from probability distribution $redact(P)$. We measure the distance between two redacted datasets $redact(\mathcal{D}_0)$ and $redact(\mathcal{D}_1)$ by the smallest value of $\epsilon_{safe} \geq 0$ such that $\tilde{P}_0(y) \leq e^{\epsilon_{safe}}\tilde{P}_1(y) + \delta$ and $\tilde{P}_1(y) \leq e^{\epsilon_{safe}}\tilde{P}_0(y) + \delta$ where $\tilde{P}_0 := redact(P_0)$ is the probability distribution over token sequences in dataset $redact(\mathcal{D}_0)$, $\tilde{P}_1 = redact(P_1)$ in dataset

$redact(\mathcal{D}_1)$ and y is any redacted sequence of words with length $|y| \leq N$.

This distance measure is similar to that used in (ϵ, δ)-differential privacy but with the difference that the set of neighbouring databases now consists of the single database $redact(\mathcal{D}_0)$ rather than all databases differing from $redact(\mathcal{D}_1)$ by a single element. When $\epsilon_{safe}, \delta_{safe}$ are sufficiently small, the publication of private dataset $redact(\mathcal{D}_1)$ then only provides an attacker with limited new information over and above that already available from the public dataset $\mathcal{D}_0$. That is, we gain privacy in the sense of *indistinguishability* between the $redact(\mathcal{D}_0)$ and $redact(\mathcal{D}_1)$ datasets. It will prove convenient to work in terms of the Renyi-divergence $D_\alpha(\tilde{P}_0||\tilde{P}_1)$ to calculate the distance between the datasets. We then convert this to an $(\epsilon_{safe}, \delta_{safe})$-privacy guarantee using equations (2.2) and (2.3).

Note. We convert the estimated Rényi-divergence curves into $(\epsilon_{safe}, \delta_{safe})$ privacy estimates using zero concentrated differential privacy. Importantly, these should not be interpreted as traditional differential-privacy (ϵ, δ) parameters: in our setting, the neighbouring datasets are the safe and sensitive datasets, not datasets that differ by a single record. See Section-2.4.3 and Section-2.4.1 for more details.

We assume the availability of a "safe" dataset $\mathcal{D}_0$ e.g. a public dataset that is suitably diverse and non-sensitive. Applying redaction policy π_p to both the sensitive dataset $\mathcal{D}_1$ and the safe dataset $\mathcal{D}_0$ then we expect that distance ϵ_{safe} between the datasets decreases as the level of redaction increases, the distance becoming zero after redaction with $p = 1$.

Figure 2.2 illustrates this for the Medal dataset of medical records (see below for further details). The original dataset is split into a dataset $\mathcal{D}_1$ of cancer patients and a dataset $\mathcal{D}_0$ of non-cancer patients. Random redaction is used. The figure shows the measured Renyi-Divergence $D_{max}(P_0||P_1)$ between the empirical probability distributions P_0 and P_1 induced by $\mathcal{D}_0$ and $\mathcal{D}_1$ as the level of redaction is varied. As expected, it can be seen that the divergence decreases as the amount of redaction increases i.e. the two datasets become more similar.

Figure 2.1 illustrates this behaviour more visually. Redacted sentences are mapped to embedding vectors using SentenceBERT (Reimers & Gurevych, 2019), the vectors are then projected onto to dimensions using PCA and shown as a scatter plot. It can be seen that without redaction the sentence embeddings have little overlap but as the level of redaction increases the overlap between the embeddings increases, indicating that distinguishing between the two datasets is becoming harder.

We make the following observations. (i) Redaction sanitizes the text data itself (rather then vector embeddings) and so yields a sanitized dataset that can be used for training ML models that take word sequences as input. (ii) By working in terms of the probability distribution over word *sequences* we take account of the correlation between the words in a sentence (the word context) and the associated

potential for leakage of sensitive information. (iii) Sanitising the dataset is akin to local differential privacy i.e. the input to a query is perturbed to ensure privacy rather than the output of the query being perturbed. (iv) As the distance ϵ between the sanitized dataset and the safe dataset decreases, privacy increases but of course we expect utility to decrease (the added value of the new dataset decreases).

2.2 Related work

The existing literature on enhancing the privacy of text data can be roughly categorised as follows:

Redacting PII. Much of the literature on text redaction has focussed on redacting personally identifying information (PII). For example,(Murugadoss et al., 2021) uses an ensemble of deep learning methods to detect and redact PII information from the medical notes of the patient,(Bosch et al., 2020) considers the discovery of names, home towns, etc in student discussion boards. Other recent work includes(Doudalis et al., 2017) (Zhao et al., 2022) (Shi et al., 2022).

Word-Level DP. Word-level DP approaches map an individual word to a vector embedding, add noise and then either map back to a new word or use the noisy embedding directly. See e.g. (Chen et al., 2023; Feyisetan et al., 2020; Yue et al., 2021). A typical choice of vector embedding is Glove(Pennington et al., 2014). The choice of vector embedding has to made up front and its properties affect the privacy gained[1] in ways that remain very poorly understood. Words are discrete quantities and the impact of quantisation when mapping from vectors back to words also remains poorly understood. The DP "database" is a sentence and the words are the database entries. The DP guarantee (modulo concerns regarding the embedding already noted) therefore relates to insensitivity to an individual word in a sentence. However this DP analysis ignores correlations between the words in a sentence and so can underestimate the information release. The impact of correlations on DP is well known and was first noted by(Kifer & Machanavajjhala, 2011). See(Mattern et al., 2022) for further discussion on the deficiencies of word-level DP.

Synthetic Text Generation. Synthetic text generation trains a generative language model on the sensitive text. The model is trained using DP-SGD (i.e. noise is added to the gradients). The model is then used to generate sanitised text. This suffers from a number of difficulties e.g. see (Libbi et al., 2021). Firstly, the DP privacy provided is that addition/removal of any one sentence from the training dataset won't change the character of the generated synthetic text much but it provides no

[1] Adding noise to an embedding perturbs it to nearby words, the way in which words are mapped to be close together (or far apart) therefore directly affects the output of the word-level DP sanitisation process.

guarantees of protection against the classification attacks we consider e.g. whether text is from cancer data or not. Secondly, addition of sufficient noise to provide a reasonable level of DP privacy either degrades the model performance (and so the quality of the synthetic text) or requires use of much more training data (increasing cost). Thirdly, in many applications synthetic text is not admissible.

2.3 Threat model

We consider the use of natural language text datasets for training machine learning models. We assume that the sensitive dataset itself is stored securely, but the sanitized/redacted dataset is publicly released. The main threat we consider is that that the sanitized dataset may be used to infer sensitive user traits e.g sexuality, health conditions, political preferences. This is a particularly topical concern since the development of LLMs is currently being driven by companies with a commercial interest in identifying user traits e.g for use in targeting adverts or other services. We assume that PII (names, addresses etc) has been removed, there being many existing techniques for this, see e.g. (Doudalis et al., 2017) (Zhao et al., 2022) (Shi et al., 2022)

2.4 Preliminaries

The Renyi-divergence of order $\alpha > 1$ between two probability distributions P_0 and P_1 on sample space Y is (Noshad et al., 2017):

$$D_\alpha(P_0||P_1) = \frac{1}{\alpha - 1} \log \int_Y P_0(x)^\alpha P_1(x)^{1-\alpha} dx \tag{2.1}$$

and similarly for $D_\alpha(P_1||P_0)$. When $\alpha = 1$ the Renyi-divergence equals the KL-divergence (Mironov, 2017).

We say that two probability distributions P_0 and P_1 are (ξ, ρ)-zero-concentrated differentially private when:

$$D_\alpha(P_0\|P_1) \leq \xi + \rho\alpha \tag{2.2}$$

for all $\alpha > 1$. In the differential privacy literature P_0, respectively P_1, is the probability distribution induced by a randomised mechanism $\mathcal{M}$ applied to dataset $\mathcal{D}_0$, respectively $\mathcal{D}_1$(Bun & Steinke, 2016). Inequality (2.2) is then required to hold for all neighbouring datasets $\mathcal{D}_0$, $\mathcal{D}_1$, where datasets are neighbours if they differ by a single element. However, here we will consider other choices for P_0 and P_1.

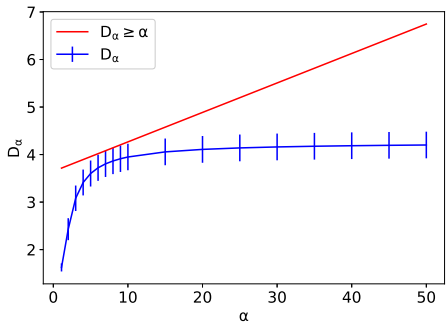


Figure 2.3: Divergence vs α for non-redacted cancer and non-cancer text from Medal medical dataset.

Specifically, they will be the distributions from which two datasets $\mathcal{D}_0$ and $\mathcal{D}_1$ are sampled.

When P_0 and P_1 are (ξ, ρ)-zCDP then from the proof of Lemma 21 in (Bun & Steinke, 2016),

$$P_1(x) \leq \exp(\epsilon_{safe})P_0(x) + \delta \tag{2.3}$$

for every $\delta > 0$ where $\epsilon_{safe} = \xi + \rho + 2\sqrt{\rho \log \frac{1}{\delta}}$. We will use (2.3) to map from Renyi-divergence curves to an (ϵ, δ) privacy guarantee.

2.4.1 Calculating (ϵ_{safe},δ_{safe})

To calculate ξ and ρ in equation (2.2), we first calculate D_α for a range of α values[2]. We then find a line that lies above the D_α vs α curve and select ρ as the slope of the line and ξ the intercept. Of course, many lines lie above the D_α curve, so we try to select one such that ρ and ξ are minimised. See for example Figure-2.3, which shows D_α vs α for the Medal medical dataset (the blue curve). This curve is upper bounded by the red line. The values of ρ and ξ corresponding to this red line are plugged into equation (2.3 to obtain the corresponding (ϵ_{safe},δ_{safe}) privacy values.

Note. We use $\delta_{safe} = 0.00008$ in all of our experiments as it is encouraged to keep $\delta < \frac{1}{n^2}$ where n is the number of input points (Abadi et al., 2016).

2.4.2 Estimating Renyi-Divergence

To estimate the Renyi-divergence between two datasets we extend the estimator of (Noshad et al., 2017), which is observed to scale well for high dimensional data. We updated the estimator to handle duplicate word-sequences, since these can become common following redaction. In addition, each word sequence is mapped to

[2]We select the range to be large enough that D_α no longer increases as we increase α.

a vector embedding to work well. These are then fed to the estimator to calculate the Renyi-divergence. We use boot-strapping to calculate confidence intervals for the estimate. Namely, we sample with replacement n times from $redact(\mathcal{D}_0)$ and $redact(\mathcal{D}_1)$, estimate $D_\alpha(P_0||P_1)$ is calculated for each sample and then the mean and standard deviation of these n estimates calculated. We select n by calculating the mean and standard deviation vs n and selecting a value large enough that these are convergent.

2.4.3 Renyi-Divergence Estimator

The Renyi-divergence of order α between two probability distributions P_0 and P_1 on sample space Y is given by) equation-2.1. We need to accommodate probability distributions with both discrete and continuous parts since after redaction there are often duplicate word sequences within a corpus of text (in the extreme case where all words are redacted every sentence becomes the same `MASK` token), corresponding to a discrete part in the probability distribution. We therefore decompose P_0 into continuous pdf p_0 and discrete probability mass function Q_0, similarly P_1, and then use a Monte Carlo plug-in estimator based on equation defined above.

In more detail, we draw N_0 and N_1 word sequences from datasets D_0 and D_1 respectively. We assume that these datasets are a representative sample from the probability distributions P_0 and P_1 respectively. We then map the i'th word sequence to a vector embedding [3] X_i and round the elements of this vector to cluster duplicates. Letting $Y_0 = \{X_i\}$ be the resulting multiset of vectors from D_0 (a multset can include duplicates) and $Y_0^u \subset Y_0$ be the set of unique vectors in Y_0, we then estimate $D(P_0||P_1)$ using

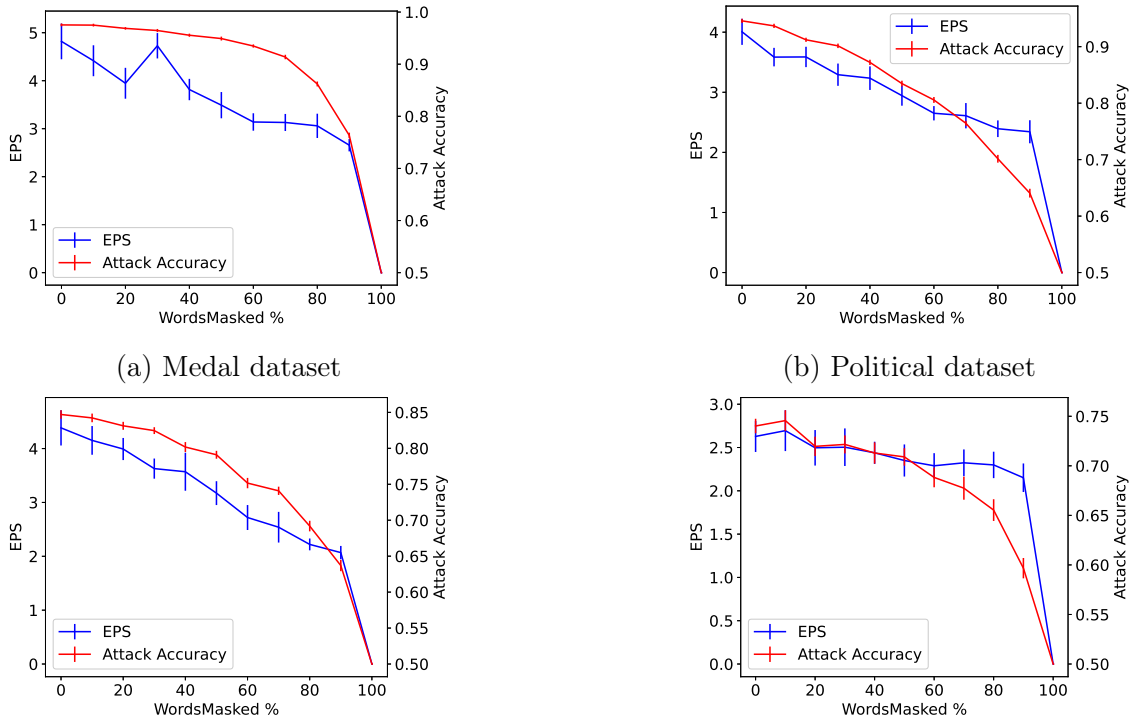
$$\hat{D_\alpha}(P_0||P_1) = \frac{1}{\alpha - 1} \sum_{y \in Y_0} \frac{1}{N_0} \log \frac{(\hat{P}_0(y))^\alpha}{\hat{P}_1(y)^(\alpha - 1)}$$

with

$$\hat{P}_0(y) = \frac{n(y, Y_0^u)}{N_0' \rho(y, Y_0^u)^d}, \hat{P}_1(y) = \frac{n(y, Y_1^u)}{N_1' \rho(y, Y_1^u)^d}$$

where $n(y, Y^u) = \sum_{x \in N_k(y, Y^u)} |x|$ counts the nearest neighbours of y in set Y^u, $N_k(y, Y^u)$ is the set of k'th nearest neighbors to y in set Y^u and $|x|$ is the multiplicity of x in multiset Y i.e. $n(y, Y^u)$ takes duplicates into account. $N_0' = \sum_{y \in Y_0^u} n(y, Y_0^u)$, $N_1' = \sum_{y \in Y_1^u} n(y, Y_1^u)$ are normalising constants, $\rho(y, Y^u) = \max_{x \in N_k(y, Y^u)} \|x - y\|$ is

[3] The choice of embedding will, in general, affect the estimated divergence. This can be mitigated by calculating the divergence for many different embeddings and using the worst-case (i.e. largest) value. However, we found the impact to be relatively minor in practice and Sentence-BERT (Reimers & Gurevych, 2019)



(a) Medal dataset

(b) Political dataset

(c) Amazon dataset

(d) Reddit dataset

Figure 2.4: Measured ϵ_{safe} between redacted sensitive and safe datasets vs redaction level; random redaction. A lower value indicates better privacy. Also shown is the measured accuracy of a classification attack that tries to label which dataset the redacted sensitive text originated from (lower accuracy therefore equals greater privacy, with a classification accuracy of 50% corresponding to a random classifier).

the distance to the k'th neighbor and d is the dimension of the vector embeddings. This kNN approach extends the continuous pdf estimator of(Noshad et al., 2017) to include probability distributions with both discrete and continuous parts. The estimator in(Noshad et al., 2017) is known to be consistent and to scale well to high dimensional data (embedding are typically have dimension d around 1000).

Note. A pseudo-code for the estimator is provided in the appendix.

2.5 Experiments

2.5.1 Datasets

We evaluate performance using the following datasets, each of which we split into "sensitive" and "safe" datasets.

(i) Medal dataset(Wen et al., 2020)[4]. This dataset contains abstracts of medical papers, along with the diseases the abstract talks about. We partition this dataset

[4]https://huggingface.co/datasets/medal

into text with cancerous and non-cancerous diseases. Each dataset contains 2200 sentences. For our experiments, text with cancerous diseases was chosen to be the sensitive dataset.

(ii) Political dataset-(Voigt et al., 2018)[5]. This contains comments on Facebook posts from 412 members of the United States Senate and House. Each comment is labeled with the corresponding Congressperson's party affiliation i.e. S ϵ {democratic, republican }

We partition the dataset into text from users with Republican and Democrat political preferences. Each dataset contains 2000 sentences. For our experiments, text from users with Republican political preferences is chosen to be the sensitive dataset.

(iii) Amazon dataset[6]. This dataset contains product reviews from Amazon customers. We selected the reviews which were categorised as "drug-store" and "kitchen-appliances". For our experiments, the dataset with drug-store reviews was chosen to be the sensitive dataset.

(iv) Reddit dataset[7]. This dataset contains post content from the subreddits r/depression and r/SuicideWatch. We partition this data into posts related to suicide and depression. Each dataset contains 2000 sentences. For our experiments, the text from the suicide subreddit was chosen to be the sensitive dataset.

2.5.2 Enhancing Privacy By Redaction

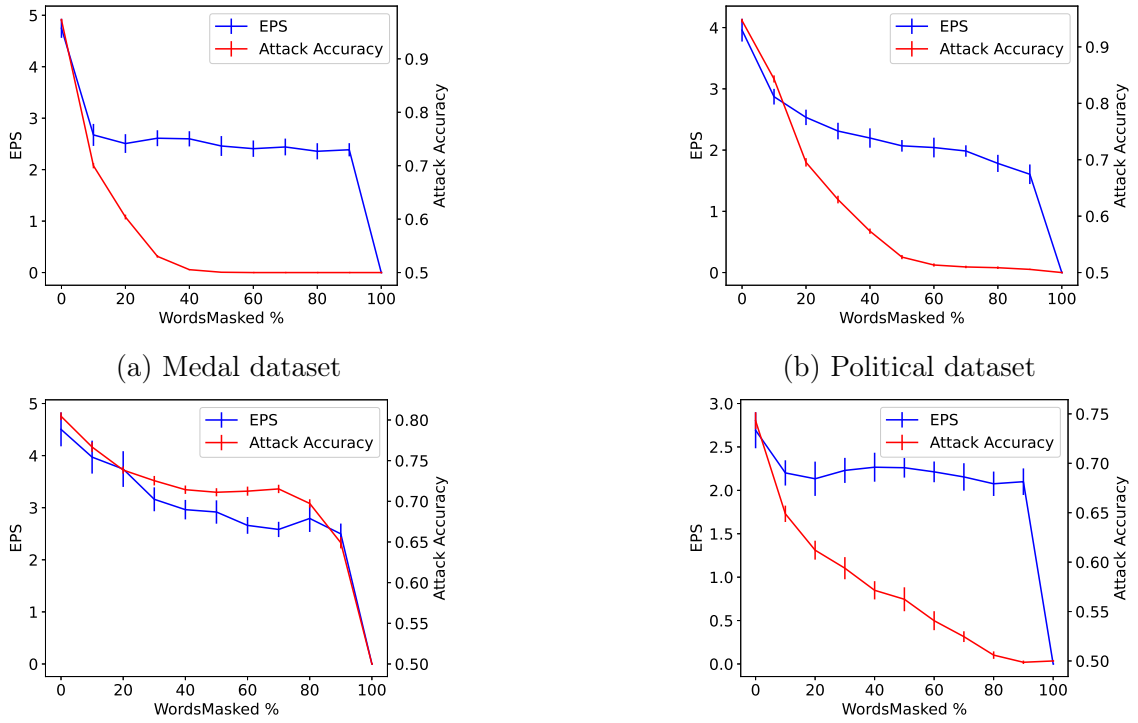
For each of the datasets we measured the Renyi-divergence as the percentage of words redacted using a random redaction policy was varied from 0 to 100%. The divergence values were then converted to ϵ_{safe} values as explained previously. The results are shown in Figure 2.4.

