DP-NMT: Scalable Differentially-Private Machine Translation

Timour Igamberdiev[1] **Doan Nam Long Vu**[1] **Felix Künnecke**[1]
Zhuo Yu[1] **Jannik Holmer**[1] **Ivan Habernal**[2]
Trustworthy Human Language Technologies
[1] Department of Computer Science, Technical University of Darmstadt
[2] Department of Computer Science, Paderborn University
timour.igamberdiev@tu-darmstadt.de
www.trusthlt.org

Abstract

Neural machine translation (NMT) is a widely popular text generation task, yet there is a considerable research gap in the development of privacy-preserving NMT models, despite significant data privacy concerns for NMT systems. Differentially private stochastic gradient descent (DP-SGD) is a popular method for training machine learning models with concrete privacy guarantees; however, the implementation specifics of training a model with DP-SGD are not always clarified in existing models, with differing software libraries used and code bases not always being public, leading to reproducibility issues. To tackle this, we introduce `DP-NMT`, an open-source framework for carrying out research on privacy-preserving NMT with DP-SGD, bringing together numerous models, datasets, and evaluation metrics in one systematic software package. Our goal is to provide a platform for researchers to advance the development of privacy-preserving NMT systems, keeping the specific details of the DP-SGD algorithm transparent and intuitive to implement. We run a set of experiments on datasets from both general and privacy-related domains to demonstrate our framework in use. We make our framework publicly available and welcome feedback from the community.[1]

1 Introduction

Privacy-preserving natural language processing (NLP) has been a recently growing field, in large part due to an increasing amount of concern regarding data privacy. This is especially a concern in the context of modern neural networks memorizing training data that may contain sensitive information (Carlini et al., 2021). While there has been a body of research investigating privacy for text classification tasks (Senge et al., 2022) and language models (Hoory et al., 2021; Anil et al., 2022), there has not been as much focus on text generation tasks, in particular neural machine translation (NMT). However, NMT is particularly worrying from a privacy perspective, due to a variety of machine translation services available online that users send their personal data to. This includes built-in NMT services to existing websites, e-mail clients, and search engines. After data has been sent to these systems, it may be further processed and used in the development of the NMT system (Kamocki and O'Regan, 2016), which has a significant risk of being memorized if trained in a non-private manner.

One of the most popular methods for tackling this privacy issue is differential privacy (DP), being a formal framework which provides probabilistic guarantees that the contribution of any single data point to some analysis is bounded. In the case of NLP and machine learning (ML), this means that a data point associated with some individual which is included in the model's training data cannot stand out 'too much' in the learning process of the model.

The DP-SGD algorithm (Abadi et al., 2016b) is one of the most standard methods to achieve this for ML systems, yet implementations of DP-SGD often lack some technical details on the specifics of the algorithm. In particular, this includes the privacy amplification method assumed for calculating the privacy budget ε when composed over all training iterations of the model. This means that the exact *strength of the privacy protection* that the resulting systems provide is not clear, with the 'standard' **random shuffling** method for iterating over batches providing a weaker privacy guarantee for the training data than **Poisson sampling**. With different implementations using different software libraries, the community currently does not have a consistent platform for conducting experiments for scalable differentially private systems, such as NMT.

To tackle this problem, we develop a modular framework for conducting research on private NMT in a transparent and reproducible manner. Our pri-

[1] https://github.com/trusthlt/dp-nmt

Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics
System Demonstrations, pages 94–105

mary goal is to allow for a deeper investigation into the applications of DP for NMT, all while ensuring that important theoretical details of the DP-SGD methodology are properly reflected in the implementation. Following previous work on DP-SGD (Subramani et al., 2021; Anil et al., 2022), we implement our framework in the JAX library (Bradbury et al., 2018), which provides powerful tools that help to reduce the significant computational overhead of DP-SGD, allowing for scalability in implementing larger systems and more extended training regimes.

Our primary contributions are as follows. First, we present `DP-NMT`, a framework developed in JAX for leading research on NMT with DP-SGD. It includes a growing list of available NMT models, different evaluation schemes, as well as numerous datasets available out of the box, including standard datasets used for NMT research and more specific privacy-related domains. Second, we demonstrate our framework by running experiments on these NMT datasets, providing one of the first investigations into privacy-preserving NMT. Importantly, we compare the random shuffling and Poisson sampling methods for iterating over training data when using DP-SGD. We demonstrate that, in addition to the theoretical privacy guarantee, there may indeed be differences in the model performance when utilizing each of the two settings.

