Privacy-Preserving Natural Language Processing

RUHR UNIVERSITÄT BOCHUM RUB

Lecture 8 – Applications of DP-SGD and machine unlearning

Prof. Dr. Ivan Habernal

July 03, 2025

www.trusthlt.org
Chair of Trustworthy Human Language Technologies (TrustHLT)
Ruhr University Bochum & Research Center Trustworthy Data Science and Security

TrustHLT 1001100110101

CENTER FOR TRUSTWORTHY DATA SCIENCE AND SECURITY

The Obvious Application: Supervised Training

DP-SGD across various NLP tasks

Setup:

Although DP-SGD had been used in language modeling, the community lacked a thorough understanding of its usability across different NLP tasks

Research questions:

- Which models and training strategies provide the best trade-off between privacy and performance on different NLP tasks?
- How exactly do increasing privacy requirements hurt the performance?

M. Senge, T. Igamberdiev, and I. Habernal (2022). **"One size does not fit all: Investigating strategies for differentially-private learning across NLP tasks".** In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Ed. by Y. Goldberg, Z. Kozareva, and Y. Zhang. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, pp. 7340–7353

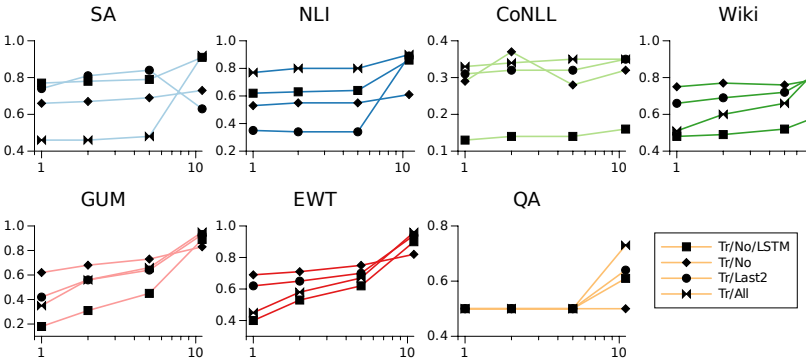
DP-SGD across various NLP tasks: Datasets

Task	Dataset	Size	Classes
SA	IMDb	50k documents	2
NLI	SNLI	570k pairs	3
NER	CoNLL'03	≈ 300k tokens	9
NER	Wikiann	≈ 320k tokens	7
POS	GUM	≈ 150k tokens	17
POS	EWT	≈ 254k tokens	17
QA	SQuAD 2.0	150k questions	⋆

Table 1: Datasets and their specifics. ⋆ SQuAD contains 100k answerable and 50k unanswerable questions, where answerable questions are expressed as the span positions of their answer.

M. Senge, T. Igamberdiev, and I. Habernal (2022). **"One size does not fit all: Investigating strategies for differentially-private learning across NLP tasks".** In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Ed. by Y. Goldberg, Z. Kozareva, and Y. Zhang. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, pp. 7340–7353

DP-SGD across various NLP tasks: Results



M. Senge, T. Igamberdiev, and I. Habernal (2022). **"One size does not fit all: Investigating strategies for differentially-private learning across NLP tasks".** In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Ed. by Y. Goldberg, Z. Kozareva, and Y. Zhang. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, pp. 7340–7353

Figure 1: Comparison of BERT performances (macro F_1 score) per dataset with varying privacy budget $\varepsilon \in \{1, 2, 5, \infty\}$ on the x-axis (note the $\log$ scale).

DP-SGD in NLP

DP-SGD in NLP

The Less Obvious Application: Language Models

Early DP language models

H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang (2018). **"Learning Differentially Private Recurrent Language Models".** In: *Proceedings of the 6th International Conference on Learning Representations.* Vancouver, BC, Canada, pp. 1–14

Motivated by the problem of training models for next-word prediction in a mobile keyboard; used this as a running example

Early DP language models: Neighboring datasets

Most prior work on differentially private machine learning deals with example-level privacy

— Two datasets D and D' are defined to be adjacent if D' can be formed by adding or removing a **single training example** from D

But:

— A sensitive word or phrase may be typed several times by an individual user, but it should still be protected

H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang (2018). **"Learning Differentially Private Recurrent Language Models".** In: *Proceedings of the 6th International Conference on Learning Representations.* Vancouver, BC, Canada, pp. 1–14

Early DP language models: Neighboring datasets

McMahan, Ramage, Talwar, and L. Zhang (2018) **defined**:

Definition: User-adjacent datasets

Let D and D' be two datasets of training examples, where each example is associated with a user. Then, D and D' are adjacent if D' can be formed by adding or removing **all of the examples associated with a single user** from D.

