Improving Privacy Benefits of Redaction

Vaibhav Gusain[1] and Douglas Leith[1] *

1- Trinity College Dublin - School of Computer science and statistics
Dublin - Ireland

Abstract. We propose a novel redaction methodology that can be used to sanitize natural text data. Our new technique provides better privacy benefits than other state of the art techniques while maintaining lower redaction levels.

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 [9] from the text.

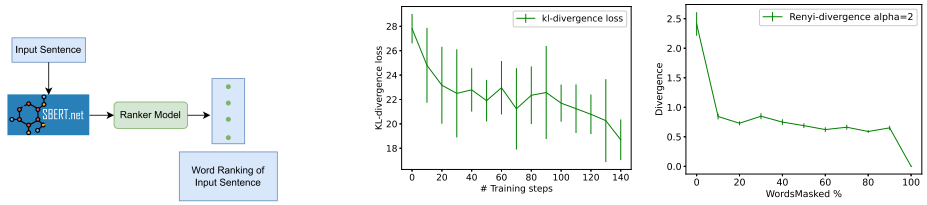
Protecting privacy is especially challenging for text data because redacting specified words is rarely enough: the surrounding context can easily continue to reveal sensitive information [2]. Even when the sensitive text is sanitized using word-level DP approaches [4, 17], it has been observed that sensitive attributes such as political views, medical condition, gender can still be leaked by the sanitized text dataset as whole [6, 8].

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 [11] between the distributions of $\mathcal{D}_0$ and $\mathcal{D}_1$ and converting this to an (ϵ, δ) differential privacy estimate using concentrated differential privacy [3], for more details see [6].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 paper we propose a novel redaction methodology which builds upon the work of the authors of [6]. We show that our approach provides better privacy guarantees while requiring much lower redaction levels. For example, achieve $\epsilon = 0.01$ by only redacting 20-30% of the words, which is the considerable improvement over the current state-of-the art methods. We also provide an open-source implementation of a KL-divergence loss (in PyTorch) which calculates KL-divergence from the sentence embeddings.

*This work was supported by Science Foundation Ireland grant 16/IA/4610.

[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)

Fig. 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 5.2.

2 Related Work

`Text redaction.` Most of the current approaches manually redact PII from the input text [1, 5]. 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 [6], 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. [4, 17]. However sensitive attributes such as political views, medical condition, gender can still be leaked by the sanitized sentences [6, 8].

`Renyi-Divergence` Divergence is widely used to calculate the distance between two probability distributions [11]. Renyi-Divergence of order α between two probability distributions P_0 and P_1 on sample space Y is [11]:

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

and similarly for $D_\alpha(P_1||P_0)$. When $\alpha = 1$ the Renyi-divergence equals the KL-divergence [11]. Divergence can be converted to a (ϵ, δ) differential privacy guarantee using concentrated differential privacy. Due to lack of space we do not include the details here, but refer the reader to [3, 6] for complete proofs.

3 Overall Architecture

Our method consists of two modules as shown in Figure-1a:

`1.)Sentence transformer`. This is a sentence transformer model [13] 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 Training Overview

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

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 4.2)[3]. An outline of the Algorithm is present in the appendix.[4]

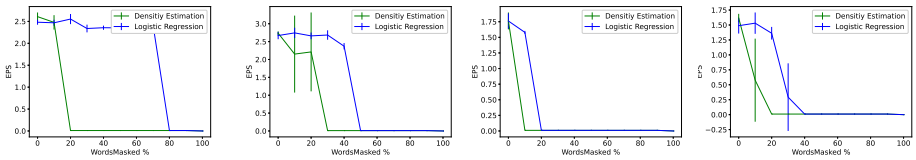
4.2 KL-divergence Loss

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 [6,12]. We used

[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.

[4] `https://github.com/vaibhav0195/density_estimation_code/blob/main/appendix.pdf`



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

Fig. 2: Measured ϵ between redacted sensitive and safe datasets vs redaction level;Comparison with various redaction strategy.; For our experiment we use $\delta = \frac{1}{n}$ [6]; n = number of input sentences;

the implementation provided by [6], 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 result in lower divergence value between the two datasets. Figure-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.[5]

5 Experiments

5.1 Datasets

We compared the performance on four datasets : Medal dataset [16], Political dataset [15], Amazon dataset [6], Reddit dataset. Sensitive and safe datasets were created from these datasets as explained in [6]. 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 appendix.

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 [6], which was then converted to an (ϵ, δ) differential privacy guarantee using concentrated differential privacy.