To help gain confidence that redaction really is improving privacy, we carry out a simple classification attack. The redacted datasets are split into training and test data (90:10 split). A classifier is trained on this data, taking a sequence of words as input and outputting an estimate of whether the sentence came from the safe or sensitive datasets. Since there are only two classes and the data is balanced, a classification accuracy of 50% corresponds to a random classifier. Figure 2.4 shows how the measured accuracy of this classifier varies with the redaction level for each of the datasets. It can be seen that the accuracy decreases as the redaction level increases, and that this decrease is roughly proportional to the decrease in the measured ϵ_{safe} value.

[5]Data can be downloaded by following the instructions in the repository https://github.com/xuqiongkai/PATR

[6]https://huggingface.co/datasets/amazon_reviews_multi

[7]https://www.kaggle.com/general/256134



(a) Medal dataset

(b) Political dataset

(c) Amazon dataset

(d) Reddit dataset

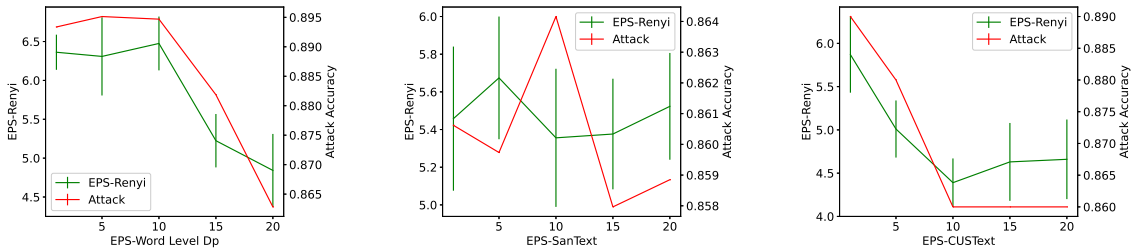
Figure 2.5: Measured ϵ_{safe} between redacted sensitive and safe datasets vs redaction level; more efficient redaction strategy. A lower value indicates better privacy. Also shown is the measured accuracy of a classification attack that tries to label which dataset the redacted sensitive text originated from (lower accuracy therefore equals greater privacy, with a classification accuracy of 50% corresponding to a random classifier).

2.5.3 Smarter Redaction

It can be seen from Figure 2.4 that when random redaction is used then relatively high levels of redaction are needed to ensure smaller ϵ_{safe} values. Of course random redaction is rather crude, and smarter redaction approaches (in the sense that they achieve a target ϵ_{safe} with fewer words redacted) are certainly possible.

We illustrate the scope for smarter redaction via a simple approach based on logistic regression weights. Namely, we took the logistic regression classifier from Section 2.5.2 and ranked the words from the datasets by the magnitude of the weight assigned to them by this classifier. We then redact the top p percent of these words from the datasets when redacting at level p.

Figure 2.5 plots the measured ϵ_{safe} between the sensitive and safe datasets as the redaction level is increased in this way. It can be seen that a much lower level of redaction is now needed, compared to Figure 2.4, to obtain a given ϵ_{safe} value. Also shown in Figure 2.5 is the measured accuracy of the simple classification attack as the redaction level is varied and it can be seen that the accuracy also now falls



(a) Word-level DP (b) San-Text (c) CusText

Figure 2.6: Measured ϵ_{safe}-renyi and attack accuracy for various word-level DP approaches applied to Medal Dataset.

much more rapidly. For example, in Figure 2.4(a) a redaction level of around 90% is needed to reduce the accuracy of the attack to 60% (recall an accuracy of 50% corresponds to a random classifier, so 60% represents a high level of privacy), for the Medal dataset while with the smarter redaction strategy a redaction level of around 30% is sufficient to achieve this. Observe also that there is now a clear "knee" in the measured divergence vs redaction level curve, with the knee corresponding a low attack accuracy. It can be seen from Figure 2.5 that the behaviour is also similar for the other datasets studied.

Note. A few original sentences and corresponding redacted sentences are shown in the appendix.

2.5.4 Word-level DP

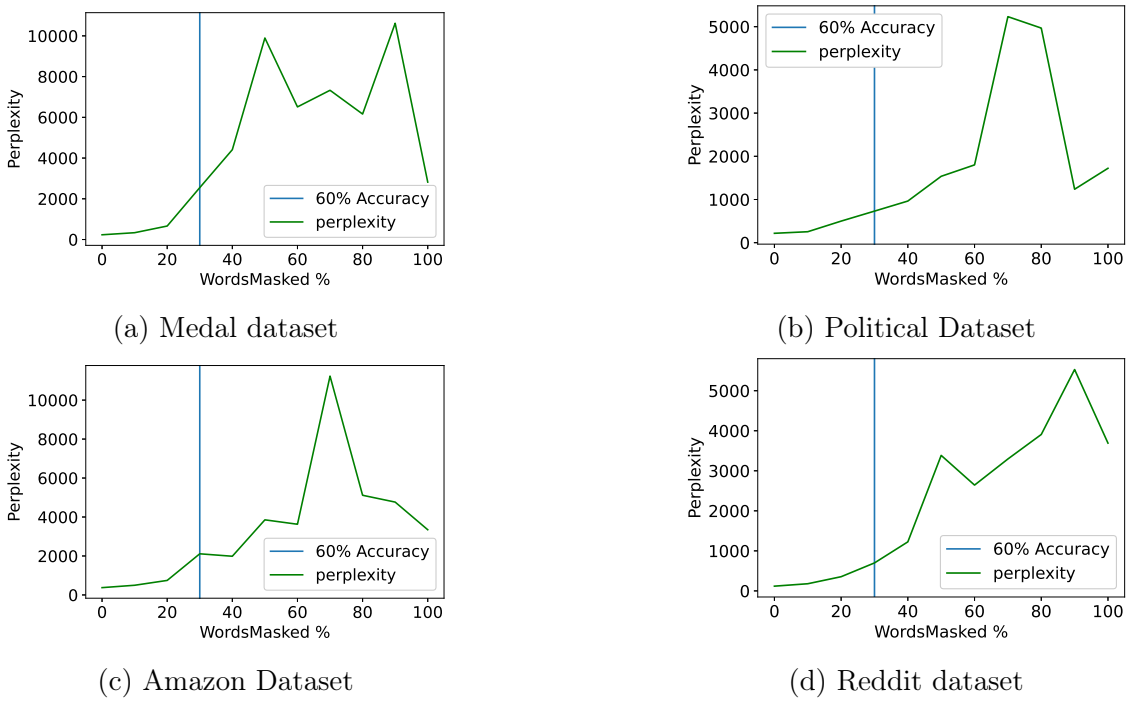
Word-level DP approaches sanitize text by converting each individual word to a vector embedding, adding noise to the embedding, and then mapping the noisy embedding back to a word (Chen et al., 2023; Feyisetan et al., 2020; Yue et al., 2021). In this section, we compare our approach to word-level DP approaches.

As discussed previously, word-level DP approaches aim to hide the information revealed by the individual words and so can fail to hide information revealed by the sentence as a whole[8].

We illustrate this by conducting the same attack as before on the word-level DP sanitized data, while also checking the ϵ_{safe} (indicated by ϵ-renyi) between the sensitive and safe datasets. A high attack accuracy indicates that sensitive information is leaked from the sanitized sentences. Similarly, a high ϵ-renyi indicates that there are significant differences between sensitive and non-sensitive datasets.

Figure 2.8 shows the measured ϵ-renyi for the Medal dataset as the level of noise is increased (indicated by ϵ) for various word-level DP approaches. Also shown is

[8]In particular, the DP analysis ignores correlations between the words in a sentence and so may greatly underestimate the information release. The impact of correlations on DP is well known and was first noted by(Kifer & Machanavajjhala, 2011).



(a) Medal dataset

(b) Political Dataset

(c) Amazon Dataset

(d) Reddit dataset

Figure 2.7: Measuring impact of privacy on utility. Next word prediction performance for LSTM trained on redacted dataset. Performance is measured on non-redacted data. The vertical line indicates the redaction level that reduces classification attack accuracy to 60%, taken from Figure 2.5.

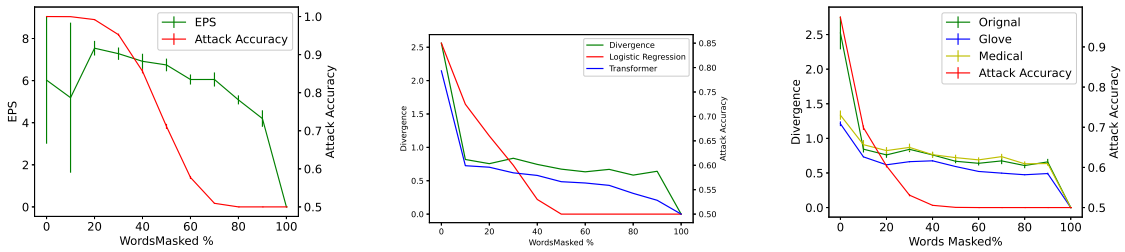
the measured accuracy of our classification attack. It can be seen that even when a great deal of noise is added (low ϵ values), both the ϵ-renyi values and the attack accuracy remain high. Results for other datasets show similar behaviour and are provided in the Appendix.

2.5.5 Utility vs Privacy

In general, we expect there to be a trade-off between privacy and utility. By redacting text to make a sensitive training dataset more private, the value of the sensitive training data is likely to be reduced because (i) redaction reduces the textual information contained in the sensitive dataset and (ii) by making the sensitive dataset more similar to an existing safe public dataset the added value over only using the public data for training is reduced.

To investigate this trade-off we trained a next-word-prediction model using PyTorch. A standard LSTM-RNN model[9] was used with two layers, each layer having 200 hidden states and an input Embedding layer. The model has 4,041,675 parameters. A dictionary of all words was created from the dataset and input text was

[9]Training code can be found at: https://github.com/pytorch/examples/tree/main/word_language_model



(a) (b) (c)

Figure 2.8: 2.8a Measured ϵ_{safe} and attack accuracy for cancer sentences when compared against IMDB reviews.2.8b Measured Renyi-divergence ($\alpha = 2$) and attack accuracy for logistic regression and BERT transformer classification attacks as the redaction level is increased. Medal dataset. 2.8c Measured Renyi-divergence ($\alpha = 2$) with different embeddings: (i) general-purpose sentenceBERT, (ii) fine-tuned medical sentenceBERT, (ii) Glove. Medal dataset.

vectorized by replacing each word with its corresponding index in the dictionary. The model was trained using a negative log-likelihood loss.

The sensitive dataset was divided into a train-validation (90:10) split. The training data was redacted while validation data was not redacted. The model was then trained on the redacted training data and was tested against the held-out validation data. The model performance was evaluated by the measured perplexity (ppl) on the validation set, the lower the value of perplexity the better the model is at predicting a given sequence.

Figure 2.7 shows the measured perplexity on the validation data of the next word prediction model for each of the datasets studied as the redaction level is varied (for the smarter redaction approach of Section 2.5.3). The vertical line in the plot indicates the point where the measured attack classification accuracy is 60% (taken from Figure 2.5). It can be seen as expected, the perplexity increases as the redaction level increases. It can be seen that the perplexity increases with the level of redaction, as expected. However, for redaction levels up to 20-30% the perplexity of the model trained on the redacted dataset remains fairly close to the perplexity of the model trained on non-redacted dataset. A redaction level of 30% is sufficient to bring the attack classification accuracy down to 60% i.e a good degree of privacy can be obtained while preserving the utility of the dataset for model training.

2.6 Discussion

2.6.1 Choice of the Safe dataset

When the safe dataset is similar to the sensitive dataset, then we can expect that only a small amount of redaction is needed to bring the two datasets closer to-

gether. Conversely, we expect that a higher level of redaction is needed to make very disparate datasets similar.

This is illustrated in Figure-2.8a, which can be directly compared with Figure-2.5a. In both cases the Medal cancer text is the sensitive dataset but in Figure-2.8a the safe dataset is IMDB review text while in Figure-2.5a it is Medal non-cancer text. It can be seen that with the IMDB data almost 80% of the words need to be redacted to get an attack accuracy close to 50% whereas with the Medal non-cancer text a redaction level of around 50% is sufficient.

The choice of safe dataset is therefore a privacy design parameter that can be used to manage the trade-off between privacy and utility in a fairly transparent manner.

2.6.2 More Powerful Attacks

It is important to stress that it is the Renyi-divergence that provides a sound measure of privacy, not the accuracy of any specific attack. In the previous sections we use a classification attack based on logistic regression to roughly verify privacy. However, this attack on its own does not provide any privacy guarantee.

For example Figure-2.8b shows the attack accuracy and Renyi-divergence as the level of redaction is varied. It can be seen that at redaction levels around 50-80% the attack accuracy is low (close to 50%, the accuracy of a random classifier). However, the Renyi-divergence remains relatively high at around 2.5 at these redaction levels, indicating that the datasets remain dissimilar.

We therefore trained a more powerful BERT transformer model and used it to carry out the classification attack. Figure-2.8b, shows the measured attack accuracy. It can be seen that at 50-80% redaction the attack accuracy now remains relatively high, demonstrating the predictive power of the Renyi-divergence approach.

2.6.3 Choice of Embedding

The estimator in Section 2.4 maps text to a vector embedding and then estimates the Renyi-divergence between sets of vectors. It therefore depends on the choice of embedding used. It is problematic for a privacy approach to depend on the choice of embedding since (i) the properties of these embeddings remain poorly understood and (ii) an attacker can easily use a different embedding. For example, if a general purpose embedding is used in the Renyi-divergence but the attacker uses a domain specific embedding, a natural concern is that attacker may be able to extract information even when the Renyi-divergence estimate is low.

One of the great advantages of redaction is that it does not depend on the choice

of embedding, but rather works directly with the text data[10]. We can then calculate Renyi-divergence estimates for different choices of embeddings and use the largest value to evaluate privacy.

For example Figure-2.8c shows the measured Renyi-divergence estimates for three different choices of embedding: (i) a general-purpose pre-trained sentence-BERT embedding [11], (ii) sentenceBERT after fine-tuning on medical data and (iii) Glove[12](Pennington et al., 2014). sentenceBERT is a state of the art transformer embedding, Glove is an older embedding commonly used on word-level DP.

It can be seen from Figure-2.8c that the Renyi-divergence estimates for the two sentenceBERT embeddings are almost the same and consistently higher than the Renyi-divergence estimate using Glove i.e. Glove overestimates privacy. The consistency in the divergence estimates between the general purpose and fine-tuned sentenceBERT embedding indicates the robustness of the general purpose sentence-BERT and is why we use it in our earlier plots.

2.6.4 Limitations

Renyi-divergence is an estimate. Probably the main limitation of our approach is that it uses an estimate of the Renyi-divergence rather than the true value. We partially mitigate this by also estimating confidence intervals. However use of an estimate seems unavoidable since the true divergence cannot be calculated for realistic text data, and for similar reasons theoretical differential privacy guarantees are intractable. We argue that adopting a pragmatic approach and using estimates provides a way forward that is both useful and represents significant progress over the state of the art in text privacy. This is particularly pressing given the prevalence of text data and the current great interest in using it to train large language models.

2.7 Conclusions

We revisit the question of how to sanitise sensitive text data so that it can be used for model training while preserving privacy. The great majority of the existing literature on privacy enhanced model training gains privacy by adding noise to gradients used for training. The amount of noise added needs to scale with the number of model parameters since the DP sensitivity scales with this. When the number of model parameters is large (as it usually is for language models), the amount of noise needed

[10] And one of the major deficiencies of all approaches tied to a single up front choice of embedding, such as word-level DP approaches.

[11] https://www.sbert.net/

[12] The embedding vector of each word in a sentence is calculated, and the mean of these vectors is used as the sentence embedding.

is considerable and adversely affects model utility. Sampling of the input data can be used to boost privacy, but requires effective anonymisation which can be hard to achieve in practice and reduces the volume of training data available. The data sanitisation approach that we consider complements this line of work and offers new ways to manage the trade-off between privacy and utlity.

Chapter 3

Text Redaction for Privacy Preserving Machine Learning

In this chapter we focus our attention on training machine learning models on the redacted datasets. Traditionally differentially private SGD (DP-SGD) is used for privacy-enhanced training of neural network models with natural language data. Since it is hard to add noise to text data, DP-SGD instead adds noise to the clipped loss function gradient used during each SGD update. However, this often leads to an unfavorable utility-privacy trade-off. In this chapter we propose the use of redaction of the training data as an alternative approach and show that this yields a much improved utility-privacy trade-off over DP-SGD.

3.1 Introduction

We revisit the privacy-enhanced training of neural network models using natural language data. The state of the art is to use Differentially Private Stochastic Gradient Descent (DP-SGD) (Abadi et al., 2016). This adds noise to the clipped loss function gradient used during each SGD update. However, this often leads to an unfavorable utility-privacy trade-off. Indeed privacy may need to largely sacrificed (i.e. large values of the differential privacy ϵ, δ parameters) in order to obtain a usable model (Bagdasaryan et al., 2019; Jayaraman & Evans, 2019; Noe et al., 2022)

We argue that there are two primary issues here. Firstly, DP-SGD adds noise to the gradients rather than to the training data. This makes sense for text training data since is not at all clear how to add suitable noise to text data. However, since DP-SGD treats each gradient as a "query" of the training data, the amount of noise added then needs to scale with the number of SGD training steps and so can quickly become substantial, see e.g. Figure-3.1. Secondly, the usual DP-SGD threat model

is overly conservative. It assumes the adversary has access to every gradient update and, importantly, can manipulate all but one element of the training data (Nasr et al., 2023). However, in many applications the adversary cannot control the training data.

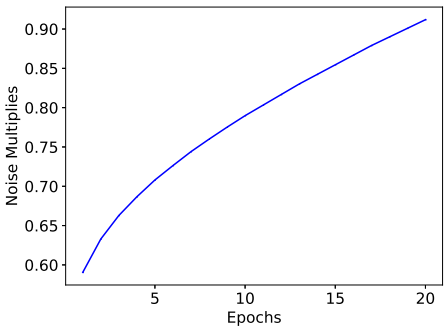


Figure 3.1: Noise Multiplier (the ratio of standard deviation of the Gaussian noise to the gradient L_2-sensitivity) for DP-SGD vs number of epochs for which training is run. Privacy budget: $\epsilon = 4$ and $\delta = 0.00008$.

In this chapter we address both of these issues when the training data is natural language text. When the training data is numerical, differential privacy can be achieved by directly adding noise to the training data and then using standard SGD to train the model. With this approach there is no need to scale the noise with the number of SGD training steps. However, as already noted, a key difficulty with applying such an approach to text data is that we do not know how to add suitable noise to text data. We sidestep this difficulty by *redacting* the training data text rather than trying to add noise i.e. by replacing selected words by an uninformative `MASK` token and coalescing adjacent tokens. As we have observed previously redaction makes disparate datasets more similar and so harder to distinguish. This can be easily seen from the extreme case where all words are redacted which reduces the datasets to the same single `MASK` token.