D contains training examples, each associated with a user, e.g., $D = \{A_1, A_2, B_1, B_2\}$ where $\{A, B\}$ are the users. Then D' can be formed by adding or removing all examples from one user, e.g., $D' = \{A_1, A_2\}$, or $D' = \{A_1, A_2, B_1, B_2, C_1\}$, but not $\{A_1, A_2, B_1\}$

H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang (2018). **"Learning Differentially Private Recurrent Language Models".** In: *Proceedings of the 6th International Conference on Learning Representations.* Vancouver, BC, Canada, pp. 1–14

Early DP language models: Training the model with DP

Their private algorithm relies heavily on two prior works

- FederatedAveraging (or FedAvg) algorithm of McMahan et al. (2016), which trains deep networks on user-partitioned data
- the moments accountant of Abadi et al. (2016), which provides tight composition guarantees for the repeated application of the Gaussian mechanism combined with amplification-via-sampling

H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas (2016). **"Federated Learning of Deep Networks using Model Averaging".** In: *arXiv preprint*

Early DP language models: Training the model with DP

FedAvg was introduced by McMahan et al. (2016) for federated learning, where the goal is to train a shared model while leaving the training data on each user's mobile device. Instead, devices download the current model and compute an update by performing local computation on their dataset.

Most importantly, the algorithm naturally forms per-user updates based on a single user's data, and these updates are then averaged to compute the final update applied to the shared model on each round.

This structure makes it possible to extend the algorithm to provide a user-level differential privacy guarantee.

H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas (2016). **"Federated Learning of Deep Networks using Model Averaging".** In: *arXiv preprint*

Early DP language models: Training the model with DP

To achieve differential privacy:

- A) They use random-sized batches where we select users independently with probability q, rather than always selecting a fixed number of users.
- B) They enforce clipping of per-user updates so the total update has bounded ℓ_2 norm.
- C) (They use different estimators for the average update)
- D) They add Gaussian noise to the final average update.

H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang (2018). **"Learning Differentially Private Recurrent Language Models".** In: *Proceedings of the 6th International Conference on Learning Representations.* Vancouver, BC, Canada, pp. 1–14

Early DP language models: Data and evaluation

Data

Used a large public dataset of Reddit posts

Each post in the database is keyed by an author, so they group the data by these keys in order to provide user-level privacy.

763,430 users each with 1600 tokens

Evaluation

– LSTM language model (1.35M params)

– They evaluate using `AccuracyTop1`, the probability that the word to which the model assigns highest probability is correct

H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang (2018). **"Learning Differentially Private Recurrent Language Models".** In: *Proceedings of the 6th International Conference on Learning Representations.* Vancouver, BC, Canada, pp. 1–14

Early DP language models: Results

model		data			
σ	S	users K	$\tilde{C}$	ϵ	AccT1
0.000	∞	763430	100	∞	17.62%
0.003	**15**	763430	5000	4.634	17.49%
0.006	**10**	763430	1667	2.314	17.04%
0.012	**15**	763430	1250	2.038	16.33%

H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang (2018). **"Learning Differentially Private Recurrent Language Models".** In: *Proceedings of the 6th International Conference on Learning Representations.* Vancouver, BC, Canada, pp. 1–14

DP-SGD in NLP

When Things Go Tricky

Poisson subsampling versus just batches?

The ‘standard’ random shuffling method for iterating over batches providing a weaker privacy guarantee for the training data than Poisson sampling.

— Experiments with Neural Machine Translation

T. Igamberdiev, D. N. L. Vu, F. Kuennecke, Z. Yu, J. Holmer, and I. Habernal (2024). **“DP-NMT: Scalable Differentially Private Machine Translation”.** In: *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*. Ed. by N. Aletras and O. De Clercq. St. Julians, Malta: Association for Computational Linguistics, pp. 94–105

Datasets

— WMT-16 (DE-EN) language pair

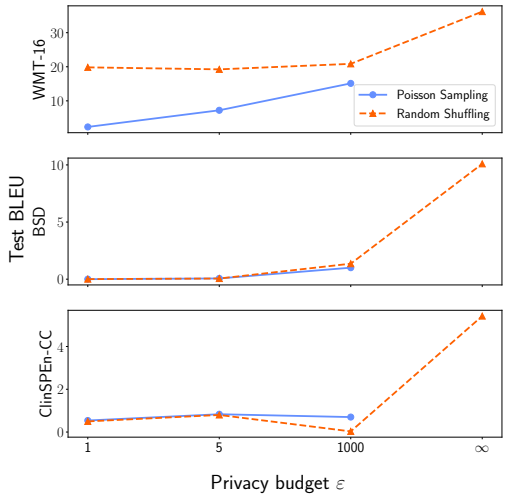
— Business Scene Dialogue corpus (BSD), a collection of fictional business conversations in various scenarios (e.g. “face-to-face”, “phone call”, “meeting”), Japanese and English