2 DP-SGD and subsampling

We describe the main ideas of differential privacy (DP) and DP-SGD in Appendix A. We refer to Abadi et al. (2016b); Igamberdiev and Habernal (2022); Habernal (2021, 2022); Hu et al. (2024) for a more comprehensive explanation.

A key aspect of the DP-SGD algorithm (see Alg. 1 in the Appendix) is **privacy amplification by subsampling**, in which a stronger privacy guarantee can be obtained for a given dataset x when a subset of this dataset is first randomly sampled (Kasiviswanathan et al., 2011; Beimel et al., 2014). If the sampling probability is q, then the overall privacy guarantee can be analyzed as being approximately $q\varepsilon$.

A key point here is the nature of this sampling procedure and the resulting privacy guarantee. The moments accountant of Abadi et al. (2016b), which is an improvement on the strong composition theorem (Dwork et al., 2010) for composing multiple DP mechanisms, assumes Poisson sampling. Under this procedure, *each data point* is included in a mini-batch with probability $q = L/N$, with L being the *lot size* and N the size of the dataset. An alternative method to Poisson sampling is uniform sampling, in which mini-batches of a fixed size are independently drawn at each training iteration (Wang et al., 2019; Balle et al., 2018).

In practice, however, many modern implementations of DP-SGD utilize **random shuffling**, with the dataset split into fixed-size mini-batches. Several training iterations thus form an epoch, in which each training data point appears exactly once, in contrast to Poisson sampling for which the original notion of 'epoch' is not quite suitable, since each data point can appear in any training iteration and there is no "single passing of the training data through the model". In Abadi et al. (2016b), the term *epoch* is redefined as $\frac{N}{L}$ lots, being essentially an expectation of the number of batches when utilizing N data points for training the model. While simply shuffling the dataset can indeed result in privacy amplification (Erlingsson et al., 2019; Feldman et al., 2022), the nature of the corresponding privacy guarantee is **not the same** as the guarantee achieved by Poisson sampling, generally being weaker. We refer to Ponomareva et al. (2023, Section 4.3) for further details.

3 Related work

3.1 Applications of DP-SGD to NLP

The application of DP-SGD to the field of NLP has seen an increasing amount of attention in recent years. A large part of these studies focus on differentially private pre-training or fine-tuning of language models (Hoory et al., 2021; Yu et al., 2021; Basu et al., 2021; Xu et al., 2021; Anil et al., 2022; Ponomareva et al., 2022; Shi et al., 2022; Wu et al., 2022; Li et al., 2022; Yin and Habernal, 2022; Mattern et al., 2022; Hansen et al., 2022; Senge et al., 2022). A primary goal is to reach the best possible privacy/utility trade-off for the trained models, in which the highest performance is achieved with the strictest privacy guarantees.

In the general machine learning setting, the exact sampling method that is used for selecting batches at each training iteration is often omitted, since this is generally not a core detail of the training methodology. Possibly for this reason, in the case of privately training a model with DP-SGD, the sampling method is also often not mentioned. However, in contrast to the non-private setting, here **sampling**

is actually a core detail of the algorithm, which has an **impact on the privacy accounting procedure**. In the case that experimental descriptions with DP-SGD include mentions of *epochs* without further clarification, this in fact suggests the use of the random shuffling scheme, as opposed to Poisson sampling, as described in Section 2. In addition, sometimes the code base is not publicly available, in which case it is not possible to validate the sampling scheme used.

Finally, standard implementations of DP-SGD in the Opacus (Yousefpour et al., 2021) and TensorFlow Privacy (Abadi et al., 2016a) libraries often include descriptions of DP-SGD implementations with randomly shuffled fixed-size batches. For instance, while Opacus currently has a `DPDataLoader` class which by default uses their `UniformWithReplacementSampler` class for facilitating the use of Poisson sampling, some of the tutorials currently offered appear to also use static batches instead.[2] A similar situation is true for TensorFlow Privacy.[3] While these libraries support per-example gradients as well, several core features of JAX make it the fastest and most scalable option for implementing DP-SGD (Subramani et al., 2021), described in more detail below in Section 4.