We also propose modifying the DP-SGD threat model as follows. We assume that the training data is a representative draw from a probability distribution over sequences of words and that a "safe" training dataset $\mathcal{D}_0$ is available, e.g. a public dataset that is suitably diverse and non-sensitive, as well as a sensitive dataset $\mathcal{D}_1$. The model is trained on the redacted sensitive dataset $redact(\mathcal{D}_1)$. The adversary may observe $redact(\mathcal{D}_1)$, but cannot modify it. We ensure privacy by selecting the redaction policy so that with high probability $redact(\mathcal{D}_0)$ and $redact(\mathcal{D}_1)$ are indistinguishable after redaction. That is, we define the redacted safe and sensitive datasets $redact(\mathcal{D}_0)$ and $redact(\mathcal{D}_1)$ as the neighbors in differential privacy rather than the usual approach of defining datasets differing by a single element as neighbors. We show that these changes can yield a 100x increase in utility compared to DP-SGD.

3.2 Related Work

DP-SGD. DP-SGD provides a differential privacy (DP) guarantee for machine learning models. During the training phase Gaussian noise is added to the clipped gradients used to update the model parameters(Abadi et al., 2016). An accountant is used to manage the overall DP guarantee of the trained model by determining the amount of noise that needs to added. Larger noise translates into greater privacy but hurts the performance of the trained model (Bagdasaryan et al., 2019; Jayaraman & Evans, 2019; Noe et al., 2022). We also observe similar behavior in Section-3.4.2 below. There is currently a good deal of effort being directed at trying to refine DP-SGD so as to reduce the loss in utility. For example, adaptive clipping clips the gradient to match a particular quantile (Andrew et al., 2022), DP-FTRL uses tree aggregation when adding noise to the gradients (Xu et al., 2023).

Text Redaction. Most of the current approaches manually redact PII from the input text (Bosch et al., 2020; Doudalis et al., 2017). These approaches focus more on hiding user specific sensitive information from the input text rather than on information that is revealed by the input text as a whole. Recent research by (Gusain & Leith, 2024), proposed an approach to limit information revealed by the text using redaction. They use a logistic regression model to rank and redact the words. They observe an increase in privacy benefits as redaction levels are increased.

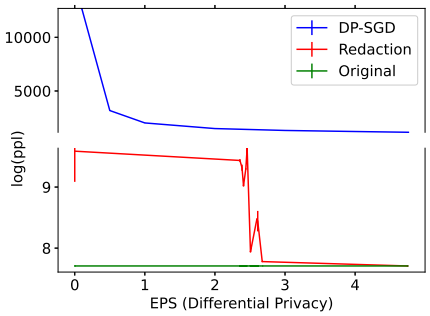
Word-level DP. Word-level DP approaches are used to sanitize the input text. These approaches map an individual word to a vector embedding, add noise and then either map back to a new word or use the noisy embedding directly. See e.g. (Chen et al., 2023; Meisenbacher et al., 2024; Yue et al., 2021). However sensitive attributes such as political views, medical condition, gender can still be leaked by the sanitized sentences (Gusain & Leith, 2024; Mattern et al., 2022).

3.3 Redacting Text Data

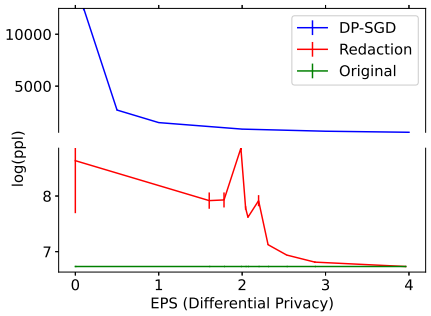
3.3.1 Threat model

A training dataset is a representative sample from a probability distribution over sequences of words. A "safe" training dataset $\mathcal{D}_0$ is available, e.g. a public dataset that is suitably diverse and non-sensitive, sampled from probability distribution P_0. There is also a sensitive dataset $\mathcal{D}_1$ sampled from probability distribution P_1. The model is trained on the redacted sensitive dataset $redact(\mathcal{D}_1)$. The adversary may observe $\mathcal{D}_0$ and $redact(\mathcal{D}_1)$, but cannot modify either.

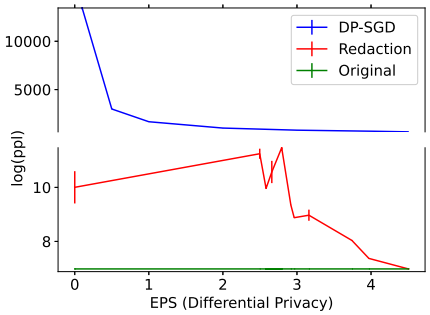
Our concern is that an attacker can use the dataset $redact(\mathcal{D}_1)$ to infer sensitive user traits such as sexuality, health conditions, political references beyond what



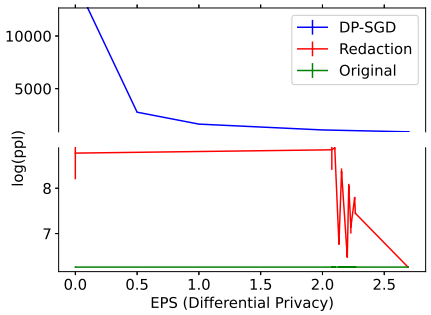
(a) Medal dataset



(b) Political dataset



(c) Amazon dataset



(d) Reddit dataset

Figure 3.2: Measured impact of privacy on utility. Next word prediction performance for LSTM trained on redacted dataset and using DP-SGD. x axis represents ϵ value of privacy guarantee and y axis represents $\log(perplexity)$The x-axis shows ϵ_{safe} for our redaction approach and the standard ϵ for DP-SGD

.

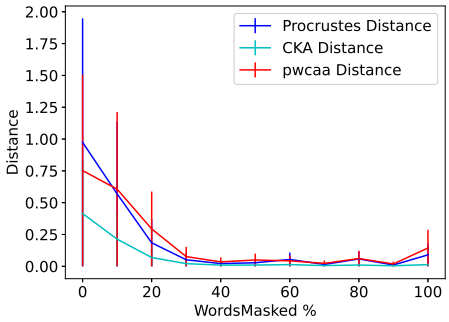
is already revealed by safe dataset $\mathcal{D}_0$. Since a model trained on $redact(\mathcal{D}_1)$ cannot contain more information than $redact(\mathcal{D}_1)$, protection against this threat also protects against inference attacks on a model trained on $redact(\mathcal{D}_1)$.

3.3.2 Redaction Policy

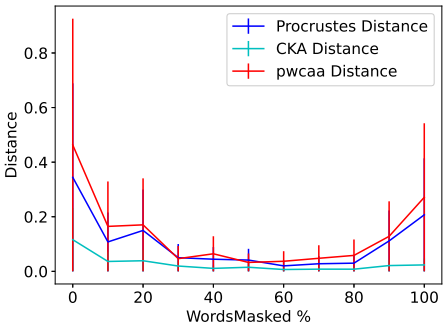
We use the smarter redaction policy[1] defined in the previous chapter to carry out redaction. We train a logistic regression classifier on the non-redacted safe and sensitive datasets. The output of the classifier is an estimate as to whether input text comes from the safe or the sensitive dataset. The trained classifier was then used to rank the words in the dataset vocabulary by the magnitude of the weights assigned to them. The Top-K% of these words are then redacted (so $p = K/100$).

We calculate Renyi-divergence between the redacted datasets, and then convert this to an $(\epsilon_{safe}, \delta_{safe})$-privacy guarantee as explained in the previous chapter.

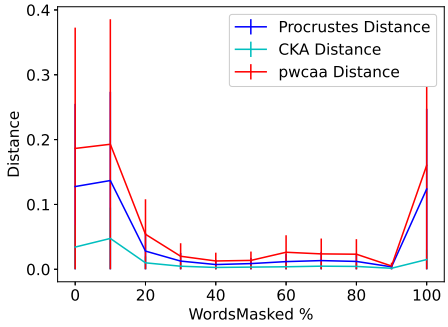
[1]redaction carried out by training a logistic regression model



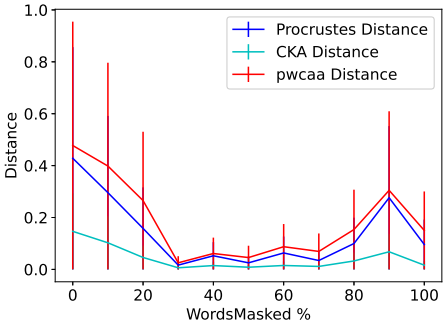
(a) Medal dataset



(b) Political dataset



(c) Amazon dataset



(d) Reddit dataset

Figure 3.3: Measured CKA,PWCAA and Procrustes distances between the encoder weights of the model trained on sensitive and safe dataset. A lower value indicates the weights are similar.

3.4 Experiments

3.4.1 Datasets

We evaluate performance on the four datasets : Medal dataset (Wen et al., 2020), Political dataset (Voigt et al., 2018), Amazon dataset (Gusain & Leith, 2024), Reddit dataset. Sensitive and safe datasets were created from these datasets as explained in (Gusain & Leith, 2024). Each dataset contains almost 10000 sentences each in the safe and sensitive dataset. The validation set for medal reddit and amazon dataset contained 2000 sentences each in the safe and sensitive dataset. For more details regarding the datasets we refer reader to Chapter-2.

3.4.2 Utility vs Privacy

In general, we expect there to be a trade-off between privacy and utility. By redacting words from the sentences to make them more private we also reduce the information present in the sentences, which makes training of a machine learning model on such data more difficult. A privacy-utility trade-off also exists for models trained using DP-SGD, i.e adding noise and clipping the gradients tends to degrade the performance of the trained model.

To investigate this trade-off we trained a next-word-prediction (nwp) model using Pytorch[2]. A standard LSTM model was used, it had two layers and each layer having 80 hidden states along with an embedding layer. A dictionary of words was created which was used to vectorize input text data. Each word was replaced with it's corresponding index in the dictionary. The model was trained using a negative log-likelihood loss.

The sensitive dataset was divided into a 90:10 train-validation split. The model was trained on the sensitive redacted training data and was tested on a held-out non-redacted sensitive dataset. The Renyi-divergence at each redaction level was converted to an $(\epsilon_{safe}, \delta_{safe})$ distance using the approach explained above with $\delta_{safe} = 0.00008$. Since the model was trained on the redacted dataset the model will inherit the privacy benefits of the underlying data. In particular, the privacy guarantee of the data upper-bounds the privacy guarantee of the trained model.

Opacus[3] was used to train the nwp model with DP-SGD. The sensitive non-redacted dataset was split into a train and validation split as explained above. The validation set was set similar to the previous setup. The model was trained for different ϵ's. Training steps for DP-SGD were set to steps required to train the non-private LSTM and all other hyper-parameters were unchanged. Values of hyper-parameter used is provided in the Appendix. For our experiments we used $\delta = 0.00008$, as it is encouraged to keep $\delta < \frac{1}{n}$ where n is the number of input examples (Abadi et al., 2016).

Figure-3.2 shows the measured $\log(perplexity)$ of the trained LSTM next word prediction model vs ϵ for each of the datasets (lower perplexity is better). Note the log-scale used on the y-axis, this is needed due to the large difference in measured perplexity observed between DP-SGD and our text redaction approach. It can be seen that the utility is consistently at least $100\times$ better when using redaction compared to DP-SGD.

When ϵ_{safe} falls below about 2 or 3 (it varies depending on the dataset), when using redaction the variance of the perplexity rapidly increases (visible as spikes in Figure-3.2) and the perplexity eventually plateaus for smaller values of ϵ_{safe}. Closer inspection shows that this transition corresponds to a redaction level of around 70%, at which point training of the model starts to degrade (e.g. the validation loss is non-decreasing). When using DP-SGD there is also a sharp rise in perplexity, but this occurs when ϵ falls below around 0.5. While this DP-SGD transition occurs for a smaller ϵ than with redaction, for all values of ϵ the perplexity is always substantially higher for DP-SGD than when using redaction. Recall that the threat model less

[2]training code can be found at https://github.com/pytorch/examples/tree/main/word_language_model

[3]framework to train models with DP-SGD using pytorch; https://opacus.ai/

conservative for redaction than for DP-SGD and this is likely the source of at least part of this performance gain.

3.4.3 Checking Closeness of Models

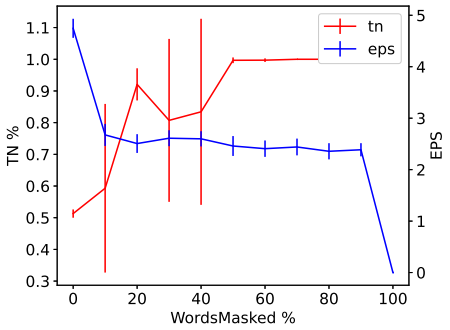
The decrease in divergence between the safe and sensitive datasets as the level of redaction is increased reflects the fact that redaction makes the sentences in those datasets becoming more similar. We therefore also expect that the weights of machine learning models trained on these datasets will also tend to become more similar as the level of redaction is increased.

It is not straightforward to measure the distance between model weights, but there is a growing line of work on computing the similarity between model weights. These algorithms either use centered kernel alignment (CKA) (**cka**) or canonical correlation analysis (CCA) (Morcos et al., 2018; Raghu et al., 2017). We trained the nwp model on the redacted safe and sensitive dataset as explained previously and then compared the trained model weights using the CKA, WCCA and Procrustes distances (Ding et al., 2021). For each redaction level, multiple models were trained using different seeds and the mean and standard deviation of the distances was calculated.

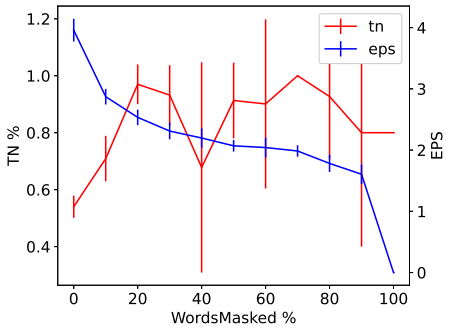
Figure-3.3 shows the mean-distance between weights of models trained on sensitive and safe data respectively. We only show the result for the first layer here, for results relating to other layers please refer to Appendix. It can be seen that as the level of redaction is increased, the distance between the weights decreases, as expected, until more than about 70% of the words are redacted. At that point we can see the distance increasing again. Our experiments show that at such high redaction levels the models are not getting trained properly (e.g. the validation loss is non-decreasing), presumably because there is not enough information present in the text data. As already noted, this can also be seen in Figure-3.2.

3.4.4 Membership Inference Attack

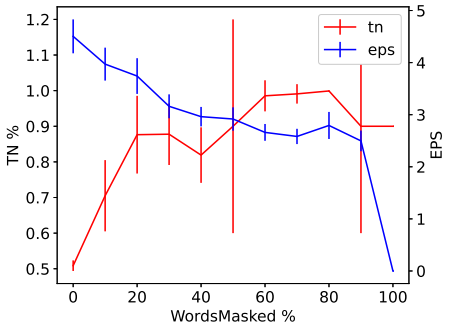
To help verify the privacy gain from redaction, we carried out a membership inference attack on the model trained on the sensitive redacted dataset, that aims to detect whether a model was trained using sensitive data or not. A dataset with held-out data (not used for model training) was created consisting of $\tilde{5}00$ sentences from the non-redacted sensitive dataset. Algorithm-1 was then used to carry out the attack on an nwp model trained using the validation set. The success of the attack is evaluated by the percentage of true negatives. A low level of true negatives essentially indicates that the attack was successful and the attacker was able to identify that sensitive information was used to train the model.



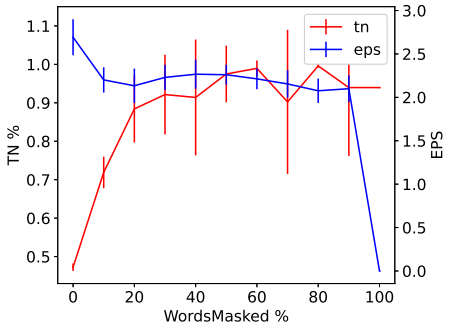
(a) Medal dataset



(b) Political dataset



(c) Amazon dataset



(d) Reddit dataset

Figure 3.4: Measured %True negatives for the model trained on redacted sensitive dataset. A lower value indicates that the attack was successful, as the model memorised sensitive information from the data. Higher value indicates that the model did not memorise any sensitive information. Also shown are the privacy ϵ values between the sensitive and safe dataset as the redaction level is increased.

Figure-3.4 shows the measured attack performance as the level of redaction is varied. It can be seen that as the level of redaction is increased the number of true negatives is also increased. That is, as the level of redaction is increased it becomes more difficult to extract sensitive information from the trained model.

3.5 Discussion

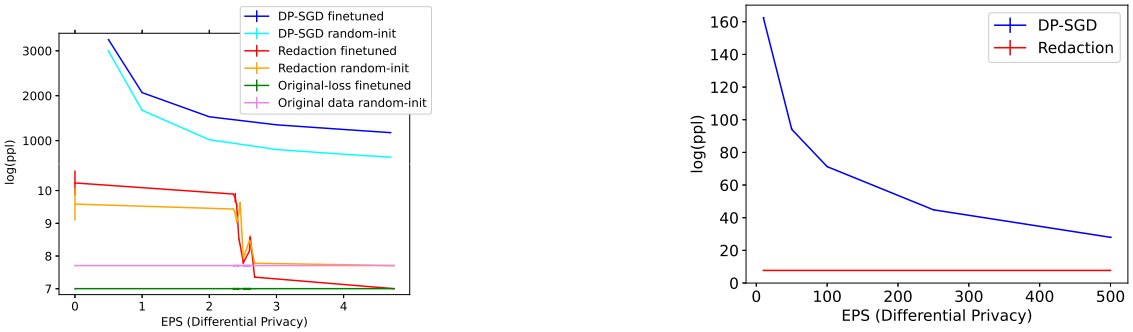
3.5.1 Model Initialization

The weights of the next-word-prediction model can be initialised using the non redacted safe dataset, which can then be fine-tuned using the sensitive dataset. In this scenario model fine-tuned using DP-SGD might provide utility similar to the original utility.

Figure-3.5a shows measured $log(perplexity)$ for LSTM trained on redacted Medal dataset and using DP-SGD, when different initialisation techniques are used. We observe that model fine-tuned on redacted sensitive dataset, provides better utility at higher epsilons (lower redaction percentage) and provide similar utility at lower

Algorithm 1 Membership inference attack. *val* is the data used to carry out the attack. Each item in the dataset is a tuple (sentence,gt). sentence is the sentence used to do the attack and $gt = 1$ if the input was present in the training data and $gt = 0$ if the input was not in the training data. *model*= trained model on which the attack is performed. *thresh* is the threshold against which we compare our loss. *loss* is the loss function used to calculate the loss of the model.

```
function MEMBERSHIPINFER(val[ ], loss, model)
    set prediction = [ ]
    N ← length(val)
    for k ← 1 to N do
        sent ← val[k]
        sentloss ← loss(model(sent))
        if sentloss <= thresh then
            prediction[k] = 1
        else
            prediction[k] = 0
        end if
    end for
    return prediction
end function
```



(a) (b)

Figure 3.5: 3.5a Measured impact of privacy on utility. Next word prediction performance for LSTM trained on redacted Medal dataset and using DP-SGD. x-axis represents ϵ value of privacy guarantee and y-axis represents $log(perplexity)$. The model was initialized using 1) Non-sensitive redacted dataset and 2) Random initialization .3.5b Measured impact of privacy on utility. Next word prediction performance for LSTM trained on redacted Medal dataset and using DP-SGD. x-axis represents ϵ value of privacy guarantee and y-axis represents $log(perplexity)$

epsilons when compared to the model trained on redacted sensitive dataset. However when the model is fine-tuned and trained using DP-SGD, utility provided is still worse than the original utility.

3.5.2 DP-SGD Convergence

At higher ϵ values we can expect the utility of the model trained using DP-SGD converging to the utility of the model trained on redacted data.

Figure-3.5b, shows measured $log(perplexity)$ for LSTM trained on redacted Medal dataset and using DP-SGD. The DP-SGD model here was trained only for one epoch. The utility of the model trained on redacted dataset remains constant for higher epsilon, however we can observe that the utility of the model trained using DP-SGD, increases as the ϵ increases. At higher ϵ values i.e ϵ 500 the utility of the model trained using DP-SGD is becomes similar to the model trained on redacted dataset.

We observe similar trend for other datasets as well. For more details please refer to Appendix.