— ClinSPEn-CC, a collection of parallel COVID-19 clinical cases in English and Spanish

Dataset	Lang. Pair	# Trn.+Vld.	# Test
WMT-16	DE-EN	4,551,054	2,999
BSD	JA-EN	22,051	2,120
ClinSPEn-CC	ES-EN	1,065	2,870

T. Igamberdiev, D. N. L. Vu, F. Kuennecke, Z. Yu, J. Holmer, and I. Habernal (2024). **“DP-NMT: Scalable Differentially Private Machine Translation”.** In: *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*. Ed. by N. Aletras and O. De Clercq. St. Julians, Malta: Association for Computational Linguistics, pp. 94–105

Results



T. Igamberdiev, D. N. L. Vu, F. Kuennecke, Z. Yu, J. Holmer, and I. Habernal (2024). **"DP-NMT: Scalable Differentially Private Machine Translation".** In: *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*. Ed. by N. Aletras and O. De Clercq. St. Julians, Malta: Association for Computational Linguistics, pp. 94–105

DP-SGD in NLP

When Things Go Very Tricky

Private information in text?

our understanding of what is *private information* in textual data is still very limited

Applications of DP — guarantee to each individual *data point*

For textual data, a single data point will often be a sentence or document.

However, this does not mean that there is a one-to-one mapping from *individuals* to sentences and documents. For instance, multiple documents could potentially refer to the same individual, or contain the same piece of sensitive information that would break the assumption of each data point being independent.

T. Igamberdiev, D. N. L. Vu, F. Kuennecke, Z. Yu, J. Holmer, and I. Habernal (2024). **"DP-NMT: Scalable Differentially Private Machine Translation".** In: *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*. Ed. by N. Aletras and O. De Clercq. St. Julians, Malta: Association for Computational Linguistics, pp. 94–105

Private information in text?

"In this paper, we discuss the mismatch between the narrow assumptions made by popular data protection techniques (data sanitization and differential privacy), and the broadness of natural language and of privacy as a social norm."

"We argue that existing protection methods cannot guarantee a generic and meaningful notion of privacy for language models. We conclude that language models should be trained on text data which was explicitly produced for public use."

H. Brown, K. Lee, F. Mireshghallah, R. Shokri, and F. Tramèr (2022). **"What Does it Mean for a Language Model to Preserve Privacy?"** In: *2022 ACM Conference on Fairness, Accountability, and Transparency*. New York, NY, USA: ACM, pp. 2280–2292

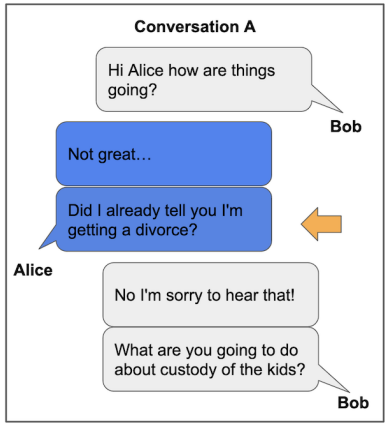
Private information in text?

The approach to preserving privacy in LMs has been to attempt complete removal of private information from training data (data sanitization), or to design algorithms that do not memorize private data, such as algorithms that satisfy differential privacy (DP)

Both methods make explicit and implicit assumptions about the structure of data to be protected, the nature of private information, and requirements for privacy, that do not hold for the majority of natural language data.

H. Brown, K. Lee, F. Mireshghallah, R. Shokri, and F. Tramèr (2022). **"What Does it Mean for a Language Model to Preserve Privacy?"** In: *2022 ACM Conference on Fairness, Accountability, and Transparency*. New York, NY, USA: ACM, pp. 2280–2292

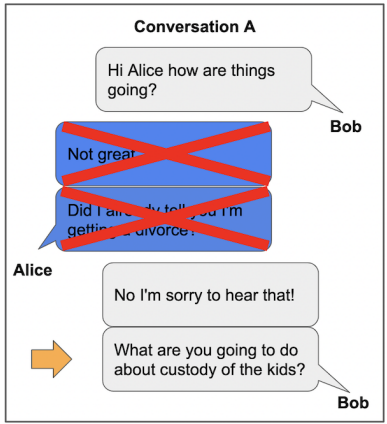
Private information in text?