We therefore stress the importance of clarifying implementation details that may not be as vital in the general machine learning setting, but are very relevant in the private setting. As described by Ponomareva et al. (2023), it is an open theoretical question as to how random shuffling and Poisson sampling differ with respect to privacy amplification gains, with known privacy guarantees being weaker for the former.

3.2 Private neural machine translation

The task of private neural machine translation remains largely unexplored, with currently no studies we could find that incorporate DP-SGD to an NMT system. Wang et al. (2021) investigate NMT in a federated learning setup (McMahan et al., 2017), with differential privacy included in the aggregation of parameters from each local model, adding Laplace noise to these parameters. Several other studies explore NMT with federated learning, but do not incorporate differential privacy in the methodology (Roosta et al., 2021; Passban et al., 2022; Du et al., 2022). Hisamoto et al. (2020), applied a membership inference attack (Shokri et al., 2017) on a 6-layer Transformer (Vaswani et al., 2017) model in the scenario of NMT as a service, with the goal of clients being able to verify whether their data was used to train an NMT model. Finally, Kamocki and O'Regan (2016) address the general topic of privacy issues for machine translation as a service. The authors examine how these MT services fit European data protection laws, noting the legal nature of various types of data processing that can occur by both the provider of such a service, as well as by the users themselves.

4 Description of software

The aim of our system is to offer a reliable and scalable approach to achieve differentially private machine translation. Figure 1 illustrates the central structure of our system. The user can upload a translation dataset that is either accessible on the HuggingFace Datasets Hub[4] or is provided by us out of the box, and integrate it seamlessly for both training and efficient privacy accounting, utilizing HuggingFace's Datasets library (Lhoest et al., 2021).

Accelerated DP-SGD with JAX and Flax Our goal is to accelerate DP-SGD training through the use of a Transformer model implemented with JAX and Flax (Bradbury et al., 2018; Heek et al., 2023). The speed of training DP-SGD in the framework can be considerably enhanced through vectorization, just-in-time (JIT) compilation, and static graph optimization (Subramani et al., 2021). JIT compilation and automatic differentiation are defined and established on the XLA compiler. JAX's main transformation methods of interest for fast DP-SGD are `grad`, `vmap`, and `pmap`, offering the ability to mix these operations as needed (Yin and Habernal, 2022). In the DP-SGD scenario, combining `grad` and `vmap` facilitates efficient computation of per-example gradients by vectorizing the gradient calculation along the batch dimension (Anil et al., 2022). Additionally, our training step is decorated by `pmap` to leverage the XLA compiler on multiple GPUs, significantly accelerating training speed. The framework offers to conduct experiments with multiple encoder-decoder models

[2] https://opacus.ai/tutorials/building_image_classifier.

[3] https://www.tensorflow.org/responsible_ai/privacy/tutorials/classification_privacy.