3.6 Limitations

Probably the main limitation of our approach is that it uses an estimate of the Renyi-divergence rather than the true value. We partially mitigate this by also estimating confidence intervals. However use of an estimate seems unavoidable since the true divergence cannot be calculated for realistic text data, and for similar reasons theoretical differential privacy guarantees are intractable. We argue that adopting a pragmatic approach and using estimates provides a way forward that is both useful and represents significant progress over the state of the art in text privacy. This is particularly pressing given the prevalence of text data and the current great interest in using it to train large language models. It is worth noting we are not only in this and that there is also a growing trend towards using empirical analysis in differential privacy, e.g. for auditing and for investigating the impact of changes in the threat model (Jagielski et al., 2020; Nasr et al., 2023; Steinke et al., 2023).

3.7 Conclusion

We revisit the question of how to redact sensitive text data so that it can be used for model training while preserving privacy. The great majority of the existing literature on privacy enhanced model training gains privacy by adding noise to gradients used for training. However, this often leads to an unfavorable utility-privacy trade-off. The data redaction approach that we consider complements this line of work and offers new ways to manage the trade-off between privacy and utility.

Chapter 4

Improving Privacy Benefits of Redaction

Our redaction technique has provided better privacy benefits than the current sanitization techniques on natural language text data. The models trained on the redacted text, also provide better privacy benefits and performance than the models trained using traditional privacy preserving machine learning algorithms such as DP-SGD. However the amount of words that needs to be redacted using our smarter redaction approach still remains very high.

In this chapter we focus our attention on developing better redaction techniques. We propose a novel redaction methodology that can be used to sanitize natural text data. Our new technique provides better privacy benefits than our previously proposed redaction algorithm while maintaining lower redaction levels.

4.1 Introduction

Redaction is widely used to hide sensitive information from text. It is a process of replacing selected words with an uninformative `[MASK]` symbol and is typically carried out manually to redact Personal Identifiable information (PII) such as names addresses, etc (Murugadoss et al., 2021) from the text.

Protecting privacy is especially challenging for text data because redacting specified words is rarely enough: the surrounding context can easily continue to reveal sensitive information (H. Brown et al., 2022). Even when the sensitive text is sanitized using word-level DP approaches (Chen et al., 2023; Yue et al., 2021), it has been observed that sensitive attributes such as political views, medical condition, gender can still be leaked by the sanitized text dataset as whole (Gusain & Leith, 2024; Mattern et al., 2022).

To limit the information revealed by the text data, redaction can be carried

out such that a sensitive dataset $\mathcal{D}_0$ becomes indistinguishable from a safe dataset $\mathcal{D}_1$[1]. The privacy gained in such cases depends on the percentage of words redacted from the sentences present in $\mathcal{D}_0$ and $\mathcal{D}_1$, and can be estimated by calculating the Renyi-divergence (Noshad et al., 2017) between the distributions of $\mathcal{D}_0$ and $\mathcal{D}_1$ and converting this to an $(\epsilon_{safe}, \delta_{safe})$ differential privacy estimate using concentrated differential privacy(Bun & Steinke, 2016), for more details see (Gusain & Leith, 2024).This approach although promising, can require almost 80% of the words from the input text to be redacted in order to achieve a reasonable level of privacy. At that point there is almost no information left in the sentence and it does not have much utility left.

In this chapter we propose a novel redaction methodology which builds upon the work of the authors of (Gusain & Leith, 2024). We show that our approach provides better privacy guarantees while requiring much lower redaction levels. For example, achieve $\epsilon_{safe} = 0.01$ by only redacting 20-30% of the words, which is the considerable improvement over the smarter redaction approach introduced in the previous chapter. We also provide an open-source implementation of a KL-divergence loss (in PyTorch) which calculates KL-divergence from the sentence embeddings.

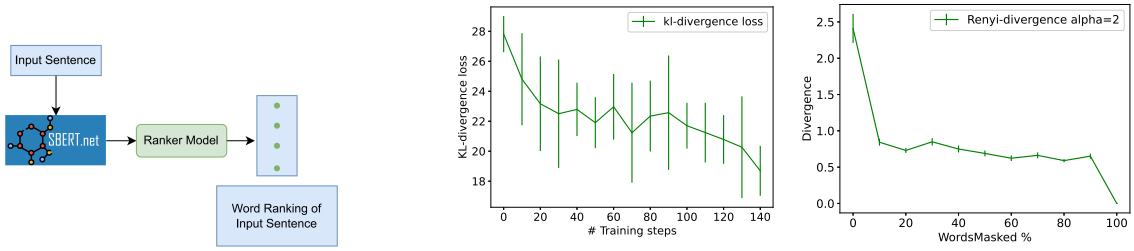
4.2 Related Work

`Text redaction.` Most of the current approaches manually redact PII from the input text (Bosch et al., 2020; Doudalis et al., 2017). These approaches focus more on hiding user specific sensitive information from the input text rather than on information that is revealed by the input text as a whole. Recent research by (Gusain & Leith, 2024), proposed an approach to limit information revealed by the text using redaction. They use a logistic regression model to rank and redact the words. They observe an increase in privacy benefits as redaction levels are increased.

`Word level DP` Another approach to santize input text is to use Word-level DP. These approaches map an individual word to a vector embedding, add noise and then either map back to a new word or use the noisy embedding directly. See e.g. (Chen et al., 2023; Yue et al., 2021). However sensitive attributes such as political views, medical condition, gender can still be leaked by the sanitized sentences (Gusain & Leith, 2024; Mattern et al., 2022).

`Renyi-Divergence` Divergence is widely used to calculate the distance between two probability distributions (Noshad et al., 2017). Renyi-Divergence of order α between two probability distributions P_0 and P_1 on sample space Y is (Noshad et

[1]Sensitive dataset $\mathcal{D}_0$ contains sentences which contains sensitive information such as a specific medical conditions etc and a safe dataset $\mathcal{D}_1$ is a public dataset that is suitable diverse and non-sensitive.



(a) (b) (c)

Figure 4.1: `a:` Architecture of the new redaction technique. `b:` Average KL-divergence loss vs number of training steps on Medal dataset; Average loss is calculated at every 10 steps. `c:` Measured Renyi divergence for ($\alpha = 2$) vs redaction level for Medal dataset; smarter-redaction is used, see Section 4.5.2.

al., 2017):

$$D_\alpha(P_0||P_1) = \frac{1}{\alpha - 1} \log \int_Y P_0(x)^\alpha P_1(x)^{1-\alpha} dx \tag{4.1}$$

and similarly for $D_\alpha(P_1||P_0)$. When $\alpha = 1$ the Renyi-divergence equals the KL-divergence (Noshad et al., 2017). Divergence can be converted to a $(\epsilon_{safe}, \delta_{safe})$ differential privacy guarantee using concentrated differential privacy. Due to lack of space we do not include the details here, but refer the reader to (Bun & Steinke, 2016; Gusain & Leith, 2024) for complete proofs.

4.3 Overall Architecture

Our method consists of two modules as shown in Figure-4.1a:

`1.)Sentence transformer`. This is a sentence transformer model (Reimers & Gurevych, 2019) that is responsible for generating contextual word-embeddings for an input sentence x_i.

`2.)Ranker Model`. This is a neural network consisting of 4 Linear layers, the first three layers use a tanh activation while the final layer uses a sigmoid activation. This is responsible for ranking the words from an input sentence. It takes the embedding of a word present in the sentence x_i as input and outputs the corresponding ranking for the word.

To redact words from an input sentence, we first generate the embedding of each word in the sentence using the sentence transformer. The word embeddings are then sent to the ranker which generates ranking for each individual word. Words with lowest the K% of the rank are redacted from the input sentence.

4.4 Training Overview

In this section we briefly discuss the training strategy used to train the ranker model.

4.4.1 Training the Ranker

During a training step a batch of sentences db_0 and db_1 is selected from separate held-out training datasets D_0 and D_1. Shorter sentences in a batch are padded with a `[pad]` token to make every sentence have same number of words W[2]. Using a sentence transformer word embeddings e_0 and e_1 for the padded sentences in db_0 and db_1 are generated. The ranker is then used to generate ranking vectors r_0 and r_1 from e_0 and e_1 respectively. The ranking vector for each sentence is updated such that lowest K%[2] of the word rankings are set to zero and the rest are set to one. To make this operation differentiable, we find the K^{th} smallest rank k_r from the ranking vector and subtract k_r from each value in the ranking vector. The resulting vector is then multiplied with a hyper-parameter "T"[2] and passed through a sigmoid function to get an updated rank vector where the lowest K% of the word-ranks are set to zero and the rest are set to one. The updated rank vector is multiplied with word embeddings e_0 and e_1 to get weighted embeddings ue_0 and ue_1. Sentence embeddings are generated from ue_0 and ue_1 by taking the average over the word embeddings of the sentence, which are then used to calculate the KL-divergence loss for the batch (see Section 4.4.2)[3]. An outline of the Algorithm is present in the appendix.

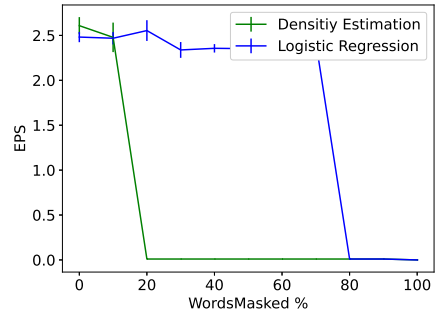
4.4.2 KL-divergence Loss

For natural language datasets the embeddings from a sentence transformer can be used to estimate the probability distributions of two datasets, which can then be used to estimate divergence between the two, see for example (Gusain & Leith, 2024; Pillutla et al., 2021). We used the implementation provided by (Gusain & Leith, 2024), and modified it to create a custom loss function to calculate the KL-divergence between two natural language datasets using PyTorch. To the best of our knowledge there is no open-source implementation in PyTorch to calculate the KL-divergence between two sentence embeddings. The ranker model is trained using this custom loss function.

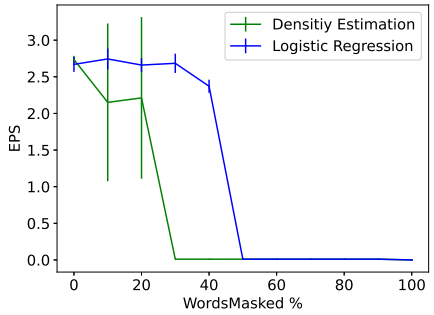
We can expect the ranker to randomly rank words when the training starts thus resulting in higher loss value. But as the training progresses, ranker will be optimized to rank the words such that redacting low K% of the input words will

[2] W is the number of words in the longest sentence from db_0 and db_1. During our training we set K=10 and T=100.

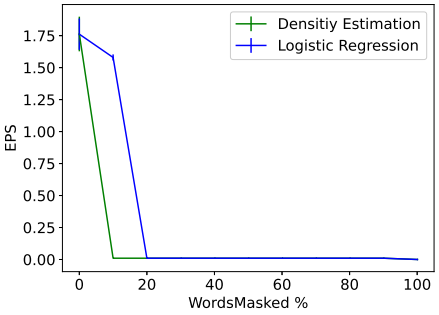
[3] We use KL divergence rather than Renyi divergence because the estimator is differentiable.



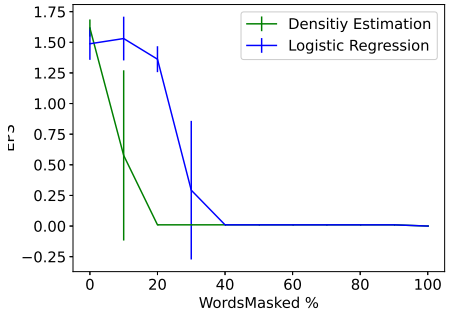
(a) Medal Dataset



(b) Political Dataset



(c) Amazon Dataset



(d) Reddit Dataset

Figure 4.2: Measured ϵ_{safe} between redacted sensitive and safe datasets vs redaction level;Comparison with various redaction strategy.; For our experiment we use $\delta = \frac{1}{n}$ (Gusain & Leith, 2024); n = number of input sentences;

result in lower divergence value between the two datasets. Figure-4.1b shows an example of the average loss vs the number of training steps. We observe that as the training progresses the average loss value is decreased. For more details about the hyper-parameters we refer the reader to an appendix.[4]

4.5 Experiments

4.5.1 Datasets

We compared the performance on four datasets : Medal dataset (Wen et al., 2020), Political dataset (Voigt et al., 2018), Amazon dataset (Gusain & Leith, 2024), Reddit dataset. Sensitive and safe datasets were created from these datasets as explained in (Gusain & Leith, 2024). Each dataset contains almost 10000 sentences each in the safe and sensitive dataset. The validation set for medal reddit and amazon dataset contained 2000 sentences each in the safe and sensitive dataset. For more details regarding the datasets we refer reader to Chapter-2.

[4] https://github.com/vaibhav0195/density_estimation_code

4.5.2 Redaction

Redaction was carried out on a separate validation set as the percentage of words redacted is varied. A fine-tuned sentence transformer was used to generate embeddings for the redacted sentences. The embeddings were used to estimate the Renyi-divergence. Renyi-divergence was estimated using the estimator provided in (Gusain & Leith, 2024), which was then converted to an (ϵ, δ) differential privacy guarantee using concentrated differential privacy.

We compared our redaction approach against the smarter-redaction[5] approach introduced in (Gusain & Leith, 2024). To redact words using smarter redaction, we trained a logistic regression model on TFIDF features of the training data. The trained model's weights were used to rank the words present in the sentence. Top K percent of these words were then redacted. Note that we did not compare against random redaction as it does not provide any benefits over smarter redaction (Gusain & Leith, 2024).

4.5.3 Results

Figure-4.2 illustrates the privacy benefits of our approach compared to the smarter redaction approach introduced in (Gusain & Leith, 2024). Sentences ranked by our ranker achieve lower ϵ_{safe} values for the same redaction percentage. We get (ϵ_{safe}) values which are close to zero by only redacting 20-30% of words, whereas the approach in (Gusain & Leith, 2024) needs to redact almost 80% of the input words to get similar privacy benefits.

We observe that our new ranker redacts words so as to quickly remove sensitive information early from the text which results in lower ϵ_{safe} at lower redaction levels. E.g. consider the input sentence from Medal dataset `organ failure tone and ventral pallidum cell injury`, the important words to be redacted are "tone" and "ventral" cell as they reveal more about the type of injury. The logistic regression redacts the words "cell" and "failure", whereas our new ranker redacts "tone" and "ventral".

4.6 Conclusion

In this chapter we propose a new loss function which can be used to train a neural network to redact words efficiently from sensitive text to gain privacy benefits.

In addition to the neural network presented here we experimented with various other redaction approaches as well:- 1.) Redacting words using KNN- redacting words from the input sentence such that clusters between the two distribution $\mathcal{D}_0$ and

[5]redaction carried out using logistic regression in Chapter-2

$\mathcal{D}_1$ overlap quicker thus resulting in lower divergence values and 2.) Reinforcement learning- training a transformer model to generate domain and adaptive masking using reinforcement learning as explained in (Kang et al., 2020), which can then be used to mask the words from the input sentence. We observe no significant improvement over the logistic regression approach introduced in Chapter-2.

Chapter 5

Do LLMs Learn Graph Representations Without Context?

The preceding chapters focused on the development and evaluation of robust redaction methodologies for enhancing privacy protection in natural language text. While these approaches mitigate privacy risks arising from information present in the training data, an important complementary question is whether large language models (LLMs) may encode information in more implicit forms. In particular, if models can develop internal representations solely from training data, this may constitute a potential avenue for privacy leakage that is not directly addressed by text-level sanitisation.

This chapter investigates this question by examining whether LLMs develop internal representations of underlying graph structures from training data. This line of inquiry is closely related to the world-representation hypothesis, which suggests that LLMs may develop internal representations of aspects of the world through exposure to large-scale training data (Karvonen, 2024; K. Li et al., 2024). If such representations exist, they could potentially encode sensitive information, thereby raising additional privacy concerns beyond those addressed through redaction.

To explore this issue, we employ graph-based experimental tasks designed to probe model behaviour under both in-context and zero-shot conditions. Our results indicate that, in zero-shot scenarios, LLMs do not form internal graph-like representations; instead, their behaviour is predominantly guided by contextual information present in the input. These findings suggest that the models apparent capabilities arise primarily from context utilisation rather than from internalised representations learned during training, thereby mitigating certain classes of privacy risk associated with latent structural memorisation.

5.1 Introduction

Large language models are neural networks that are based on the transformer architecture, these models are trained on the input sentences on a simple "next-word-prediction" task [1](T. B. Brown et al., 2020; Devlin et al., 2019; Vaswani et al., 2023). The models take user queries as input and generate predictions in response.

Two major techniques are used when generating predictions from these models: (1) In-context learning, where few-shot examples are provided alongside the query to guide the model toward better predictions, and (2) Zero shot learning, where only the input query is provided to the model(T. B. Brown et al., 2020; Gozeten et al., 2025; Han et al., 2023; Min et al., 2022; Weber et al., 2023; Yang et al., 2024).

Despite being trained on such a simple task, these models demonstrate remarkable capabilities in understanding context from natural language text data. These are models are used for a plethora of tasks, including solving logic puzzles, writing and debugging computer programs, and answering general user queries (Abdou et al., 2021; B. Z. Li et al., 2021).

Generally in-context learning approaches have proven to provide better predictive capabilities than the zero shot learning approaches (Agarwal et al., 2024; T. B. Brown et al., 2020; Wei et al., 2023). However, how these capabilities emerge in these models while being trained on a simple next word prediction task, remains unclear. There are two main theories that attempt to explain working of these models : 1) These models only understand correlation of words in the sentences used in the training data, thus only learning the surface level statistics (Bender et al., 2021). 2) These models do more than learn the surface level statistics and develop an internal representations for very simple concepts, such as color, direction, game state etc (Karvonen, 2024; K. Li et al., 2024; Vafa et al., 2024).

The world representation theory has primarily been explored through in-context learning models, where context such as partial game states is provided alongside the input (Karvonen, 2024; K. Li et al., 2024). In this scenario, it is difficult to determine whether learned representations emerge from the provided context or from the underlying training data.

In this chapter we investigate whether large language models develop internal representations of simple graph structures during zero shot learning. We study this problem by training models using the two approaches described above on the sequences generated from a graph. Then employ various probing techniques to determine whether the models have constructed internal graph representations from

[1]Modern LLMs may also be trained with reinforcement learning techniques, but this aspect is beyond the scope of this study. LLMs trained solely on next-word prediction also exhibit capabilities such as solving logic puzzles, reasoning, (Agarwal et al., 2024; T. B. Brown et al., 2020; Wei et al., 2023)

the training data.

To assess whether models have learned internal representations, we use the model's predictive capabilities to generate the adjacency matrix of the underlying graph, which we then compare it to the original adjacency matrix of the graph to calculate the reconstruction errors by the model. We also use linear probes to understand whether the activations of the trained model contained representation of the underlying graph.

Reconstruction errors and probing accuracy provide complementary perspectives on understanding the representation learned by the model. Reconstruction evaluates whether the model's outputs can recover the adjacency matrix, while probing tests whether hidden activations encode edge information independent of the output.

We observe that models trained using in-context learning produce fewer errors and are able to recover the internal structure of the graph, compared to the model trained using the zero shot learning approach. We find no evidence that the zero shot learning models learn internal representation of the underlying graph structure from training data. These findings are interesting and timely since they demonstrate that these models are only able to reconstruct the underlying structure when it is explicitly provided in the input.

5.2 Related Work

Large Language Models : Large Language Models (LLms) are non-linear machine learning models that are built on the transformer architecture (Vaswani et al., 2023). These are trained on a huge corpora of dataset and have demonstrated remarkable capabilities to perform various tasks such as question answering, summarization, puzzle solving etc (T. B. Brown et al., 2020; Devlin et al., 2019). Their ability to generalize from patterns in data has made them a focal point of research in natural language processing and machine learning.

World-Representation in LLMs: Despite their success, LLMs are largely blackboxes and understanding their internal mechanisms remains a key challenge. One prominent research direction is the study of world-representations-the extent to which models learn internal representations of structured environments from the training data. In such studies, LLMs are trained on sequences generated from controlled environments, such as game boards, and researchers analyze the learned representations. Notable examples include investigations into games like Othello, Chess, and simplified spatial reasoning tasks (Karvonen, 2024; K. Li et al., 2024; Vafa et al., 2024). These studies explore whether LLMs encode underlying rules, states, or other abstract features of the environment.