H. Brown, K. Lee, F. Mireshghallah, R. Shokri, and F. Tramèr (2022). **"What Does it Mean for a Language Model to Preserve Privacy?"** In: *2022 ACM Conference on Fairness, Accountability, and Transparency*. New York, NY, USA: ACM, pp. 2280–2292

Figure 2: Original conversation. Private information indicated by orange arrows.

Private information in text?



H. Brown, K. Lee, F. Mireshghallah, R. Shokri, and F. Tramèr (2022). **"What Does it Mean for a Language Model to Preserve Privacy?"** In: *2022 ACM Conference on Fairness, Accountability, and Transparency*. New York, NY, USA: ACM, pp. 2280–2292

Figure 3: Alice's messages removed. Bob's last message still includes her private information.

Private information in text?

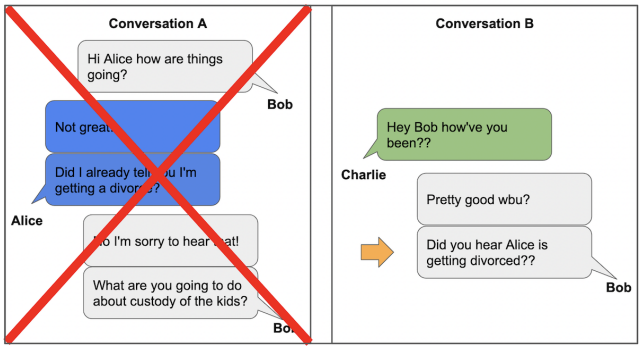


Figure 4: The whole original conversation is removed. Conversation B still contains Alice's private information though she is not in the conversation.

H. Brown, K. Lee, F. Mireshghallah, R. Shokri, and F. Tramèr (2022). **"What Does it Mean for a Language Model to Preserve Privacy?"** In: *2022 ACM Conference on Fairness, Accountability, and Transparency*. New York, NY, USA: ACM, pp. 2280–2292

What we covered so far

- Pure $(\varepsilon, 0)$ differential privacy
- Central and Local DP
- Approximate (ε, δ)-DP
- Mechanisms: Laplace, Exponential, Randomized response, Gaussian
- Post processing and composition
- Differentially-private Stochastic Gradient Descent
- Practical applications of DP-SGD in NLP

Machine unlearning

Why machine unlearning?[1]

To edit away undesired things from a (pre-trained) model, such as

- private data
- stale knowledge
- copyrighted materials
- toxic/unsafe content, dangerous capabilities, and misinformation

without retraining the model from scratch

K. Z. Liu (2024). **Machine unlearning in 2024.** URL: `https://ai.stanford.edu/~kzliu/blog/unlearning`. Blog

[1] This lecture is largely based on a great blog-post by Ken Ziyu Liu, `https://ai.stanford.edu/~kzliu/blog/unlearning`

What is machine unlearning?

K. Z. Liu (2024). **Machine unlearning in 2024.** URL: https://ai.stanford.edu/~kzliu/blog/unlearning. Blog

Removing the influences of particular training data from a trained model

Unlearning on a target model seeks to produce an **unlearned model** that is equivalent to (or at least 'behaves like') a **retrained model** that is trained on the **same data** of target model, **minus** the information to be unlearned

Many facets of machine unlearning

The precise definitions of unlearning, the techniques, the guarantees, and the metrics/evaluations would depend on:

- The ML task (e.g., binary classification or language modeling)
- The data to unlearn (e.g., a set of images, news articles, or the knowledge of making napalm)
- The unlearning algorithm (e.g., heuristic fine-tuning vs deleting model components)
- The goal of unlearning (e.g., for user privacy or harmfulness removal)

K. Z. Liu (2024). **Machine unlearning in 2024.** URL: https://ai.stanford.edu/~kzliu/blog/unlearning. Blog

Machine unlearning

Approaches to unlearning

Early papers on unlearning

Y. Cao and J. Yang (2015). **"Towards Making Systems Forget with Machine Unlearning".** In: *2015 IEEE Symposium on Security and Privacy*. San Jose, CA: IEEE, pp. 463–480

"To forget a piece of training data completely, systems need to revert the effects of the data on the extracted features and models."

Cao and J. Yang (2015) coined the term **machine unlearning**

What would be the easiest method for unlearning?