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

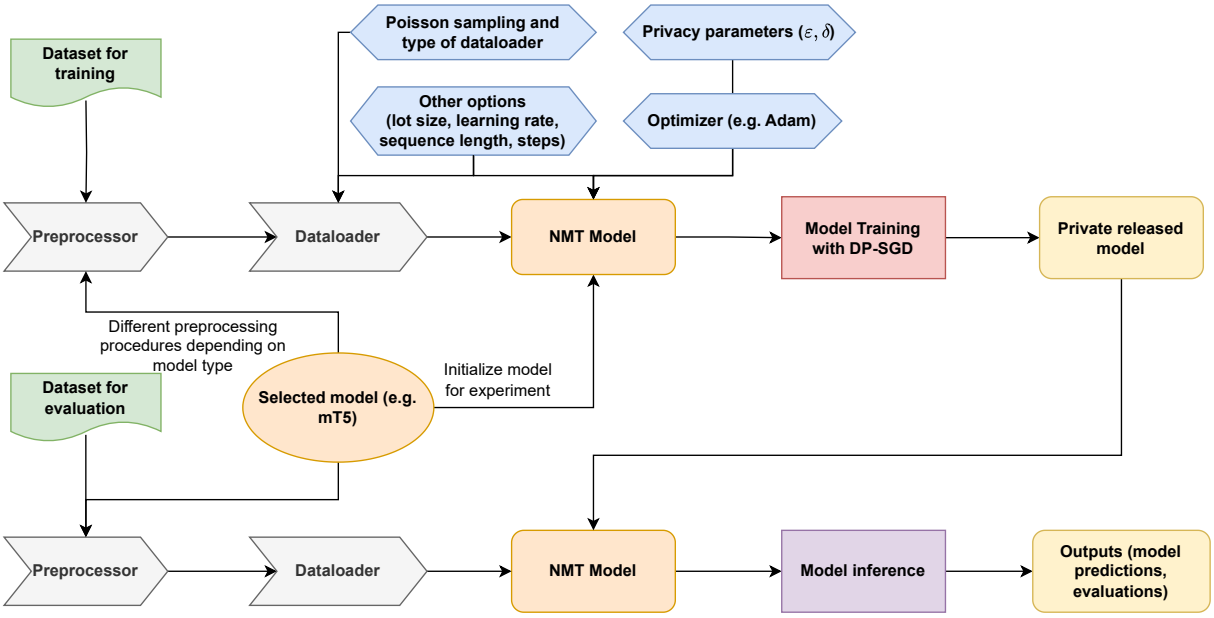

Figure 1: Framework Pipeline. Similar components are represented with different colors. Green: Dataset selection. Blue: Experimental configurations (including privacy settings). Grey: Dataset preparation. Orange: Model-specific elements. Red: Model training. Purple: Model inference. Yellow: Output of experiments.

and integrate new seq2seq models, in addition to existing ones, such as mBART (Liu et al., 2020), T5 (Raffel et al., 2020), and mT5 (Xue et al., 2021). When selecting a model, the corresponding preprocessor will prepare the dataset accordingly. This allows the software to be flexible and modular, enabling researchers to exchange models and datasets to perform a range of private NMT experiments.

Model training and inference The experimental workflow of our framework works in two phases, namely model training and model inference. For both phases, the process begins with a data loader that can be either a framework-provided dataset or a user-specified dataset. Subsequently, the loaded dataset is prepared based on user-defined parameters, including standard options (e.g. sequence length), as well as parameters relating to DP-SGD (e.g. data loader type, sampling method, and batch size). After selecting the model, the user separates it into different procedures according to the model type. Subsequently, the model is initiated, optionally from a checkpoint that has already been trained. Then, the primary experiment is carried out based on the specified mode, which includes (1) fine-tuning on an existing dataset, (2) using an existing fine-tuned checkpoint to continue fine-tuning on the dataset, or (3) inference without teacher forcing.

Integrating `DPDataloader` from Opacus One notable improvement in our software is the incorporation of the `DPDataloader` from Opacus (Yousefpour et al., 2021) for out-of-the-box Poisson sampling. This is different from the existing approaches in JAX used by Yin and Habernal (2022); Subramani et al. (2021); Ponomareva et al. (2022), who employ iteration over a randomly shuffled dataset, which theoretically provides weaker DP bounds. Evaluation metrics such as BLEU (Papineni et al., 2002) and BERTScore (Zhang et al., 2020) are available for each mode. We incorporate the differential privacy component during the training phase of the systems.

Engineering challenges for LLMs Throughout development, we encountered multiple engineering challenges. Initially, our academic budget limitations made it difficult to train a larger model due to the significant memory consumption during per-example gradient calculations. Consequently, we anticipated a relatively small physical batch size on each GPU. We attempted to freeze parts of the model for faster training and improved memory efficiency, as Senge et al. (2022) noted. However, in Flax, the freezing mechanism only occurs during the optimization step and does not affect per-example gradient computation. Therefore, it does not solve the issue of limited physical batch sizes. Multiple reports suggest that increasing the

lot size leads to better DP-SGD performance due to an improved gradient signal-to-noise ratio and an increased likelihood of non-duplicated example sampling across the entire dataset (Hoory et al., 2021; Yin and Habernal, 2022; Anil et al., 2022). However, compared to previous work on large models that mostly relied on dataset iteration (Yin and Habernal, 2022; Ponomareva et al., 2022), implementing the original DP-SGD with large lots using Poisson sampling, a large language model (LLM) with millions of parameters, and on multiple GPUs presents a challenge that makes comparison difficult. To address this issue, we first conduct a sampling process on a large dataset, then divide it into smaller subsets that the GPU can handle. We then build up the large lot using gradient accumulation. It is crucial that we refrain from implementing any additional normalization operations that might change the gradient sensitivity (Ponomareva et al., 2023; Hoory et al., 2021), prior to the noise addition step.