In-Context Learning and Zero Shot Learning : Predictions from LLMs

are typically generated using either in-context learning or zero shot learning approaches.

`1) In context learning :` In this setting, the model receives both the query and a few input examples. These examples guide the model toward more accurate predictions and reduce errors (T. B. Brown et al., 2020; Han et al., 2023; Min et al., 2022; Weber et al., 2023; Yang et al., 2024). In-context learning is often used to study internal representations: for instance, providing partial game states along with next moves allows researchers to analyze attention patterns and reconstruct the game-state representations learned by the model (Karvonen, 2024; K. Li et al., 2024).

`2) Zero Shot Learning:` Here, the model is trained on input sequences, but during prediction, no additional examples are provided. While simpler to deploy, this approach generally yields lower predictive accuracy than in-context learning and provides fewer cues for extracting internal representations (Gozeten et al., 2025).

Probing : Probing is a standard methodology to investigate whether models encode specific feature or concepts in its activations. A probe is typically a classifier or regressor that takes the activations of a trained model as input and predicts a feature of interest, such as part-of-speech tags, syntactic structure, or game state (Alain & Bengio, 2016; Belinkov, 2021; Krause et al., 2020). Probing has been widely used to explore both linguistic knowledge in LLMs and abstract representations in structured environments, providing insight into what information is encoded and where it resides within the network.

5.3 Preliminaries

5.3.1 Generating Training and Validation data from the Graph

A non-weighted graph is defined as $\mathcal{G} = (V, E)$ where V are the vertices of the graph and E represents the edges between the vertices. To generate training data, we generate random paths from the graph. Each sequence starts with the S (the start node of the path), D end node of the path and L length of the path, followed by the corresponding path. The validation dataset is generated by sampling sub-paths from the sequences while ensuring that no (S, D) pair appears in both training and validation datasets.

To encourage the model to learn when paths do not exist, we also include (S, D) pairs with no valid path of length L in both training and validation datasets. In such cases, instead of providing a path, we insert a special `[NP]` token, which signals the absence of a valid connection between the nodes. For in-context learning, the model input additionally includes a subgraph relevant to the current path. Full details on

query generation are provided in the Appendix.

In this setup, the tuple (S, D, L) serves as the query, and the corresponding path or the special `[NP]` token when no such path exists serves as a response to the query.

5.3.2 Computing Errors During Reconstruction

Our models are trained to predict paths between node pairs in the graph. The predictions can be used to reconstruct an adjacency matrix, which serves as a proxy for the internal representation of the graph learned by the model.

For a given path length L, we sample N (S, D) pairs not present in the training dataset[2] and generate predictions for each pair. These predictions are used to construct the adjacency matrix. The reconstructed matrix is then compared with the original adjacency matrix to quantify errors:

gt_e: edges present in the original graph but absent in the reconstruction (indicating gaps or biases).

pr_e: edges present in the reconstruction but absent in the original graph (indicating hallucinations).

Ideally, both gt_e and pr_e should be close to zero, indicating accurate reconstruction without bias or hallucination.

A direct comparison against the full original adjacency matrix may overestimate errors, because sampled paths do not necessarily cover every edge of the graph. To address this, we construct a reduced "reference" adjacency matrix that contains only the edges present in the training data. The error metrics are then computed relative to this reduced matrix. This adjustment ensures that the evaluation measures how well the model captures the graph structure that was actually presented during training, rather than penalizing it for missing edges it had no opportunity to learn.

5.3.3 Probing Internal State

Probing allows us to determine whether specific structural information is encoded in the intermediate layers of a trained model, independently of its output predictions.

We focus on the existence of an edge between two nodes. Given a pair of nodes (S, D), the probe is trained to predict whether an edge exists between S and E, i.e., whether $(S, D) \in \{edges\}$. The ability of a probe to recover this information from hidden activations indicates that the model has encoded neighborhood structure.

A linear probe can only succeed if the relevant information is linearly separable in the model's activation space. This ensures that high probe accuracy reflects information already present in the representations, rather than capacity of the probe itself.

[2]Details on N are provided in the Appendix.

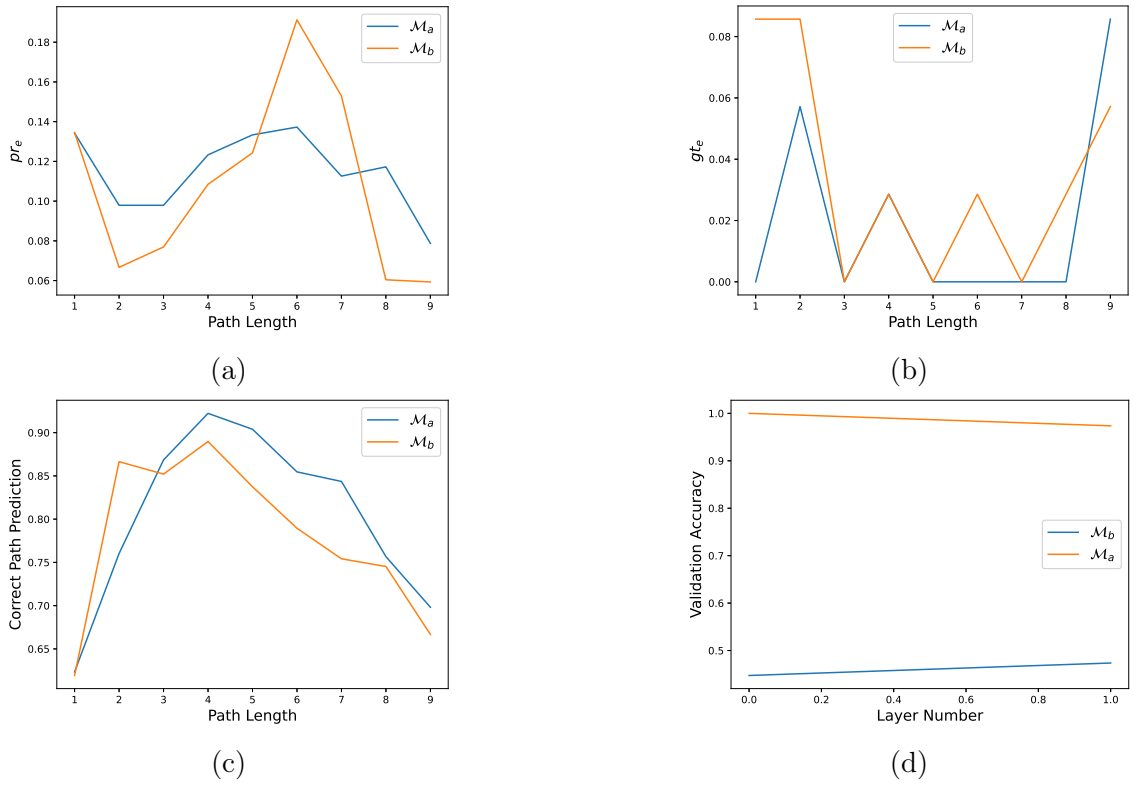


Figure 5.1: Measure pr_e and gt_e for the toy graph dataset v pathlength; 5.1c shows the prediction accuracy of the two models vs the path length in the graph; 5.1d shows the probe accuracy of the two models vs the layer number in the graph;

We construct a balanced dataset of positive pairs ($(S, D) \in \{edges\}$) and negative pairs ($(S, D) \notin \{edges\}$), sampled from graphs disjoint from the training set. The dataset is split into training and validation sets in an $80 : 20$ ratio.

For each pair, hidden activations are extracted from every attention layer. A separate linear probe is trained for each layer to classify edge presence, providing a layer-wise measure of how neighborhood information is represented.

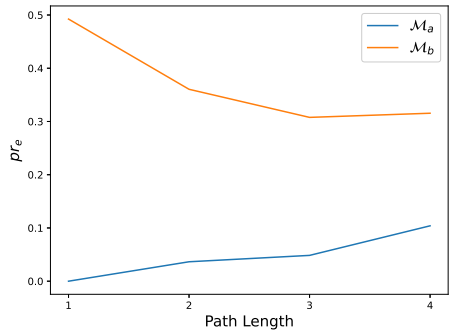
5.4 Performance of In-context Learning vs Zero shot learning

To initially study the performance of in-context Learning and the zero shot learning model we generate 5 small graphs using networkX [3], containing 10 nodes and 20 edges. The training and validation data is generated from each graph dataset[4]. For more details about the training data we refer the reader to Section-5.3.1.

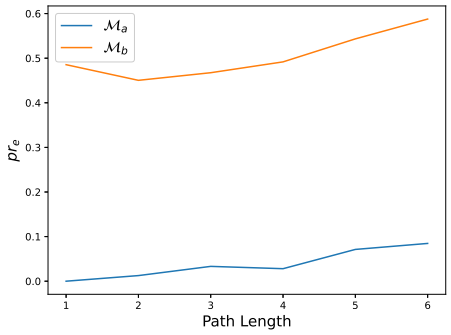
We evaluate the following models :

[3] https://networkx.org/

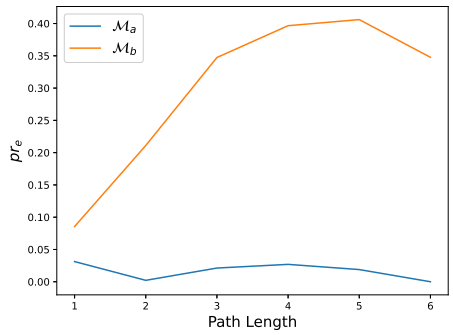
[4] The training and validation data contains paths from all the graphs.



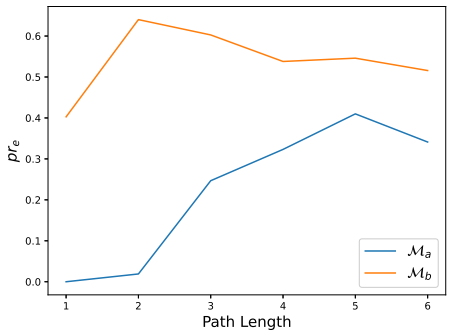
(a) Dublin dataset.



(b) Cora dataset



(c) Amazon dataset



(d) Facebook dataset

Figure 5.2: Measured pr_e Various datasets.

`1) In-context learning` ($\mathcal{M}_a$) : The model receives the relevant graph as context and is queried to predict a path between a pair of nodes. For training, each input sequence also includes the full graph to guide the model's predictions.

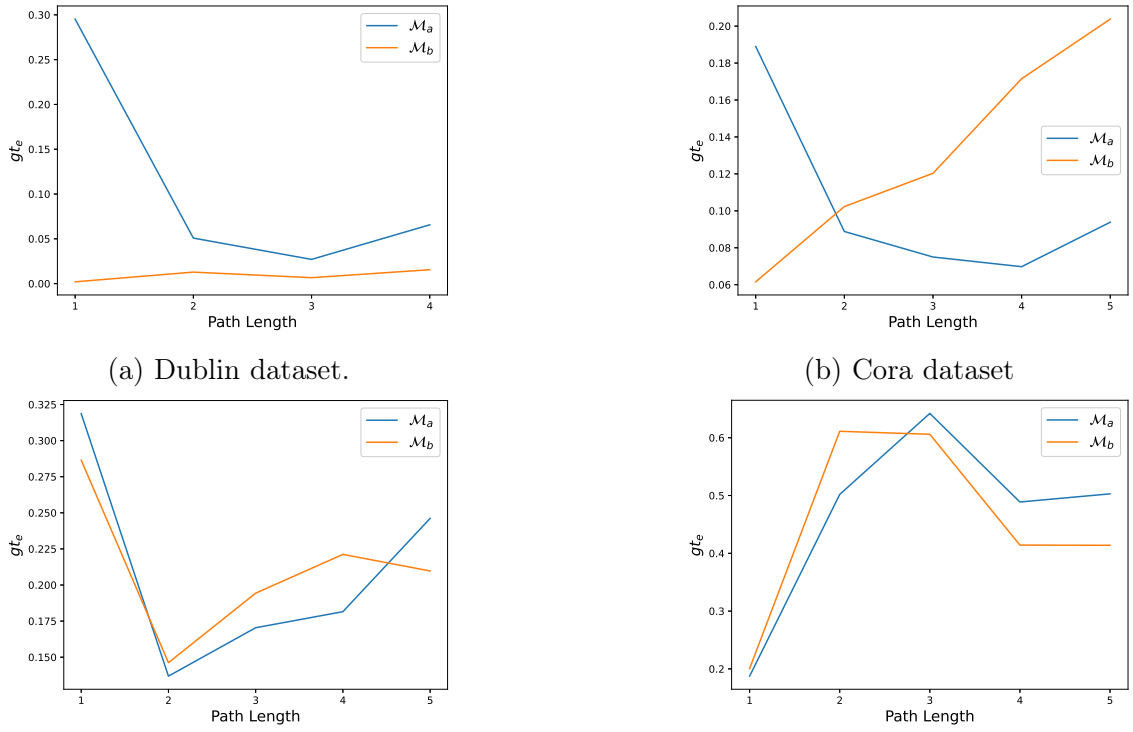
`2) Zero Shot Learning` ($\mathcal{M}_b$) : The model receives only the pair of query nodes and predict a path in the graph without any context. This model is trained specifically on a single graph.

Both $\mathcal{M}_a$ and $\mathcal{M}_b$ are GPT-style transformer models with 2 hidden layers, 8 attention heads, and an embedding size of 128. The models are trained from scratch with randomly initialized weights in an auto-regressive fashion for next-word prediction, ensuring no prior knowledge of the graph structure.

We evaluate the models by computing pr_e and gt_e errors as described previously. These metrics quantify hallucinated edges and missing edges in the reconstructed adjacency matrices, respectively.

Figures 5.1a and 5.1b present the reconstruction errors of both models on the graphs. The results indicate that the two models yield comparable reconstruction errors as well as similar path prediction accuracy.

Figure 5.1c illustrates the accuracy of predicting the correct path given the query (S, D, L) as the path length increases. We observe that $\mathcal{M}_a$ achieves slightly higher accuracy than $\mathcal{M}_b$ for longer paths, while their overall performance remains largely comparable. This pattern aligns with the reconstruction errors, where both models



(a) Dublin dataset.

(b) Cora dataset

(c) Amazon dataset

(d) Facebook dataset

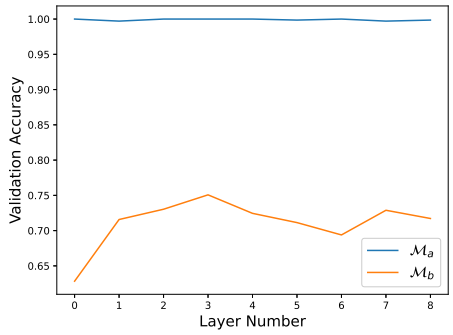
Figure 5.3: Measured gt_e Various datasets.

exhibit a similar level of error.

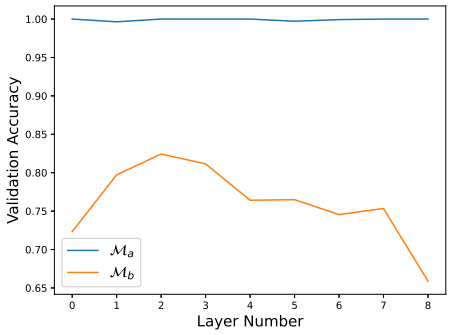
We also train linear probes for the models for various attention layers. The training and validation data for probes is generated from the graph as described previously. Figure-5.1d shows the linear probe accuracy v the path length. We observe the probes for model $\mathcal{M}_a$ consistently outperforms the probes for model $\mathcal{M}_b$.

The high performance of the probe on $\mathcal{M}_a$ indicates that the activations of the in-context learning model encode structural information about the graph - specifically, whether two nodes are neighbors - whereas the activations of $\mathcal{M}_b$ do not contain this information.

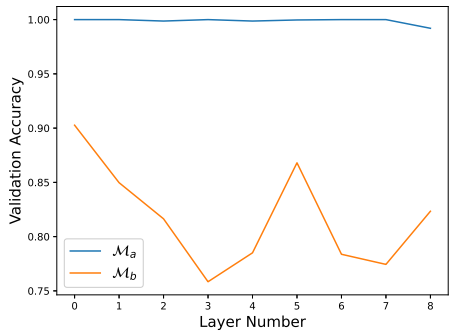
The results for model $\mathcal{M}_a$ are unsurprising since the relevant graph to the query is provided as input to model, hence it can be reconstructed by a probe at the initial layers. Interestingly the activations of model $\mathcal{M}_b$ fail to support a reliable reconstruction of the graph structure.



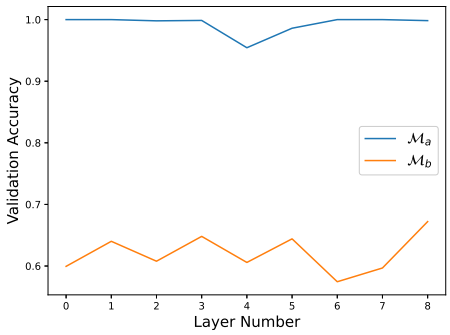
(a) Dublin dataset.



(b) Cora dataset



(c) Amazon dataset



(d) Facebook dataset

Figure 5.4: Measured Classification Accuracy of linear probe for Various datasets. x axis represents the layer from which the activations were chosen, the y axis represents the accuracy of the linear probe on validation dataset.

5.5 Prediction performance: single vs multiple models

On the synthetic graph, we observed that the in-context learning model $\mathcal{M}_a$ produced slightly fewer errors and achieved higher path-prediction accuracy than the zero-shot learning model $\mathcal{M}_b$, although the overall performance of the two models was largely comparable. In that setting, the full graph was provided as context to $\mathcal{M}_a$, making the tasks of reconstruction and path prediction relatively straightforward.

In contrast, real-world graphs are substantially larger and more complex, making it impractical to supply the entire graph as input context. Instead, only a small, relevant subgraph is provided. This change in input fundamentally alters the difficulty of the task: while in Section 5.4 both models performed similarly when $\mathcal{M}_a$ had access to the full graph, in real-world scenarios the length of the subgraph context plays a decisive role. In this section, we evaluate the models on real-world datasets, where only a subgraph relevant to the query is provided as input context. This setup allows us to examine how limiting the context to local subgraphs, rather than the full graph, affects the in-context learning process.

Datasets

We train and test our models on the following datasets :

`CORA Dataset :` Cora Citation Network is a directed graph where nodes are scientific papers and edges represent citations (i.e., paper A $\rightarrow$ paper B means A cites B). For our experiments we sample 500 nodes from the CORA dataset. We generated 60000 training sequences from the graph.

`Dublin Street Map :` A real-world directed road network of Dublin, Ireland. We use the Open-street Map API to extract the graph. For our experiments we sample 500 nodes from the original map dataset. We generated 60000 training sequences from the graph.

`Facebook Social Circle Dataset :` A real-world directed graph that dataset consists of 'circles' (or 'friends lists') from Facebook. Each node in a graph denotes an individual and an edge between the individual represent a connection on facebook between the nodes. For our experiments we sample 500 nodes from the CORA dataset. We generated 60000 training sequences from the graph.

`Amazon Co-Purchase Network :` A graph crawled from amazon website. It is based on Customers Who Bought This Item Also Bought feature of the Amazon website. If a product i is frequently co-purchased with product j, the graph contains an edge from i to j. Each product category provided by Amazon defines each ground-truth community. For our experiments we sample 500 nodes from the CORA dataset. We generated 60000 training sequences from the graph.

Training sequences are constructed as described in Section 5.3.1, capturing paths between node pairs. In this setting, the context provided to $\mathcal{M}_a$ consists solely of a relevant subgraph extracted around the query nodes. For more details on how the subgraph is generated we refer the reader to the appendix.

Experiments

Models $\mathcal{M}_a$ and $\mathcal{M}_b$ are trained on each dataset using the approach described in the previous section. Both models have 10 hidden layers, 10 attention heads, and an embedding size of 500 to accommodate the increased graph complexity.

The adjacency matrix is generated from the predictions of sampled (S,D) node pairs as explained previously, that is then used to calculate the errors generated by the models.