Naive approach to unlearning

Retrain the features and models **from scratch** after removing the data to be forgotten

Y. Cao and J. Yang (2015). **"Towards Making Systems Forget with Machine Unlearning".** In: *2015 IEEE Symposium on Security and Privacy*. San Jose, CA: IEEE, pp. 463–480

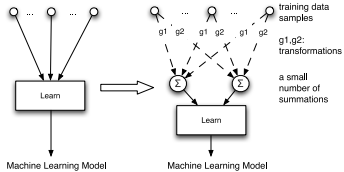
Pros:

- 100% provable: If the data points were not used for training, they cannot end up in the target model

Cons:

- Lack of scalability: When the training dataset is large, this approach is slow

Unlearning by Cao and J. Yang (2015) (part 1)

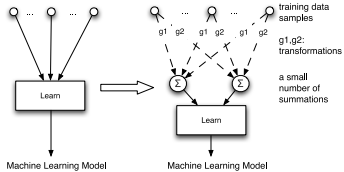


Y. Cao and J. Yang (2015). **"Towards Making Systems Forget with Machine Unlearning".** In: *2015 IEEE Symposium on Security and Privacy*. San Jose, CA: IEEE, pp. 463–480

- The model consists of a small number **summations**, the learning algorithm depends only on the summations, not individual data

Unlearning: Recompute only a small number of terms, asymptotically faster than retraining from scratch

Unlearning by Cao and J. Yang (2015) (part 2)



Y. Cao and J. Yang (2015). **"Towards Making Systems Forget with Machine Unlearning".** In: *2015 IEEE Symposium on Security and Privacy*. San Jose, CA: IEEE, pp. 463–480

- Summations are saved together with the trained model
- Each summation is the sum of some efficiently computable transformation of the training data samples

Unlearning: Recompute only a small number of terms, asymptotically faster than retraining from scratch

Limitations of unlearning by Cao and J. Yang (2015)

A. Sekhari, J. Acharya, G. Kamath, and A. T. Suresh (2021). **"Remember What You Want to Forget: Algorithms for Machine Unlearning".** In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan. Vol. 34. Curran Associates, Inc., pp. 18075–18086

- Algorithm limited to very structured problems only (Sekhari, Acharya, Kamath, and Suresh, 2021)

Machine unlearning

Sharded, Isolated, Sliced, and Aggregated (SISA) training

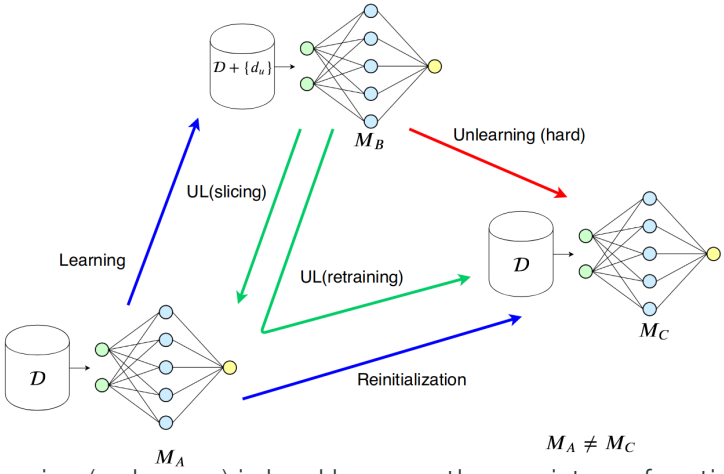
Unlearning versus Differential Privacy

ML models potentially memorize training data → important to unlearn what they have learned from data to be deleted.

- Enforcing ε-differential privacy with $\varepsilon \neq 0$ does **not** alleviate the need for an unlearning mechanism
- While DP algorithms guarantee a bound on how much individual training points contribute to the model, there is a **non-zero** contribution from each point (if this was not the case, the model would not be able to learn at all)

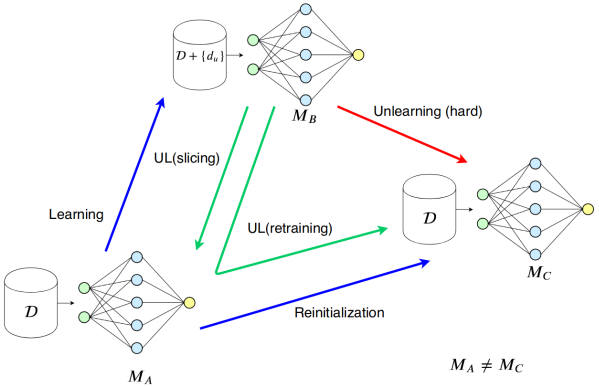
We require that a point has **no influence** on the model once it has been unlearned

L. Bourtoule, V. Chandrasekaran, C. A. Choquette-Choo, H. Jia, A. Travers, B. Zhang, D. Lie, and N. Papernot (2021). **"Machine Unlearning".** In: *2021 IEEE Symposium on Security and Privacy (SP)*. San Francisco, CA, USA: IEEE, pp. 141–159



L. Bourtoule, V. Chandrasekaran, C. A. Choquette-Choo, H. Jia, A. Travers, B. Zhang, D. Lie, and N. Papernot (2021). **"Machine Unlearning".** In: *2021 IEEE Symposium on Security and Privacy (SP)*. San Francisco, CA, USA: IEEE, pp. 141–159

Unlearning (red arrow) is hard because there exists no function that measures the influence of augmenting the dataset $\mathcal{D}$ with point d_u and fine-tuning a model M_A already trained on $\mathcal{D}$ to train (left blue arrow) a model M_B for $\mathcal{D} + \{d_u\}$.