5 Experiments

To demonstrate our framework in use, fill the gaps on current knowledge of the privacy/utility trade-off for the task of NMT, as well as examine the effects of using random shuffling vs. Poisson sampling, we run a series of experiments with DP-SGD on several NMT datasets, using a variety of privacy budgets.

5.1 Datasets

We utilize datasets comprising two main types of settings. The first is the general NMT setting for comparing our models with previous work and investigating the effectiveness of DP-SGD on a common NMT dataset. For this we utilize WMT-16 (Bojar et al., 2016), using the German-English (DE-EN) language pair as the focus of our experiments.

The second setting is the more specific target domain of private texts that we are aiming to protect with differentially private NMT. For the sake of reproducibility and ethical considerations, we utilize datasets that *imitate* the actual private setting of processing sensitive information, namely business communications and medical notes, but are themselves publicly available. The first dataset is the Business Scene Dialogue corpus (BSD) (Rikters et al., 2019), which is a collection of fictional business conversations in various scenarios (e.g. "face-to-face", "phone call", "meeting"), with parallel data for Japanese and English. While the original corpus consists of half English → Japanese and half Japanese → English scenarios, we combine both into a single Japanese → English (JA-EN) language pair for our experiments.

The second dataset is ClinSPEn-CC (Neves et al., 2022), which is a collection of parallel COVID-19 clinical cases in English and Spanish, originally part of the biomedical translation task of WMT-22. We utilize this corpus in the Spanish → English (ES-EN) direction. These latter two datasets simulate a realistic scenario where a company or public authority may train an NMT model on private data, for later public use. We present overall statistics for each dataset in Table 1.

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

Table 1: Dataset statistics. Trn.: Train, Vld.: Validation.

5.2 Experimental setup

For each of the above three datasets, we fine-tune a pre-trained mT5 model (Xue et al., 2021), opting for the `mT5-small`[5] version due to computational capacity limitations described in Section 4. We compare ε values of $\infty, 1000, 5$, and 1, representing the non-private, weakly private, moderately private, and very private scenarios, respectively (see Lee and Clifton (2011); Hsu et al. (2014); Weiss et al. (2023) for a more detailed discussion on selecting the 'right' ε value). We fix the value of δ to 10^{-8} for all experiments, staying well below the recommended $\delta \ll \frac{1}{N}$ condition (Abadi et al., 2016b).

For all of the above configurations, we compare two methods of selecting batches of data points from the dataset for our DP-SGD configurations, namely **random shuffling** and **Poisson sampling**. Following previous work (Hoory et al., 2021; Anil et al., 2022; Yin and Habernal, 2022), we utilize very large batch sizes for both of these methods, setting L to a large value and building up the resulting drawn batches with gradient accumulation for the latter method, as described in Section 4. We refer to Appendix B for a more detailed description of our hyperparameter search. We evaluate our

[5] https://huggingface.co/google/mt5-small

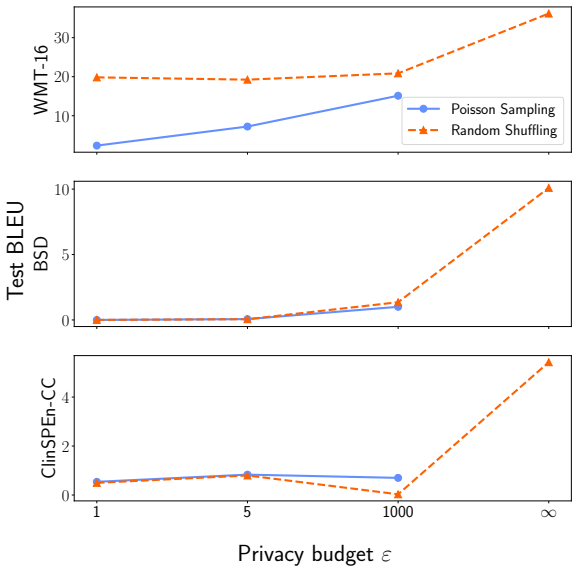

Figure 2: Test BLEU scores for each of the three datasets using varying privacy budgets, comparing the random shuffling and Poisson sampling methods to iterate over the dataset. Non-private results are additionally shown for each dataset ($\varepsilon = \infty$) with random shuffling. Lower ε corresponds to a stronger privacy guarantee.

model outputs using BLEU (Papineni et al., 2002), and BERTScore (Zhang et al., 2020) metrics.