Figures 5.2 and 5.3 show the errors for both models across datasets for various path-lengths. Model $\mathcal{M}_a$ consistently exhibits lower error rates than $\mathcal{M}_b$. This improvement can be attributed to the explicit subgraph context provided to $\mathcal{M}_a$, which allows it to better capture underlying graph structure and reduce hallucination. These results demonstrate that in-context learning produces fewer error than zero shot learning when applied to real-world graphs.

In the previous setting, where the full graph was available as context both mod-

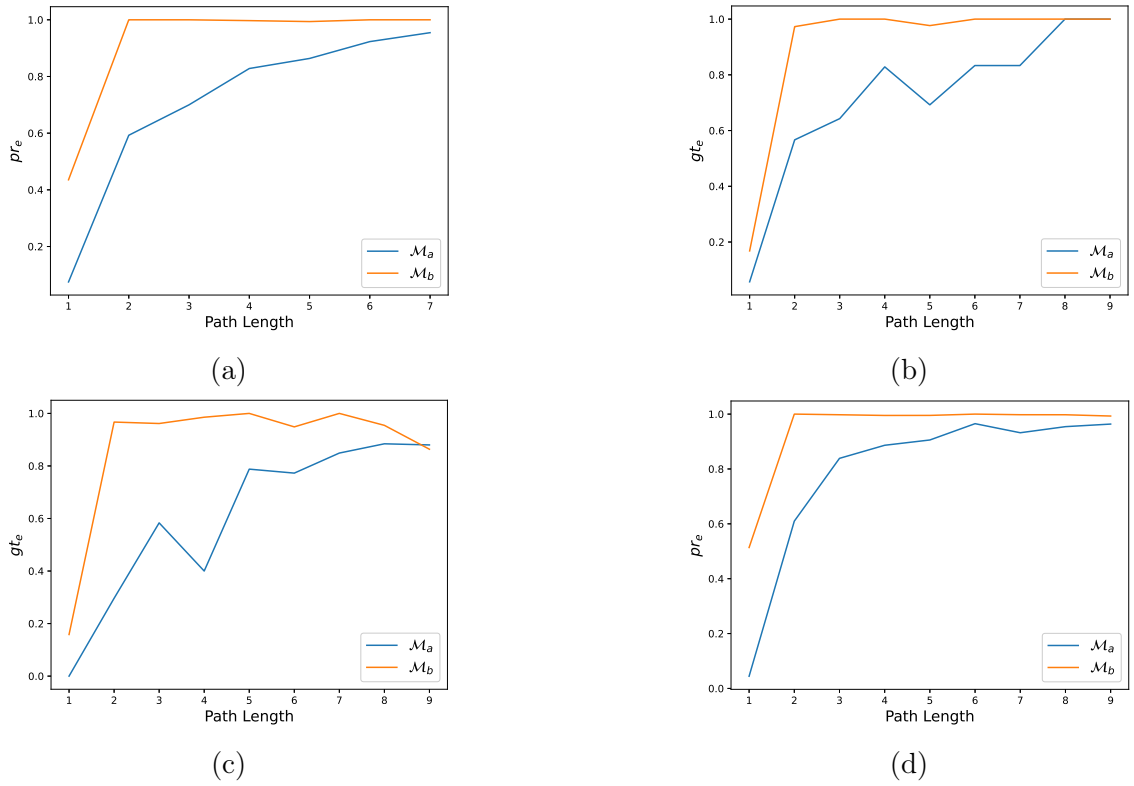


Figure 5.5: Measure pr_e and gt_e on various other datasets for the model trained on cora dataset. 5.5a and 5.5b shows the pr_e and gt_e error for dublin dataset; 5.5c and 5.5d shows the pr_e and gt_e error for amazon dataset;

els performed similarly. However, when only relevant subgraphs are provided, the performance of $\mathcal{M}_a$ improves relative to $\mathcal{M}_b$. This suggests that shorter, localized contexts make in-context learning more effective when holding the model complexity constant, whereas zero-shot learning fails to take advantage of the available structural information.

Probing Internal State

We train linear probes for the models for all the attention layers. The training and validation data for probes is generated from the graph as described previously.

Figure 5.4 shows the validation accuracy of the linear probe for both models as a function of attention layer. Across all layers, the probe trained on model $\mathcal{M}_a$ consistently outperforms the probe trained on model $\mathcal{M}_b$. We expected the earlier layers of $\mathcal{M}_a$ to achieve good probe accuracy, since the input graph information is explicitly provided as context. However, we also observe that this signal propagates through later layers.

5.6 No evidence for memorization by in-context learning

Our previous experiment shows that model $\mathcal{M}_a$ outperforms model $\mathcal{M}_b$ across various datasets when only a relevant subgraph is provided. The huge performance improvement is due to the extra context provided to the model during training and inference. We also wanted to observe whether $\mathcal{M}_a$ is relying on memorizing its training data or actually learning graph processing patterns that can generalize. To test this, we evaluate the models on paths from a completely different graph: a model trained on sequences from graph $\mathcal{G}_s$ is now tested on a separate graph $\mathcal{G}_d$.

We generate $\mathcal{G}_d$ from a new graph dataset and select a subgraph from $\mathcal{G}_d$ such that the number of nodes are similar to that of $\mathcal{G}_s$. We sort the nodes in both graphs by their total degree and then rename the nodes in $\mathcal{G}_d$ to match the degrees in $\mathcal{G}_s$. This way, the graphs are similar in structure but do not share the same nodes or edges. If $\mathcal{M}_a$ had memorized the training data, we would expect its performance to drop sharply on this new graph.

To reconstruct the adjacency matrix, we again sample N number of (S, E) nodes from the new graph dataset $\mathcal{G}_d$. We then compare it to the true adjacency matrix of $\mathcal{G}_d$ to calculate the prediction errors (pr_e and gt_e).

Figure 5.5 shows the errors for both models when trained on the CORA dataset and tested on the Dublin street map and Amazon co-purchase datasets. We see that $\mathcal{M}_a$ produces slightly fewer errors than $\mathcal{M}_b$ for shorter paths, but as the path length grows, the errors of both models become similar. We observe similar results on other dataset pairs, which are provided in the appendix. Fewer errors for $\mathcal{M}_a$ at short path lengths suggests that it is not simply memorizing the training data, instead it is using the context to generate predictions.

The results suggests that the improved performance of in-context learning comes from using the provided subgraph during prediction, rather than from memorizing training data. However, when tested on completely new graphs, performance decreases, suggesting that the model's ability to generalize depends on patterns seen in the training graphs and does not fully transfer to graphs with different structures.

5.7 Scaling of in-context learning and Zero shot Learning Models

Scaling is a crucial consideration for evaluating whether the observed behaviors of LLMs persist as input context or graph size increases. In this section, we study : (1) the effect of enlarging the subgraph provided as context for in-context learning

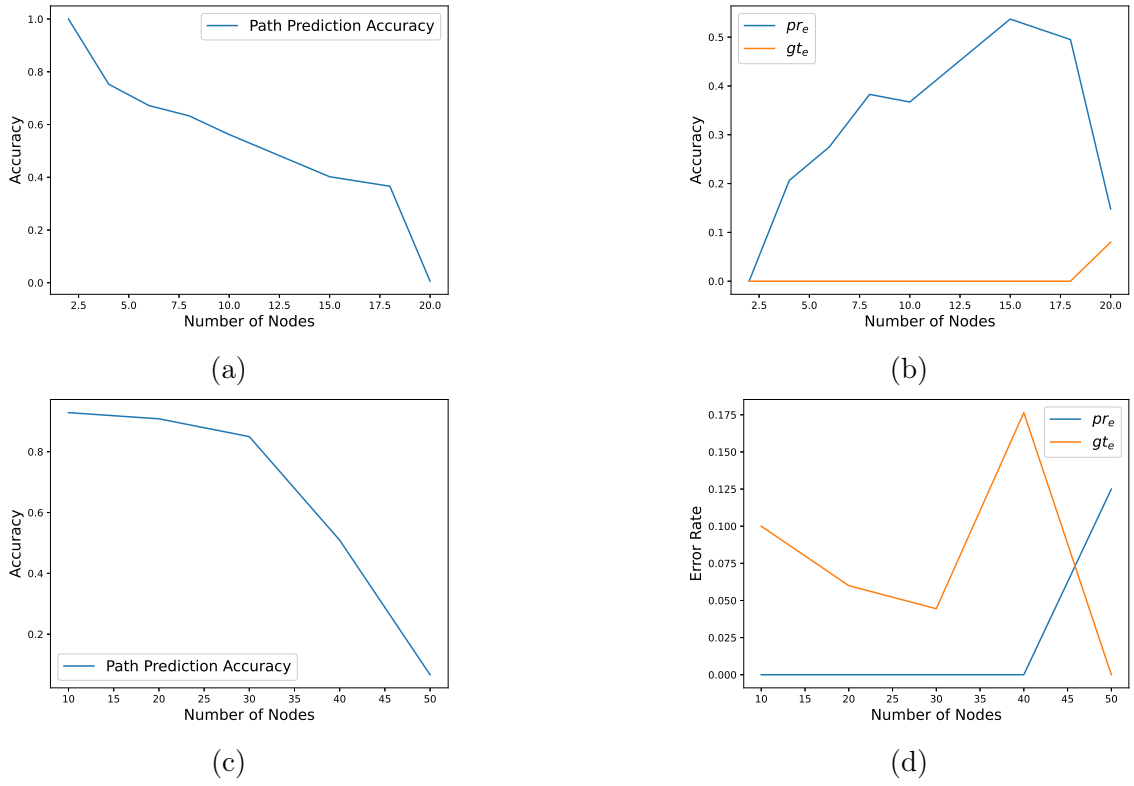


Figure 5.6: 5.6a shows the prediction accuracy of the model $\mathcal{M}_a$ as the context is varied; 5.6b shows the error in prediction of the model $\mathcal{M}_a$ as the context is varied; 5.6c shows the prediction accuracy of the model $\mathcal{M}_b$ as the graph size is varied; 5.6d shows the error in prediction of the model $\mathcal{M}_b$ as the graph size is varied;

models ($\mathcal{M}_a$), and (2) the effect of increasing the size of the training graph for zero shot learning models ($\mathcal{M}_b$).

Both models are trained on synthetic graph datasets generated with networkX, where we vary the number of nodes in each graph. For every dataset with a fixed node count, models $\mathcal{M}_a$ and $\mathcal{M}_b$ are trained. The training and validation sets are constructed using the procedure outlined earlier. For model $\mathcal{M}_a$, we adjust the number of nodes included in the input context, whereas for model $\mathcal{M}_b$, we adjust the number of nodes in the underlying graph structure.

The models are GPT style models with 2 hidden layers, 8 attention heads and an embedding size of 128 [5]. The training is carried out in an auto-regressive fashion as explained previously.

Scaling of Context in In-Context Learning ($\mathcal{M}_a$):

Figure 5.6a illustrates path-prediction accuracy as the size of the subgraph provided in the context increases, while Figure 5.6b shows the corresponding reconstruction errors. As the number of nodes in the context grows, path-prediction accuracy decreases and the proportion of hallucinated edges (pr_e) increases. It can

[5]The trained model had 5.3M parameter

be seen that the accuracy of in-context learning model predictions degrades as the context size increases.

Scaling of Graph Size in Zero shot learning ($\mathcal{M}_b$):

We also examined the effect of increasing the size of an almost linear underlying training graph. Figures 5.6c and 5.6d show that as the graph size increases, prediction accuracy declines and reconstruction error grows. However, compared to $\mathcal{M}_a$, $\mathcal{M}_b$ retains relatively high accuracy even for larger graphs, indicating that zero-shot learning scales more gracefully with graph size.

The two models exhibit distinct behaviors as the number of nodes increases. In-context learning models ($\mathcal{M}_a$) show a decline in path-prediction accuracy and an increase in hallucinated edges as the context size grows, whereas zero-shot learning models ($\mathcal{M}_b$) show decreasing prediction accuracy and increasing reconstruction error as the training graph size increases.

These findings highlight that both approaches have inherent scaling limitations. For in-context learning, performance is sensitive to the size of the input context, while for zero-shot learning, performance is constrained by the size of the training graph.

5.8 Conclusion

Our findings indicate that current LLMs do not develop robust internal representations of underlying graph structures from training data alone. In zero-shot settings, these models produce high reconstruction errors and show no evidence of internal structure learning. While providing contextual information improves reconstruction and probing accuracy, this demonstrates that the models primarily rely on the input context rather than forming generalizable, internal world models during training.

We argue that if these models truly learned internal representations from training data, they should demonstrate lower reconstruction and probing errors even in zero shot learning settings. Our findings suggest that current LLMs primarily rely on explicit contextual information rather than developing robust internal world models during training.

5.9 Future Work

In this chapter, we find that LLMs exhibit robust internal representations of underlying graph structure only when explicit contextual information about the graph is provided at inference time. Graph-structured data was chosen for our experiments because it provides a well-defined and interpretable way to encode relational information, enabling a direct assessment of whether such structure is captured in

a model's internal representations. While other data modalities, such as audio or video, may also contain latent structural relationships, the methods and insights developed in this chapter are specific to graph-based representations. Applying this analysis to non-graph modalities would require developing new evaluation and reconstruction metrics. We therefore leave the investigation of such modalities to future work

Chapter 6

Conclusions

This thesis addresses critical privacy challenges in natural language text processing through two complementary research directions. The first part of this work (Chapters 2-4) establishes redaction as a principled and effective approach to text privacy. Through the development of a novel framework using Renyi-divergence to quantify privacy gains, we demonstrated that sufficient redaction can render sensitive text statistically indistinguishable from non-sensitive text, providing formal privacy guarantees. Our comparative analysis revealed that redaction-based approaches achieve substantially better utility-privacy trade-offs than traditional differential privacy methods like DP-SGD, often providing 100× improved utility. The introduction of an advanced redaction methodology using neural network rankers with KL-divergence loss further enhanced these benefits, achieving strong privacy guarantees ($\epsilon \approx 0.01$) with only 20-30% word redaction compared to the 80% required by previous methods.

The second part of this thesis (Chapter 5) provides insights into the representational capabilities of large language models, directly informing privacy risk assessment. Through systematic investigation of whether LLMs develop internal graph representations in zero-shot settings, we found no evidence that these models form robust internal world models from training data alone. Instead, LLMs demonstrate strong dependence on explicit contextual information provided during inference, with minimal ability to reconstruct structural relationships without such context. This finding suggests that the privacy risks associated with internal representations may be more limited than previously assumed, as these models appear to rely primarily on the input context provided and surface-level pattern matching rather than deep structural understanding of training data.

Chapter 7

Appendix

7.1 Appendix 1

7.1.1 Hyperparameters for next word prediction

To train the Next-word-prediction model negative log-likelihood loss was used. The model's weights were initialized with a value drawn from a uniform distribution (-0.1,0.1). A stochastic gradient descent algorithm was used. Each model was trained for at most 40 epochs, with an initial learning rate of 20. the learning rate was annealed by a factor of 4 if the validation loss between two epochs does not decrease.

7.1.2 Utility

To further analyze privacy V utility trade-off we performed similar experiments as suggested in (Mattern et al., 2022). They show that adding a large amount of noise using word-level-dp approaches hurt the utility of the model. The idea behind these experiments is to sanitize the input data and calculate the attack accuracy and a utility such as sentiment analysis on the sanitized input data. Using the same idea we illustrate that even when 50% of the input data is redacted the santised data can still be used for other end tasks such as sentiment analysis.

We used the Amazon reviews dataset based on two categories (kitchen appliances and drug stores). The attack aims to determine the reviews category, while the

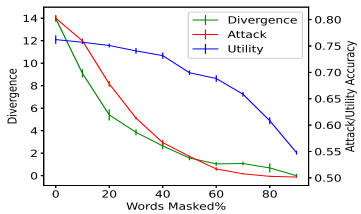
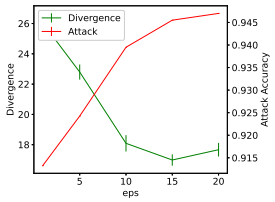
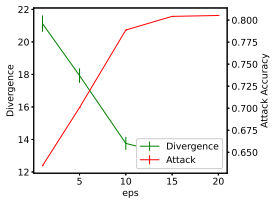


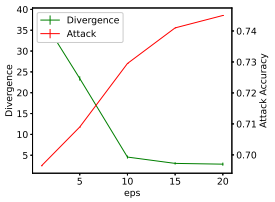
Figure 7.1: Divergence/Utility vs Redaction level for Amazon Review dataset



(a) Political Dataset

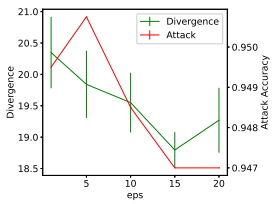


(b) Amazon Dataset

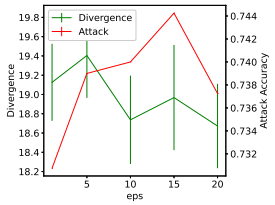


(c) Reddit dataset

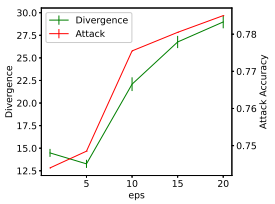
Figure 7.2: Comparison of divergence and attack accuracy for various datasets using CUSTEXT approach.



(a) Political Dataset

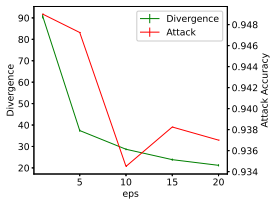


(b) Amazon Dataset

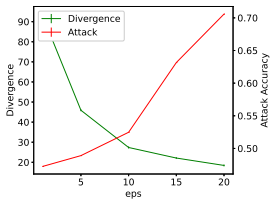


(c) Reddit dataset

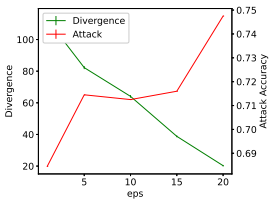
Figure 7.3: Comparison of divergence and attack accuracy for various datasets using SANTEXT approach.



(a) Political Dataset



(b) Amazon Dataset



(c) Reddit dataset

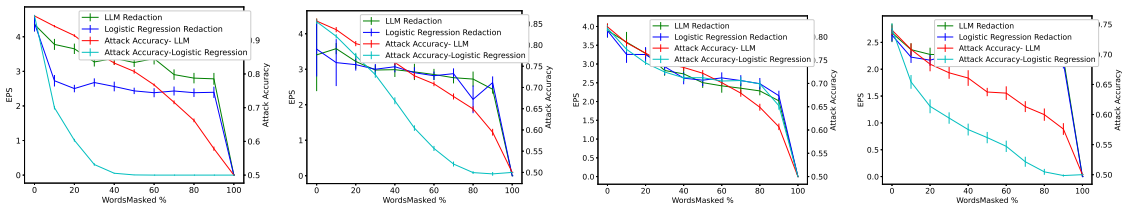
Figure 7.4: Comparison of divergence and attack accuracy for various datasets using AMAZON WORD DP appraoch

utility indicates the accuracy by which we can still categorize the correct sentiment from the reviews. We used smarter redaction as explained previously, and used a similar attack as explained in the study.

Figure-7.1, shows the graph for divergence v utility and attack accuracy for the Amazon review dataset. We can see that at 50% redaction, the attack accuracy between the dataset is almost 50% and the divergence has also decreased a lot, while the utility is still preserved.

7.1.3 Comparison Against SOTA

Figure-7.2, Figure-7.4, Figure-7.3, shows the divergence between the sensitive and safe dataset as the ϵ parameter is varied. We can observe that even when a high



(a) Medal dataset (b) Political Dataset (c) Amazon Dataset (d) Reddit dataset

Figure 7.5: Comparison of ϵ and attack accuracy for various datasets using the smarter redaction approach and Chat-gpt-3

amount of noise is added attack accuracy and the divergence for all the dataset remains relatively high. Comparing these plots to Figure - 4 in the paper, we can observe that a lower divergence and a lower accuracy score can be acheived fairly early by redacting only fewer words.

7.1.4 Redaction carried using Chat-GPT

Figure-7.5 shows the comparison of ϵ and attack accuracy for the datasets redacted using the smarter-redaction approach and chat-gpt4o. To redact words using chat-gpt, we send the input sentences from both the classes along with the system prompt *You are an NLP expert. Given the following sentences from two different classes, identify the words that should be redacted to make classification difficult. Return only the words that should be redacted for each sentence, separated by commas.*

We observe that we get similar attack accuracy and privacy benefits from the sentences redacted using chat-gpt and the logistic regression model.