L. Bourtoule, V. Chandrasekaran, C. A. Choquette-Choo, H. Jia, A. Travers, B. Zhang, D. Lie, and N. Papernot (2021). **"Machine Unlearning".** In: *2021 IEEE Symposium on Security and Privacy (SP)*. San Francisco, CA, USA: IEEE, pp. 141–159

This makes it impossible to revert to model M_A without saving its parameter state before learning about d_u. We call this model slicing (short green arrow). In the absence of slicing, one must retrain (curved green arrow) the model without d_u, resulting in a model M_C that is different from the original model M_A.

SISA training approach

Retraining from scratch while omitting data points that need to be unlearned: most straightforward way, provable guarantees

SISA training replicates the model being learned several times where each replica receives a disjoint shard (or subset) of the dataset—similar to current distributed training strategies. We refer to each replica as a constituent model.

SISA training deviates from other strategies: there is no flow of information between constituent models

L. Bourtoule, V. Chandrasekaran, C. A. Choquette-Choo, H. Jia, A. Travers, B. Zhang, D. Lie, and N. Papernot (2021). **"Machine Unlearning".** In: *2021 IEEE Symposium on Security and Privacy (SP)*. San Francisco, CA, USA: IEEE, pp. 141–159

SISA training approach (contd.)

Each shard is further partitioned into slices, where each constituent model is trained incrementally (and iteratively, in a stateful manner) with an increasing number of slices.

At inference, the test point is fed to each constituent and all the constituents' responses are aggregated, similar to the case of ML ensembles

When a data point is to be unlearned, only the constituent model whose dataset contains this point is affected

L. Bourtoule, V. Chandrasekaran, C. A. Choquette-Choo, H. Jia, A. Travers, B. Zhang, D. Lie, and N. Papernot (2021). **"Machine Unlearning".** In: *2021 IEEE Symposium on Security and Privacy (SP)*. San Francisco, CA, USA: IEEE, pp. 141–159

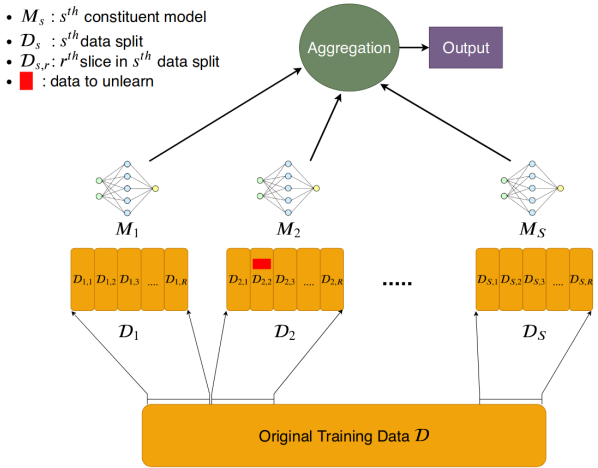
SISA training approach (contd.)

L. Bourtoule, V. Chandrasekaran, C. A. Choquette-Choo, H. Jia, A. Travers, B. Zhang, D. Lie, and N. Papernot (2021). **"Machine Unlearning".** In: *2021 IEEE Symposium on Security and Privacy (SP)*. San Francisco, CA, USA: IEEE, pp. 141–159

Sharding is possible for any model or hypothesis class: it has no impact on how training is performed beyond the smaller set of data each model has access to

Slicing is possible for any iterative learning algorithm that is stateful: the algorithm should be such that it can continue to learn from its current state when presented with new data. Gradient descent naturally falls under that category

SISA training approach (contd.)