5.3 Results and Discussion

Figure 2 shows the results of our experiments, reporting BLEU scores on the test partition of each dataset.

Privacy/utility trade-off We verify the soundness of our models in the non-private setting ($\varepsilon = \infty$) by comparing with past non-private results, particularly for the commonly used WMT-16 dataset. For WMT-16 DE-EN, we reach a BLEU score of 36.2, being similar to past models (e.g. Wei et al. (2021) obtain a BLEU score of 38.6 using their 137B parameter FLAN model). In the case of BSD and ClinSPEn-CC, these datasets are not as 'standard' within the NMT community, and therefore have a more limited chance for comparison.

For private results, we can see a clear difference between the drop in WMT-16 performance vs. that of BSD and ClinSPEn-CC. This is not at all surprising, given that the latter two datasets are vastly smaller in comparison to WMT-16, making it far more difficult to train an NMT model, particularly in the noisy setting of DP-SGD. In addition, ClinSPEn-CC contains a large amount of complicated medical terminology that adds an extra layer of difficulty for a model. We therefore need to conduct further investigations into applications of DP-SGD to very small datasets in order to reach more meaningful ε values.

Method of dataset iteration When comparing random shuffling with Poisson sampling, we can see practically no difference for BSD and ClinSPEn-CC, most likely due to the low DP-SGD results for these two datasets. The differences are more notable for WMT-16, where there is a clear gap between the two sets of configurations. For instance, at $\varepsilon = 1$, WMT-16 shows a BLEU score of 19.83 when using random shuffling, in contrast to 2.35 with Poisson sampling. The latter method therefore shows a far greater drop from the non-private setting, improving more gradually as ε is increased.

There are several possible explanations for this. With Poisson sampling, while each data point has an equal probability of being drawn to make up a particular batch, it is possible that some data points end up being drawn more frequently than others for several training iterations. This may have an impact on the model learning process, possibly missing out on the signal from certain useful data points at various stages of training. Another reason may be that we simply require additional hyperparameter optimization with Poisson sampling, expanding the search space further.

6 Conclusion

We have introduced `DP-NMT`, a modular framework developed using the JAX library, with the goal of leading research on neural machine translation with DP-SGD. To demonstrate our framework in use, we have presented several experiments on both general and privacy-related NMT datasets, comparing two separate approaches for iterating over training data with DP-SGD, and facilitating in filling the research gap on the privacy/utility trade-off in this task. We are continuing to actively expand the framework, including the integration of new models and NMT datasets. We hope that our framework will help to expand research into privacy-preserving NMT and welcome feedback from the community.

Ethics and Limitations

An important ethical consideration with regards to our framework is its intended use. We strive to further the field of private NMT and improve

the current knowledge on how to effectively apply differential privacy to data used in NMT systems. However, applications of differential privacy to textual data are still at an early research stage, and **should not currently be used in actual services that handle real sensitive data of individuals.**

The primary reason for this is that our understanding of what is *private information* in textual data is still very limited. Applications of differential privacy in the machine learning setting provide a privacy guarantee to each individual *data point*. In the context of DP-SGD, this means that if any single data point is removed from the dataset, the impact on the resulting model parameter update is bounded by the provided multiplicative guarantee in Eqn. 1. In other words, it does not stand out 'too much' in its contribution to training the model.

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 and identically distributed (i.i.d.) in the DP setting. Thus, we require further research on how to properly apply a privacy guarantee to individuals represented within a textual dataset. We refer to Klymenko et al. (2022); Brown et al. (2022); Igamberdiev and Habernal (2023) for a more comprehensive discussion on this.

Acknowledgements

This project was supported by the PrivaLingo research grant (Hessisches Ministerium des Innern und für Sport). The independent research group TrustHLT is supported by the Hessian Ministry of Higher Education, Research, Science and the Arts. Thanks to Luke Bates for helpful feedback on a preliminary draft.

References

Martin Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2016a. Tensorflow: A system for large-scale machine learning. In *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation*, OSDI'16, page 265–283, USA. USENIX Association.

Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016b. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318.