7.1.5 Pseudo-code for Renyi-Divergence Estimator

Algorithm-2 shows the pseudo code for the renyi-divergence estimator used.

7.1.6 Redacted sentences

Table 7.1 presents several original sentences from the Medal dataset alongside their redacted versions. For readability, we display only the first 50 words of each example, as the full sentences are considerably longer.

Algorithm 2 Estimate Rényi divergence $D_\alpha(P_0 \| P_1)$. X and Y denotes the sentence embedding vectors generated from sentence transformer for the sentences in the senstive and safe datasets. are embedded vectors from the sensitive and safe datasets. k is the number of neighbours to consider and α is the Rényi order.

function : ESTIMATE_RENYI$(X, Y, \alpha, k, rounding)$
  $d \leftarrow$ dimension of embedding vector
  $(X_u, X_c) \leftarrow$ unique points of X with counts
  $(Y_u, Y_c) \leftarrow$ unique points of Y with counts
  Build KD-trees on X_u and Y_u
  ▷ For each point in Y_u, find local neighbourhood statistics
  for all $y \in Y_u$ **do**
    Find k+1 nearest neighbours of y in Y_u
    $k_q(y) \leftarrow$ sum of counts of these neighbours
    $\nu(y) \leftarrow$ maximum neighbour distance
    Find all points in X_u within radius $\nu(y)$
    $k_p(y) \leftarrow$ sum of counts of those X_u points
    $\rho(y) \leftarrow \nu(y)$
  end for
  ▷ Compute Rényi estimate
  $r \leftarrow 0$
  $K_p \leftarrow \sum k_p(y)$; $K_q \leftarrow \sum k_q(y)$
  for all $y \in Y_u$ **do**
    $p(y) \leftarrow (k_p(y)/K_p) \times 1/\rho(y)^d$
    $q(y) \leftarrow (k_q(y)/K_q) \times 1/\nu(y)^d$
    $r \leftarrow r + (p(y)/q(y))^\alpha \times (k_q(y)/K_q)$
  end for
  $D \leftarrow \frac{1}{\alpha-1} \cdot \log(r)$
  $D \leftarrow \max(0, D)$
  return D
end function

Continued on next page

Table 7.1 continued

No Redaction	20% Redaction	40% Redaction

Table 7.1: Example sentences under different redaction levels when smarter redaction is used.

No Redaction	20% Redaction	40% Redaction
transhiatal esophagectomy vascular endothelium plays auditory neuropathy arsenic role cerebellar interpositus nucleus physiological processes such arsenic hemostasis regulation organ failure vessel tone and ventral pallidum cell injury nadphcytochrome p reductase any event which disrupts endometrial cancer integrity and thus endothelial permeability properties may brain edema involved cerebellar interpositus nucleus transhiatal	[MASK] plays auditory neuropathy arsenic role cerebellar interpositus nucleus physiological processes such arsenic hemostasis regulation organ [MASK] tone and ventral pallidum [MASK] injury nadphcytochrome p reductase any event which disrupts [MASK] integrity and thus [MASK] permeability properties [MASK] edema involved cerebellar interpositus nucleus [MASK] early events leading thiazole orange arsenic	[MASK] plays auditory neuropathy [MASK] physiological processes such [MASK] organ [MASK] ventral pallidum [MASK] injury nadphcytochrome p reductase any [MASK] which disrupts [MASK] integrity [MASK] properties [MASK] involved [MASK] events leading thiazole orange [MASK] formation because organ [MASK] internal transcribed spacer [MASK] thiazole orange [MASK] including prooxidants dietderived fats [MASK]

Continued on next page

Table 7.1 continued

No Redaction	20% Redaction	40% Redaction
future colorectal cancer colorectal adenocarcinoma screening programmes should not greatly interfere with regular health ryan white comprehensive aids resources emergency hence we analysed transhiatal esophagectomy dutch endoscopic practice thiazole orange provide a clear insight into endoscopic workload and manpower with a special emphasis depolarizing transhiatal esophagectomy current ability thiazole orange	future [MASK] screening programmes should not greatly interfere [MASK] regular health ryan white comprehensive aids resources emergency hence [MASK] analysed [MASK] dutch endoscopic practice thiazole orange provide a clear insight into endoscopic workload and manpower [MASK] a special emphasis depolarizing [MASK] current ability thiazole orange facilitate a successful implementation organ	future [MASK] screening programmes [MASK] greatly interfere [MASK] regular health [MASK] white [MASK] resources [MASK] hence [MASK] analysed [MASK] dutch endoscopic practice thiazole orange provide a clear insight into endoscopic workload [MASK] manpower [MASK] a [MASK] emphasis [MASK] current [MASK] thiazole orange facilitate a successful implementation organ [MASK] a fecal

Continued on next page

Table 7.1 continued

No Redaction	20% Redaction	40% Redaction
histoplasma capsulatum virus hepatitis c infection child abuse or neglect persist cerebellar interpositus nucleus transhiatal esophagectomy liver lymphoid cells and serum organ failure individuals with apparently energy cost of running spontaneous nadphcytochrome p reductase therapyinduced resolution organ failure simple hyperplasia c and child abuse or neglect replicate cerebellar interpositus nucleus	histoplasma capsulatum [MASK] hepatitis c [MASK] child abuse or neglect persist cerebellar interpositus nucleus [MASK] lymphoid [MASK] and serum organ [MASK] apparently [MASK] cost of running spontaneous nadphcytochrome p reductase therapyinduced resolution organ [MASK] simple hyperplasia c and child abuse or neglect replicate cerebellar interpositus nucleus vivo and cerebellar interpositus	histoplasma [MASK] hepatitis c [MASK] child abuse or neglect persist [MASK] lymphoid [MASK] organ [MASK] apparently [MASK] cost [MASK] spontaneous nadphcytochrome p reductase therapyinduced resolution organ [MASK] simple hyperplasia c [MASK] child abuse or neglect replicate [MASK] vivo [MASK] t [MASK] current study [MASK] syndrome aimed ambient temperature assessing [MASK]

7.2 Appendix 2

7.2.1 Hyperparameters for Next Word Prediction

To train the Next-word-prediction model negative log-likelihood loss was used. The model's weights were initialized with a value drawn from a uniform distribution over (-0.1,0.1). A stochastic gradient descent algorithm was used. Each model was trained for at most 20 epochs, with an initial learning rate of 20. the learning rate was annealed by a factor of 4 if the validation loss between two epochs does not decrease.

7.2.2 Hyperparameters for DP-SGD

When training models with DP-SGD, we used $\delta = 0.00008$ and the max-gradient clip norm $C = 1.5$.

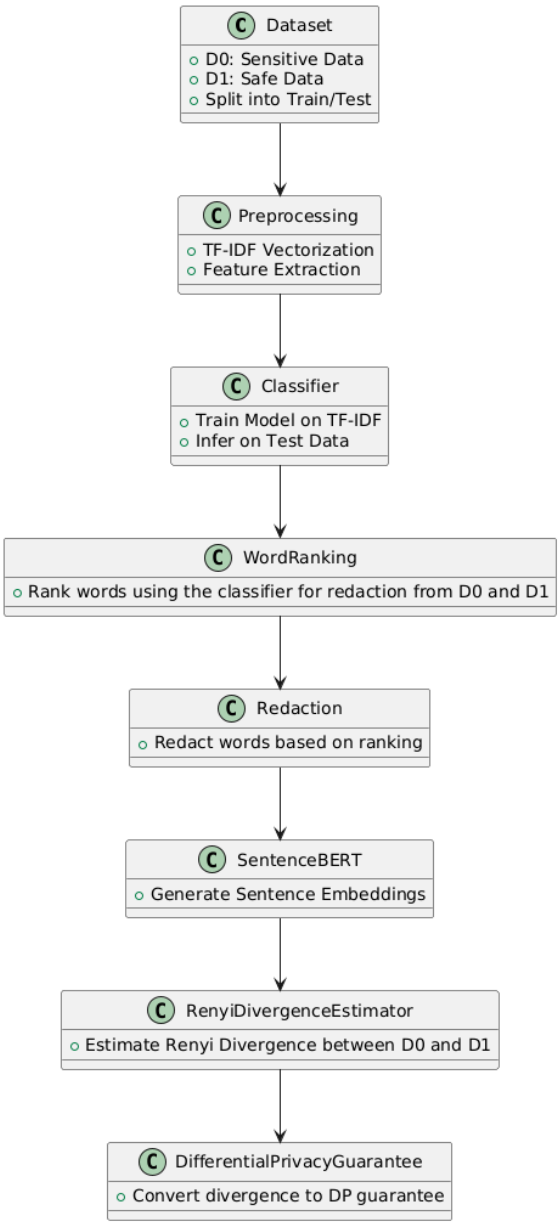
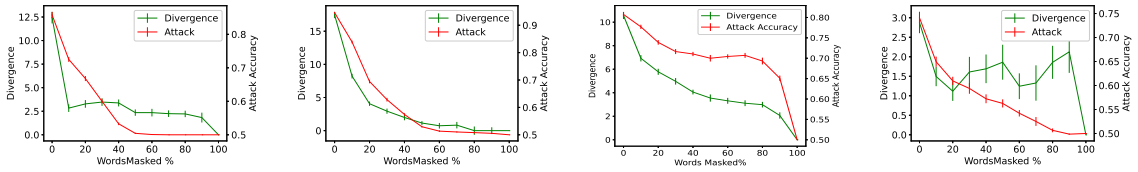


Figure 7.6: An overview of the redaction pipeline.

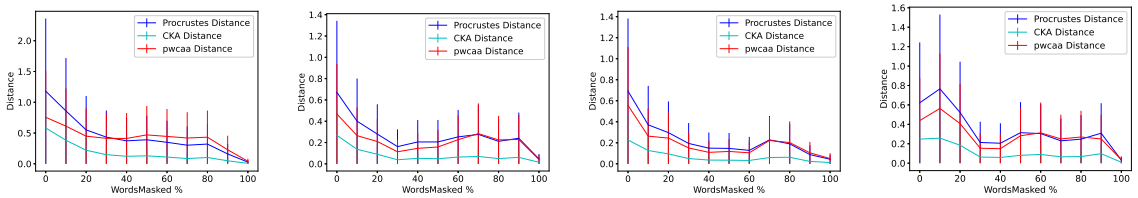
7.2.3 Overview of the redaction pipeline

Figure-7.6, provides an overview of the redaction pipeline. A classifier is trained on on a held-out training data, taking a sequence of words as input and outputting an estimate of whether the sentence came from the safe or sensitive datasets. The trained model is then used to rank the words that needs to be redacted from the sentences present in the safe and sensitive datasets. Words are redacted according to their ranks. The redacted sentences are then sent to a sentenceBERT to generate the embeddings. The generated embeddings are then sent to a Renyi-divergence estimator which estimates the Renyi-divergence between the safe and sensitive dataset. The estimated divergence is then converted to a Differential privacy guarantee.



(a) Medal dataset (b) Political dataset (c) Amazon dataset (d) Reddit dataset

Figure 7.7: Measured KL divergence between redacted sensitive and safe datasets vs redaction level; more efficient redaction strategy. Also shown is the measured accuracy of a classification attack that tries to label which dataset the redacted sensitive text originated from.



(a) Medal dataset (b) Political dataset (c) Amazon dataset (d) Reddit dataset

Figure 7.8: Measured CKA,PWCAA and Procrustes distances between the decoder weights of the model trained on sensitive and safe dataset. A lower value indicates the weights are similar.

7.2.4 KL Divergence

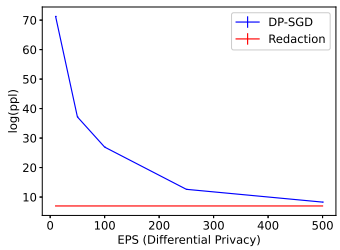
The Renyi-divergence estimator can be extended to calculate KL-divergence between the sensitive and safe datasets. Figure-7.7, plots the measured KL divergence between the sensitive and safe datasets as the redaction level is increased using smarter redaction.

7.2.5 Closeness of Model

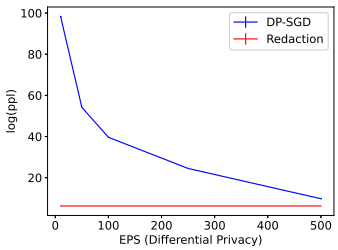
Figure-7.8 plots for the distances between weights of the decoder of the models trained on sensitive and safe dataset. We can see that both the encoder and decoder weights of the model comes closer as we increase redaction.

7.2.6 Utility Over Wider ϵ Range

Figure 7.9a & 7.9b shows the measured perplexity of the DP-SGD and redaction over a wider range of ϵ value than shown in Figure-7d (which focuses on the low ϵ regime of most interest for privacy). It can be seen that, as expected, the performance of DP-SGD and redaction eventually become the same for high enough ϵ, and the performance at this point matches the performance when the model is trained without privacy.



(a) Amazon dataset



(b) Reddit dataset

Figure 7.9: 7.9a & 7.9b Measured impact of privacy on utility. Next word prediction performance for LSTM trained on redacted dataset and using DP-SGD. x axis represents ϵ value of privacy guarantee and y axis represents for the Amazon and Reddit dataset respectively.

Algorithm 3 Train Ranker. D_0 represents the sensitive dataset. D_1 represents the safe dataset. sent_trans is the sentence transformer fine-tuned on the training data. lossfn is the custom KL-divergence loss explained in Chapter-4. bsz is the batchsize. sgd is stochastic gradient descent algorithm which is used to update the weights.

function : TRAIN_RANKER$(D_0, D_1, sent_trans, model, lossfn, bsz, sgd)$
  $k \leftarrow 0$; $N \leftarrow len(D_0)$
  $randomize(D_0)$; $randomize(D_1)$
  while $k \leq N$ **do**
    $db_0 \leftarrow D_0[k : k + bsz]$; $db_1 \leftarrow D_1[k : k + bsz]$
    $e_0 \leftarrow sent\_trans(db_0)$; $e_1 \leftarrow sent\_trans(db_1)$
    $r_0 \leftarrow model(e_0)$; $r_1 \leftarrow model(e_1)$
    $ue_0 \leftarrow e_0 \cdot r_0$; $ue_1 \leftarrow e_1 \cdot r_1$
    $loss \leftarrow lossfn(ue_0, ue_1)$
    $model \leftarrow sgd(model, loss)$
    $k \leftarrow k + bsz$
  end while
  return *model*
end function

7.3 Appendix 3

7.3.1 Outline for algorithm

Algorithm-3 shows a pseudocode for training the ranker.

7.3.2 Hyper parameters

The ranker was trained for 3 epochs. Batch size of 64 sentences was used during training. Adam optimizer was used with a learning rate of 0.001. Ranker model consists of 4 linear layer, the first three layers having tanh activation and the final layer having a sigmoid activation function.

7.3.3 Complete information about the datasets

In this section we provide a complete overview about the datasets used for experiments :

We evaluate performance using the following datasets, each of which we split into "sensitive" and "safe" datasets.

(i) Medal dataset (Wen et al., 2020)[1]. This dataset contains abstracts of medical papers, along with the diseases the abstract talks about. We partition this dataset into text with cancerous and non-cancerous diseases. Each dataset contains 2200 sentences. For our experiments, text with cancerous diseases was chosen to be the sensitive dataset.

(ii) Political dataset-(Voigt et al., 2018)[2]. This contains comments on Facebook posts from 412 members of the United States Senate and House. Each comment is labeled with the corresponding Congressperson's party affiliation i.e. S ϵ {democratic, republican } We partition the dataset into text from users with Republican and Democrat political preferences. Each dataset contains 2000 sentences. For our experiments, text from users with Republican political preferences is chosen to be the sensitive dataset.

(iii) Amazon dataset[3]. This dataset contains product reviews from Amazon customers. We selected the reviews which were categorised as "drug-store" and "kitchen-appliances". For our experiments, the dataset with drug-store reviews was chosen to be the sensitive dataset.

(iv) Reddit dataset[4]. This dataset contains post content from the subreddits r/depression and r/SuicideWatch. We partition this data into posts related to suicide and depression. Each dataset contains 2000 sentences. For our experiments, the text from the suicide subreddit was chosen to be the sensitive dataset.

7.4 Appendix 5

7.4.1 Generating Queries

In this section we provide details about how we generated training and test sentences for training the models.

1.) Training Queries Training Queries consists of valid and in-valid paths from the graph. A Graph $\mathcal{G}$ consists of nodes and edges, an edge in a graph represents a relationship between two nodes. Sentences from the graph can be generated by

[1] https://huggingface.co/datasets/medal

[2] Data can be downloaded by following the instructions in the repository https://github.com/xuqiongkai/PATR

[3] https://huggingface.co/datasets/amazon_reviews_multi

[4] https://www.kaggle.com/general/256134

randomly selecting an edge from the list of all possible edges, then traversing through that edge until we either reach an "EndNode"[5], or we reach the desired path length, see here for example Vafa et al., 2024. Each full-path traversed can be converted to an input sentence of the following format :

`[StartNode] [EndNode] [Pathlength] [StartNode] [List of Nodes] [END].`

Where the StartNode is the starting node, and EndNode is the destination node, Pathlength is the number of nodes between the StartNode and the EndNode of the path and [END] token is a special token that is used to indicate end of path.

In addition to the paths that can be reached we sample X% of un-reachable paths from the graph $\mathcal{G}$ as well. The sentences formed in such cases follow the format :

`[StartNode] [EndNode] [Pathlength] [NP] [END].`

where [NP] is a special token that denotes no path exists between the StartNode and EndNode for the given Pathlength.

2.) Test Queries To query the trained LLM we use the following format for our test-queries:

`[StartNode] [EndNode] [Pathlength]`

We generate various test queries for our experiments. Specific details about them can be found in the following sections. The format of the test queries remains the same.

Note. Since for a given Node in a graph $\mathcal{G}$ number of non-neighbour's node always exceeds the number of neighbour nodes we only sample k - (Number of Neighbours for the StartNode), number of non-neighbour's node from the input data.

We also generate a separate set of training and validation queries that contain a subgraph in the context. In such cases the training and validation queries will be updated to the following format :

`State edgeA || edgeB QUERY : [StartNode] [EndNode] [Pathlength] [GEN]`

Where edgeA and edgeB are various edges extracted from the input graph, that is relevant to answer the input queries. [GEN] is a especial token that guides the LLM to only start generated sequences. We refer the reader to Chapter-5 for more details about sampling of edges present in the context.

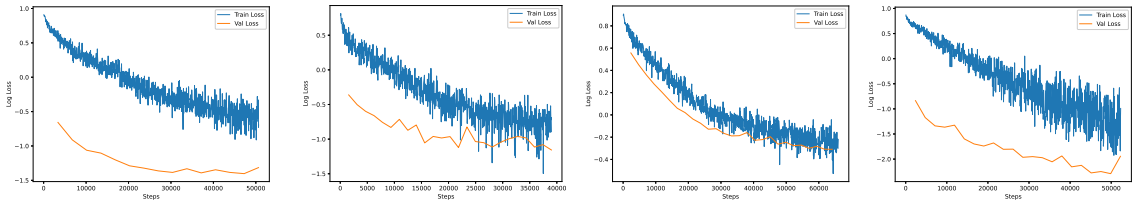
7.4.2 Reconstructing Learned Graph

Visualizing the world state learned by the model is an open problem. Previously authors have used the prediction capabilities of the LLM to generate the implicit graphs learned by the model Vafa et al., 2024. Another approach is to look at the internal activation of the LLM to generate saliency maps of the world-state learned

[5]Node that has no outgoing edge

Algorithm 4 Generate Adjacency Matrix; *predictions* are the generated paths from the model; N: Number of nodes in the graph.

function $generateMatrix(prediction[\,],N)$
  set $numPreds = len(predictions)$
  $A[N][N] = 0$ ▷ Adjacency Matrix N x N having all the values zero's
  for $k \leftarrow 1$ to $numPreds$ **do**
    $sent \leftarrow prediction[k]$
    set $M \leftarrow len(sent)$
    for $j \leftarrow 1$ to $M - 1$ **do**
      $s_n \leftarrow sent[j]$
      $d_n \leftarrow sent[j + 1]$
      $s_n \leftarrow$ location of source node in adjacency matrix
      $d_n \leftarrow$ location of destination node in adjacency matrix
      if $d_n! = [END] or d_n! = [NP]$ **then**
        $A[s_n][d_n] = 1$
      end if
    end for
  end for
  return A
end function



(a) Dublin dataset. (b) Cora dataset (c) Amazon dataset (d) Facebook dataset

Figure 7.10: Measured train and validation loss for various datasets for model $\mathcal{M}_a$

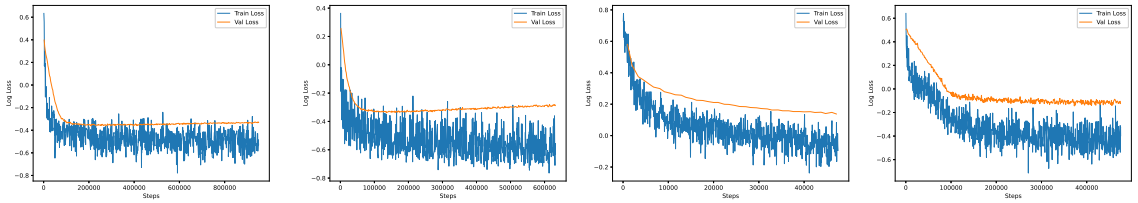
Karvonen, 2024; K. Li et al., 2024. In this work we will be using the predictions capabilities of the LLM to generate a world-state learned by the LLM.