L. Bourtoule, V. Chandrasekaran, C. A. Choquette-Choo, H. Jia, A. Travers, B. Zhang, D. Lie, and N. Papernot (2021). **"Machine Unlearning".** In: *2021 IEEE Symposium on Security and Privacy (SP)*. San Francisco, CA, USA: IEEE, pp. 141–159

SISA inference and testing

S. Greengard (2022). **"Can AI learn to forget?"** In: *Communications of the ACM* 65.4, pp. 9–11

Inference: a voting strategy where each constituent contributes equally to the final outcome through a simple label-based majority vote

SISA framework tested on more than a million images; typical speed improvements ranged from 2.45 to 4.63-times for unlearning tasks (Greengard, 2022)

SISA limitations

S. Greengard (2022). **"Can AI learn to forget?"** In: *Communications of the ACM* 65.4, pp. 9–11

While the SISA concept is promising, it has limitations:

- For example, by reducing the amount of data per shard, there is an impact on machine learning and a lower-quality outcome is likely

SISA limitations

SISA = model checkpointing, in which the learner **preemptively stores backup models** in which certain points have been excluded

- This strategy makes it easy to quickly return an appropriate backup model upon receiving a deletion request
- The downside is that one typically has to store a number of additional models which scales with the training data size, which may be **prohibitively large**

A. Sekhari, J. Acharya, G. Kamath, and A. T. Suresh (2021). **"Remember What You Want to Forget: Algorithms for Machine Unlearning".** In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan. Vol. 34. Curran Associates, Inc., pp. 18075–18086

Machine unlearning

Unlearning with probabilistic theoretical guarantees

Unlearning without storing the original data

Suppose a learning algorithm A over dataset S outputs the model $A(S)$

An unlearning algorithm $\bar{A}$ takes as input the model $A(S)$ and a set $U \subseteq S$ of data samples that are to be deleted, and is required to output a new model $\tilde{w}$

The unlearning algorithm $\bar{A}$ can also access some additional data statistics $T(S)$

$$\bar{A}(U, A(S), T(S)) \mapsto \tilde{w} \in W$$

The unlearning algorithm does not have access to the entire original dataset S and hence cannot retrain from scratch

A. Sekhari, J. Acharya, G. Kamath, and A. T. Suresh (2021). **"Remember What You Want to Forget: Algorithms for Machine Unlearning".** In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan. Vol. 34. Curran Associates, Inc., pp. 18075–18086

Formalizing unlearning

$$\bar{A}(U, A(S), T(S)) \mapsto \tilde{w} \in W$$

Probability of the unlearning algorithm outputting a certain unlearned model:

$$\Pr\big(\bar{A}(U, A(S), T(S)) \in W\big)$$

$S \setminus U$ is the original dataset S after removing samples to be unlearned U

Probability of the unlearning algorithm outputting a certain unlearned model (but this time the original model and statistics were trained on $S \setminus U$):

$$\Pr\big(\bar{A}(\emptyset, A(S \setminus U), T(S \setminus U)) \in W\big)$$

A. Sekhari, J. Acharya, G. Kamath, and A. T. Suresh (2021). **"Remember What You Want to Forget: Algorithms for Machine Unlearning".** In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan. Vol. 34. Curran Associates, Inc., pp. 18075–18086

ε, δ-unlearning by Sekhari, Acharya, Kamath, and Suresh (2021)

For $0 \leq \varepsilon \leq 1$ and $\delta > 0$:

$$\Pr\big(\bar{A}(U, A(S), T(S)) \in W\big) \leq \exp(\varepsilon) \cdot \Pr\big(\bar{A}(\emptyset, A(S \setminus U), T(S \setminus U)) \in W\big) + \delta$$

and

$$\Pr\big(\bar{A}(\emptyset, A(S \setminus U), T(S \setminus U)) \in W\big) \leq \exp(\varepsilon) \cdot \Pr\big(\bar{A}(U, A(S), T(S)) \in W\big) + \delta$$

The above states that with high probability, an observer cannot differentiate between the two cases

A. Sekhari, J. Acharya, G. Kamath, and A. T. Suresh (2021). **"Remember What You Want to Forget: Algorithms for Machine Unlearning".** In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan. Vol. 34. Curran Associates, Inc., pp. 18075–18086

ε, δ-unlearning by Sekhari, Acharya, Kamath, and Suresh (2021)

A. Sekhari, J. Acharya, G. Kamath, and A. T. Suresh (2021). **"Remember What You Want to Forget: Algorithms for Machine Unlearning".** In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan. Vol. 34. Curran Associates, Inc., pp. 18075–18086

Retraining from scratch would mean

$$\bar{A}(U, A(S), T(S)) = A(S \setminus U)$$

but it would require $T(S)$ contain the entire dataset S

Summary: Sekhari, Acharya, Kamath, and Suresh (2021) proposed an unlearning algorithm for convex loss functions

Machine unlearning

Examples of Machine Unlearning in NLP

Adapting SISA

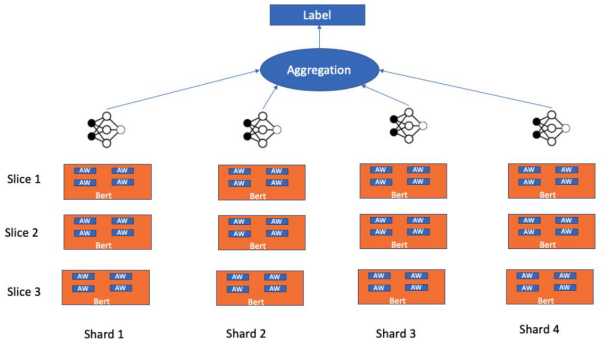


Figure 1: Architecture of SISA-A which uses S slices and R shards. One model is built for each shard and the labels are aggregated using a majority voting strategy.