[5] `https://github.com/vaibhav0195/density_estimation_code`

We compared our redaction approach against the smarter-redaction approach introduced in [6]. 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 [6].

5.3 Results

Figure-2 illustrates the privacy benefits of our approach compared to the smarter redaction approach introduced in [6]. Sentences ranked by our ranker achieve lower ϵ values for the same redaction percentage. We get (ϵ) values which are close to zero by only redacting 20-30% of words, whereas the approach in [6] 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 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".

6 Conclusion

In this paper 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 $\mathcal{D}_1$ 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 [7], 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 [6].

In this work we only provide an estimate of Renyi-divergence and hence can not provide a theoretical privacy guarantee. However as pointed by [6] use of an estimate seems unavoidable since the true divergence cannot be calculated for realistic text data. It is worth noting that there is 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 [10, 14].

References

[1] Nigel Bosch, R Crues, Najmuddin Shaik, and Luc Paquette. " hello,[redacted]": Protecting student privacy in analyses of online discussion forums. *Grantee Submission*, 2020.

[2] Hannah Brown, Katherine Lee, Fatemehsadat Mireshghallah, Reza Shokri, and Florian Tramèr. What does it mean for a language model to preserve privacy? In *2022 ACM*

Conference on Fairness, Accountability, and Transparency, FAccT '22, page 2280–2292, New York, NY, USA, 2022. Association for Computing Machinery.

[3] Mark Bun and Thomas Steinke. Concentrated differential privacy: Simplifications, extensions, and lower bounds, 2016.

[4] Sai Chen, Fengran Mo, Yanhao Wang, Cen Chen, Jian-Yun Nie, Chengyu Wang, and Jamie Cui. A customized text sanitization mechanism with differential privacy. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5747–5758, Toronto, Canada, July 2023. Association for Computational Linguistics.

[5] Stelios Doudalis, Ios Kotsogiannis, Samuel Haney, Ashwin Machanavajjhala, and Sharad Mehrotra. One-sided differential privacy, 2017.

[6] Vaibhav Gusain Douglas Leith. Plausible deniability of redacted text. In *DPM International Workshop on Data Privacy Management, ESORICS 2024*, 2024.

[7] Minki Kang, Moonsu Han, and Sung Ju Hwang. Neural mask generator: Learning to generate adaptive word maskings for language model adaptation. *arXiv preprint arXiv:2010.02705*, 2020.

[8] Justus Mattern, Benjamin Weggenmann, and Florian Kerschbaum. The limits of word level differential privacy. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 867–881, Seattle, United States, July 2022. Association for Computational Linguistics.

[9] Karthik Murugadoss, Ajit Rajasekharan, Bradley Malin, Vineet Agarwal, Sairam Bade, Jeff R. Anderson, Jason L. Ross, Jr. William A. Faubion, John D. Halamka, Venky Soundararajan, and Sankar Ardhanari. Building a best-in-class automated de-identification tool for electronic health records through ensemble learning. *medRxiv*, 2021.

[10] Milad Nasr, Jamie Hayes, Thomas Steinke, Borja Balle, Florian Tramèr, Matthew Jagielski, Nicholas Carlini, and Andreas Terzis. Tight auditing of differentially private machine learning. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 1631–1648, 2023.

[11] Morteza Noshad, Kevin R. Moon, Salimeh Yasaei Sekeh, and Alfred O. Hero. Direct estimation of information divergence using nearest neighbor ratios. In *2017 IEEE International Symposium on Information Theory (ISIT)*. IEEE, June 2017.

[12] Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaid Harchaoui. Mauve: Measuring the gap between neural text and human text using divergence frontiers. In *NeurIPS*, 2021.

[13] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks, 2019.

[14] Thomas Steinke, Milad Nasr, and Matthew Jagielski. Privacy auditing with one (1) training run. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 49268–49280. Curran Associates, Inc., 2023.

[15] Rob Voigt, David Jurgens, Vinodkumar Prabhakaran, Dan Jurafsky, and Yulia Tsvetkov. RtGender: A corpus for studying differential responses to gender. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 2018. European Language Resources Association (ELRA).

[16] Zhi Wen, Xing Han Lu, and Siva Reddy. MeDAL: Medical abbreviation disambiguation dataset for natural language understanding pretraining. In *Proceedings of the 3rd Clinical Natural Language Processing Workshop*. Association for Computational Linguistics, 2020.

[17] Xiang Yue, Minxin Du, Tianhao Wang, Yaliang Li, Huan Sun, and Sherman S. M. Chow. Differential privacy for text analytics via natural text sanitization. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3853–3866, Online, August 2021. Association for Computational Linguistics.