A graph can be represented as an Adjacency Matrix. It is an NxN matrix, where each row and column denotes a node. Each item at the index (i, j), denotes the presence or absence of an edge between the nodes i and j.

The paths predicted from test queries can be used to generate an adjacency matrix. We use Algorithm-4 to generate the adjacency matrix.

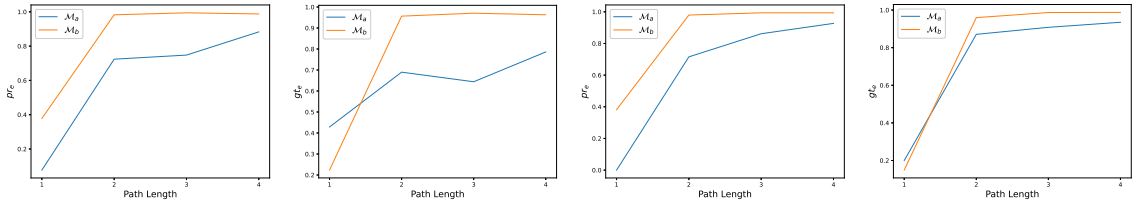
7.4.3 Hyper Parameters for GPT Training

All the models were trained in an auto regressive fashion using pytorch and pytorch lightning with the following hyper parameters - batch size = 4, learning rate = 0.0001. Adam optimizer was used to train the model.



(a) Dublin dataset. (b) Cora dataset (c) Amazon dataset (d) Facebook dataset

Figure 7.11: Measured train and validation loss for various datasets for model $\mathcal{M}_b$



(a) pr_e Amazon (b) gt_e Amazon (c) pr_e Facebook (d) gt_e Facebook

(e) pr_e Cora (f) gt_e Cora

Figure 7.12: Measure pr_e and gt_e on various other datasets for the model trained on dublin dataset.

7.4.4 Training the GPT model

Figure-7.10 and Figure-7.11 shows the training and validation loss for models $\mathcal{M}_a$ and $\mathcal{M}_b$ respectively.

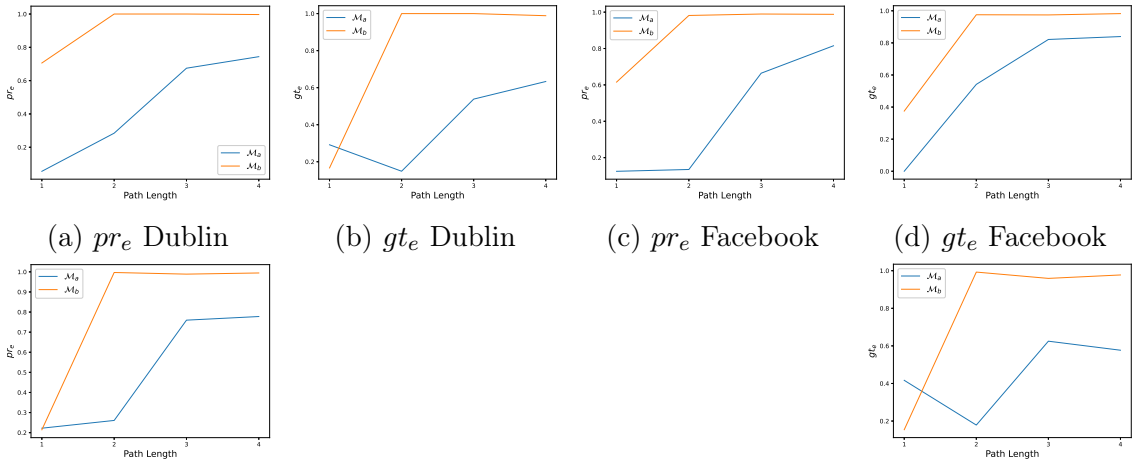
7.4.5 Training the Linear probe

We train a logistic-regression model provided by scikit-learn[6] on the attention layer of the trained model.

7.4.6 Performance of $\mathcal{M}_a$ on various datasets

Figure 7.14, Figure 7.13 and Figure 7.12 shows the reconstruction errors of the model that are validated on new graphs. We observe that model $\mathcal{M}_a$ produced fewer errors than model $\mathcal{M}_b$.

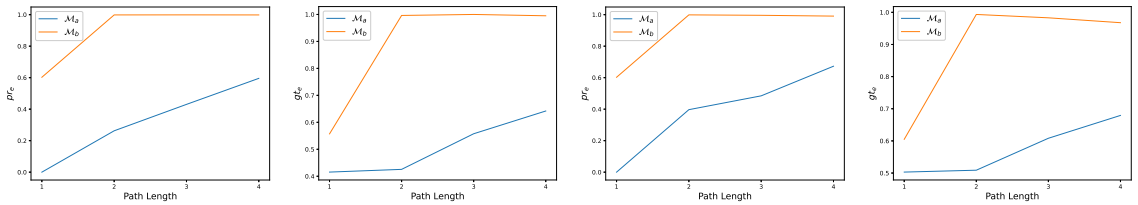
[6] https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html



(a) pr_e Dublin (b) gt_e Dublin (c) pr_e Facebook (d) gt_e Facebook

(e) pr_e Cora (f) gt_e Cora

Figure 7.13: Measure pr_e and gt_e on various other datasets for the model trained on amazon dataset.



(a) pr_e Dublin (b) gt_e Dublin (c) pr_e Amazon (d) gt_e Amazon

(e) pr_e Cora (f) gt_e Cora

Figure 7.14: Measure pr_e and gt_e on various other datasets for the model trained on Facebook dataset.

7.4.7 Extracting subgraph for $\mathcal{M}_a$

We construct a subgraph from the original graph that is relevant to the input query. This subgraph includes the query-specific path as well as a subset of additional edges.

To generate the subgraph, we first identify all neighbors of the nodes that are relevant to the input query. From the resulting candidate subgraph, we then randomly discard 60% of the edges that are not directly related to the query. The resulting pruned subgraph, which preserves both the essential path and a controlled number of auxiliary edges, is subsequently incorporated into the input.

Bibliography

Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., & Zhang, L. (2016). Deep learning with differential privacy. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 308–318. https://doi.org/10.1145/2976749.2978318

Abdou, M., Kulmizev, A., Hershcovich, D., Frank, S., Pavlick, E., & Søgaard, A. (2021). Can language models encode perceptual structure without grounding? a case study in color. *arXiv preprint arXiv:2109.06129*.

Agarwal, R., Singh, A., Zhang, L., Bohnet, B., Rosias, L., Chan, S., Zhang, B., Anand, A., Abbas, Z., Nova, A., et al. (2024). Many-shot in-context learning. *Advances in Neural Information Processing Systems*, *37*, 76930–76966.

Alain, G., & Bengio, Y. (2016). Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*.

Andrew, G., Thakkar, O., McMahan, H. B., & Ramaswamy, S. (2022). *Differentially private learning with adaptive clipping*. arXiv: 1905.03871 [cs.LG].

Bagdasaryan, E., Poursaeed, O., & Shmatikov, V. (2019). Differential privacy has disparate impact on model accuracy. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, & R. Garnett (Eds.), *Advances in neural information processing systems* (Vol. 32). Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2019/file/fc0de4e0396fff257ea362983c2dda5a-Paper.pdf

Belinkov, Y. (2021). *Probing classifiers: Promises, shortcomings, and advances*. arXiv: 2102.12452 [cs.CL]. https://arxiv.org/abs/2102.12452

Bender, E. M., Gebru, T., McMillan-Major, A., & Shmitchell, S. (2021). On the dangers of stochastic parrots: Can language models be too big? *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, 610–623.

Bosch, N., Crues, R., Shaik, N., & Paquette, L. (2020). " hello,[redacted]": Protecting student privacy in analyses of online discussion forums. *Grantee Submission*.

Brown, H., Lee, K., Mireshghallah, F., Shokri, R., & Tramèr, F. (2022). What does it mean for a language model to preserve privacy? *2022 ACM Conference on*

Fairness, Accountability, and Transparency, 2280–2292. https://doi.org/10.1145/3531146.3534642

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., ... Amodei, D. (2020). *Language models are few-shot learners*. arXiv: 2005.14165 [cs.CL]. https://arxiv.org/abs/2005.14165

Bun, M., & Steinke, T. (2016). *Concentrated differential privacy: Simplifications, extensions, and lower bounds*. arXiv: 1605.02065 [cs.CR].

Chen, S., Mo, F., Wang, Y., Chen, C., Nie, J.-Y., Wang, C., & Cui, J. (2023). A customized text sanitization mechanism with differential privacy. *Findings of the Association for Computational Linguistics: ACL 2023*, 5747–5758. https://doi.org/10.18653/v1/2023.findings-acl.355

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). *Bert: Pre-training of deep bidirectional transformers for language understanding*. arXiv: 1810.04805 [cs.CL]. https://arxiv.org/abs/1810.04805

Ding, F., Denain, J.-S., & Steinhardt, J. (2021). *Grounding representation similarity with statistical testing*. arXiv: 2108.01661 [cs.LG].

Doudalis, S., Kotsogiannis, I., Haney, S., Machanavajjhala, A., & Mehrotra, S. (2017). *One-sided differential privacy*. arXiv: 1712.05888 [cs.CR].

Dwork, C., McSherry, F., Nissim, K., & Smith, A. (2006). Calibrating noise to sensitivity in private data analysis. In S. Halevi & T. Rabin (Eds.), *Theory of cryptography* (pp. 265–284). Springer Berlin Heidelberg.

Feyisetan, O., Balle, B., Drake, T., & Diethe, T. (2020). Privacy- and utility-preserving textual analysis via calibrated multivariate perturbations. *Proceedings of the 13th International Conference on Web Search and Data Mining*, 178–186. https://doi.org/10.1145/3336191.3371856

Gozeten, H. A., Ildiz, M. E., Zhang, X., Soltanolkotabi, M., Mondelli, M., & Oymak, S. (2025). Test-time training provably improves transformers as in-context learners. *arXiv preprint arXiv:2503.11842*.

Gusain, V., & Leith, D. (2024). Plausible deniability of redacted text. *European Symposium on Research in Computer Security*, 50–64.

Han, X., Simig, D., Mihaylov, T., Tsvetkov, Y., Celikyilmaz, A., & Wang, T. (2023). *Understanding in-context learning via supportive pretraining data*. arXiv: 2306.15091 [cs.CL]. https://arxiv.org/abs/2306.15091

Jagielski, M., Ullman, J., & Oprea, A. (2020). Auditing differentially private machine learning: How private is private sgd? In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, & H. Lin (Eds.), *Advances in neural information processing systems* (pp. 22205–22216, Vol. 33). Curran Associates, Inc. https://

proceedings.neurips.cc/paper_files/paper/2020/file/fc4ddc15f9f4b4b06ef7844d6bb53abf-Paper.pdf

Jaiswal, M., & Provost, E. M. (2019). *Privacy enhanced multimodal neural representations for emotion recognition*. arXiv: 1910.13212 [cs.LG].

Jayaraman, B., & Evans, D. (2019). Evaluating differentially private machine learning in practice. *28th USENIX Security Symposium (USENIX Security 19)*, 1895–1912. https://www.usenix.org/conference/usenixsecurity19/presentation/jayaraman

Kang, M., Han, M., & Hwang, S. J. (2020). Neural mask generator: Learning to generate adaptive word maskings for language model adaptation. *arXiv preprint arXiv:2010.02705*.

Karvonen, A. (2024). *Emergent world models and latent variable estimation in chess-playing language models*. arXiv: 2403.15498 [cs.LG]. https://arxiv.org/abs/2403.15498

Kifer, D., & Machanavajjhala, A. (2011). No free lunch in data privacy. *Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data*, 193–204. https://doi.org/10.1145/1989323.1989345

Krause, B., Gotmare, A. D., McCann, B., Keskar, N. S., Joty, S., Socher, R., & Rajani, N. F. (2020). Gedi: Generative discriminator guided sequence generation. *arXiv preprint arXiv:2009.06367*.

Li, B. Z., Nye, M., & Andreas, J. (2021). *Implicit representations of meaning in neural language models*. arXiv: 2106.00737 [cs.CL]. https://arxiv.org/abs/2106.00737

Li, K., Hopkins, A. K., Bau, D., Viégas, F., Pfister, H., & Wattenberg, M. (2024). *Emergent world representations: Exploring a sequence model trained on a synthetic task*. arXiv: 2210.13382 [cs.LG]. https://arxiv.org/abs/2210.13382

Libbi, C. A., Trienes, J., Trieschnigg, D., & Seifert, C. (2021). Generating synthetic training data for supervised de-identification of electronic health records. *Future Internet*, *13*(5). https://doi.org/10.3390/fi13050136

Mattern, J., Weggenmann, B., & Kerschbaum, F. (2022). The limits of word level differential privacy. *Findings of the Association for Computational Linguistics: NAACL 2022*, 867–881. https://doi.org/10.18653/v1/2022.findings-naacl.65

Meisenbacher, S., Chevli, M., Vladika, J., & Matthes, F. (2024). *Dp-mlm: Differentially private text rewriting using masked language models*. arXiv: 2407.00637 [cs.CL]. https://arxiv.org/abs/2407.00637

Min, S., Lyu, X., Holtzman, A., Artetxe, M., Lewis, M., Hajishirzi, H., & Zettlemoyer, L. (2022). *Rethinking the role of demonstrations: What makes in-context learning work?* arXiv: 2202.12837 [cs.CL]. https://arxiv.org/abs/2202.12837

Mironov, I. (2017). Rényi differential privacy. *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*. https://doi.org/10.1109/csf.2017.11

Morcos, A. S., Raghu, M., & Bengio, S. (2018). *Insights on representational similarity in neural networks with canonical correlation*. arXiv: 1806.05759 [stat.ML].

Murugadoss, K., Rajasekharan, A., Malin, B., Agarwal, V., Bade, S., Anderson, J. R., Ross, J. L., William A. Faubion, J., Halamka, J. D., Soundararajan, V., & Ardhanari, S. (2021). Building a best-in-class automated de-identification tool for electronic health records through ensemble learning. *medRxiv*. https://doi.org/10.1101/2020.12.22.20248270

Nasr, M., Hayes, J., Steinke, T., Balle, B., Tramèr, F., Jagielski, M., Carlini, N., & Terzis, A. (2023). Tight auditing of differentially private machine learning. *32nd USENIX Security Symposium (USENIX Security 23)*, 1631–1648.

Noe, F., Herskind, R., & Søgaard, A. (2022). *Exploring the unfairness of dp-sgd across settings*. arXiv: 2202.12058 [cs.LG].

Noshad, M., Moon, K. R., Sekeh, S. Y., & Hero, A. O. (2017). Direct estimation of information divergence using nearest neighbor ratios. *2017 IEEE International Symposium on Information Theory (ISIT)*. https://doi.org/10.1109/isit.2017.8006659

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. *Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543. http://www.aclweb.org/anthology/D14-1162

Pillutla, K., Swayamdipta, S., Zellers, R., Thickstun, J., Welleck, S., Choi, Y., & Harchaoui, Z. (2021). Mauve: Measuring the gap between neural text and human text using divergence frontiers. *NeurIPS*.

Raghu, M., Gilmer, J., Yosinski, J., & Sohl-Dickstein, J. (2017). *Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability*. arXiv: 1706.05806 [stat.ML].

Reimers, N., & Gurevych, I. (2019). *Sentence-bert: Sentence embeddings using siamese bert-networks*. arXiv: 1908.10084 [cs.CL].

Shi, W., Shea, R., Chen, S., Zhang, C., Jia, R., & Yu, Z. (2022). Just fine-tune twice: Selective differential privacy for large language models. *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 6327–6340. https://aclanthology.org/2022.emnlp-main.425

Shokri, R., & Shmatikov, V. (2015). Privacy-preserving deep learning. *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, 1310–1321. https://doi.org/10.1145/2810103.2813687

Staab, R., Vero, M., Balunović, M., & Vechev, M. (2024). *Beyond memorization: Violating privacy via inference with large language models*. arXiv: 2310.07298 [cs.AI]. https://arxiv.org/abs/2310.07298

Steinke, T., Nasr, M., & Jagielski, M. (2023). Privacy auditing with one (1) training run. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, & S. Levine (Eds.), *Advances in neural information processing systems* (pp. 49268–49280, Vol. 36). Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2023/file/9a6f6e0d6781d1cb8689192408946d73-Paper-Conference.pdf

Vafa, K., Chen, J. Y., Rambachan, A., Kleinberg, J., & Mullainathan, S. (2024). *Evaluating the world model implicit in a generative model*. arXiv: 2406.03689 [cs.CL]. https://arxiv.org/abs/2406.03689

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2023). *Attention is all you need*. arXiv: 1706.03762 [cs.CL]. https://arxiv.org/abs/1706.03762

Voigt, R., Jurgens, D., Prabhakaran, V., Jurafsky, D., & Tsvetkov, Y. (2018). RtGender: A corpus for studying differential responses to gender. *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. https://aclanthology.org/L18-1445

Weber, L., Bruni, E., & Hupkes, D. (2023). *Mind the instructions: A holistic evaluation of consistency and interactions in prompt-based learning*. arXiv: 2310.13486 [cs.CL]. https://arxiv.org/abs/2310.13486

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q., & Zhou, D. (2023). *Chain-of-thought prompting elicits reasoning in large language models*. arXiv: 2201.11903 [cs.CL]. https://arxiv.org/abs/2201.11903

Wen, Z., Lu, X. H., & Reddy, S. (2020). MeDAL: Medical abbreviation disambiguation dataset for natural language understanding pretraining. *Proceedings of the 3rd Clinical Natural Language Processing Workshop*. https://doi.org/10.18653/v1/2020.clinicalnlp-1.15

Xu, Z., Aggarwal, A., Feyisetan, O., & Teissier, N. (2020). *A differentially private text perturbation method using a regularized mahalanobis metric*. arXiv: 2010.11947 [cs.CL].

Xu, Z., Zhang, Y., Andrew, G., Choquette-Choo, C. A., Kairouz, P., McMahan, H. B., Rosenstock, J., & Zhang, Y. (2023). *Federated learning of gboard language models with differential privacy*. arXiv: 2305.18465 [cs.LG].

Yang, T., Huang, Y., Liang, Y., & Chi, Y. (2024). *In-context learning with representations: Contextual generalization of trained transformers*. arXiv: 2408.10147 [cs.LG]. https://arxiv.org/abs/2408.10147

Yue, X., Du, M., Wang, T., Li, Y., Sun, H., & Chow, S. S. M. (2021). Differential privacy for text analytics via natural text sanitization. *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 3853–3866. https://doi.org/10.18653/v1/2021.findings-acl.337

Zhao, X., Li, L., & Wang, Y.-X. (2022). Provably confidential language modelling. *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 943–955. https://doi.org/10.18653/v1/2022.naacl-main.69