V. Bannihatti Kumar, R. Gangadharaiah, and D. Roth (2023). **"Privacy Adhering Machine Un-learning in NLP".** In: *Findings of the Association for Computational Linguistics: IJCNLP-AACL 2023 (Findings).* Nusa Dua, Bali: Association for Computational Linguistics, pp. 268–277

The model goes through the data, slice by slice saving a checkpoint after training for each slice in each shard

Adapting SISA

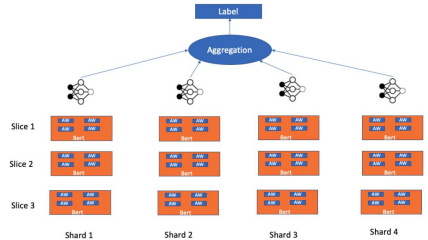


Figure 1: Architecture of SISA-A which uses S slices and R shards. One model is built for each shard and the labels are aggregated using a majority voting strategy.

V. Bannihatti Kumar, R. Gangadharaiah, and D. Roth (2023). **"Privacy Adhering Machine Un-learning in NLP".** In: *Findings of the Association for Computational Linguistics: IJCNLP-AACL 2023 (Findings).* Nusa Dua, Bali: Association for Computational Linguistics, pp. 268–277

Un-learning request: pick the shard and the slide where this data point is present

Delete the datapoint from this slice, take the previous checkpoint, continue training

Adapting SISA: Choice of checkpoints

SISA-FC

- Fine-tune only the final fully-connected (FC) layers of BERT

Adapters (Houlsby, Giurgiu, Jastrzebski, Morrone, De Laroussilhe, Gesmundo, Attariyan, and Gelly, 2019)

- Adapters in the Encoder blocks of the transformer. This increases the memory footprint of the model, it only accounts for about 1–5% of the model parameters (95–99% memory benefits)

V. Bannihatti Kumar, R. Gangadharaiah, and D. Roth (2023). **"Privacy Adhering Machine Un-learning in NLP".** In: *Findings of the Association for Computational Linguistics: IJCNLP-AACL 2023 (Findings).* Nusa Dua, Bali: Association for Computational Linguistics, pp. 268–277

N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly (2019). **"Parameter-efficient transfer learning for NLP".** In: *Proceedings of the 36th international conference on machine learning.* Ed. by K. Chaudhuri and R. Salakhutdinov. Vol. 97. Proceedings of machine learning research. PMLR, pp. 2790–2799

Another approach to unlearning in transformers

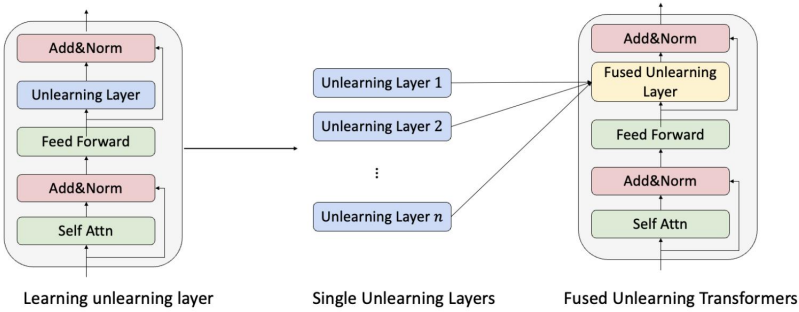
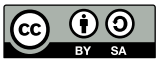


Figure 1: Overall process of our EUL framework. The unlearning layers are plugged into transformer layers after the feed-forward networks. During training, only the unlearning layers are learned to forget requested data while the original LLMs remain unchanged. For every deletion request, an unlearning layer is learned first and then merged with other unlearning layers via our designed fusion mechanism to form the fused unlearning transformer which satisfies a series of deletion requests.

J. Chen and D. Yang (2023). **"Unlearn What You Want to Forget: Efficient Unlearning for LLMs".** In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Singapore: Association for Computational Linguistics, pp. 12041–12052

License and credits



Credits

Ivan Habernal

Content from ACL Anthology papers licensed under CC-BY
`https://www.aclweb.org/anthology`

Partly inspired by lectures from Gautam Kamath