Rohan Anil, Badih Ghazi, Vineet Gupta, Ravi Kumar, and Pasin Manurangsi. 2022. Large-scale differentially private BERT. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6481–6491, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Borja Balle, Gilles Barthe, and Marco Gaboardi. 2018. Privacy amplification by subsampling: Tight analyses via couplings and divergences. *Advances in neural information processing systems*, 31.

Priyam Basu, Tiasa Singha Roy, Rakshit Naidu, and Zumrut Muftuoglu. 2021. Privacy enabled financial text classification using differential privacy and federated learning. In *Proceedings of the Third Workshop on Economics and Natural Language Processing*, pages 50–55, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Amos Beimel, Hai Brenner, Shiva Prasad Kasiviswanathan, and Kobbi Nissim. 2014. Bounds on the sample complexity for private learning and private data release. *Machine learning*, 94:401–437.

Ondrej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, et al. 2016. Findings of the 2016 conference on machine translation (wmt16). In *First conference on machine translation*, pages 131–198. Association for Computational Linguistics.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. 2018. JAX: composable transformations of Python+NumPy programs. `http://github.com/google/jax`.

Hannah Brown, Katherine Lee, Fatemehsadat Mireshghallah, Reza Shokri, and Florian Tramèr. 2022. What does it mean for a language model to preserve privacy? In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 2280–2292.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. 2021. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650.

Yichao Du, Zhirui Zhang, Bingzhe Wu, Lemao Liu, Tong Xu, and Enhong Chen. 2022. Federated nearest neighbor machine translation. In *The Eleventh International Conference on Learning Representations*.

Cynthia Dwork and Aaron Roth. 2013. The Algorithmic Foundations of Differential Privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3-4):211–407.

Cynthia Dwork, Guy N Rothblum, and Salil Vadhan. 2010. Boosting and differential privacy. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 51–60. IEEE.

Úlfar Erlingsson, Vitaly Feldman, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Abhradeep Thakurta. 2019. Amplification by shuffling: From local to central differential privacy via anonymity. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2468–2479. SIAM.

Vitaly Feldman, Audra McMillan, and Kunal Talwar. 2022. Hiding among the clones: A simple and nearly optimal analysis of privacy amplification by shuffling. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 954–964. IEEE.

Ivan Habernal. 2021. When differential privacy meets NLP: The devil is in the detail. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1522–1528, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ivan Habernal. 2022. How reparametrization trick broke differentially-private text representation learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 771–777, Dublin, Ireland. Association for Computational Linguistics.

Victor Petrén Bach Hansen, Atula Tejaswi Neerkaje, Ramit Sawhney, Lucie Flek, and Anders Søgaard. 2022. The impact of differential privacy on group disparity mitigation. In *Proceedings of the Fourth Workshop on Privacy in Natural Language Processing*, pages 12–12.

Jonathan Heek, Anselm Levskaya, Avital Oliver, Marvin Ritter, Bertrand Rondepierre, Andreas Steiner, and Marc van Zee. 2023. Flax: A neural network library and ecosystem for JAX. `http://github.com/google/flax`.

Sorami Hisamoto, Matt Post, and Kevin Duh. 2020. Membership inference attacks on sequence-to-sequence models: Is my data in your machine translation system? *Transactions of the Association for Computational Linguistics*, 8:49–63.

Shlomo Hoory, Amir Feder, Avichai Tendler, Sofia Erell, Alon Peled-Cohen, Itay Laish, Hootan Nakhost, Uri Stemmer, Ayelet Benjamini, Avinatan Hassidim, and Yossi Matias. 2021. Learning and Evaluating a Differentially Private Pre-trained Language Model. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1178–1189, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Justin Hsu, Marco Gaboardi, Andreas Haeberlen, Sanjeev Khanna, Arjun Narayan, Benjamin C Pierce, and Aaron Roth. 2014. Differential privacy: An economic method for choosing epsilon. In *2014 IEEE 27th Computer Security Foundations Symposium*, pages 398–410. IEEE.

Lijie Hu, Ivan Habernal, Lei Shen, and Di Wang. 2024. Differentially Private Natural Language Models: Recent Advances and Future Directions. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics*, page (to appear), Malta. Association for Computational Linguistics.

Timour Igamberdiev and Ivan Habernal. 2022. Privacy-Preserving Graph Convolutional Networks for Text Classification. In *Proceedings of the Language Resources and Evaluation Conference*, pages 338–350, Marseille, France. European Language Resources Association.

Timour Igamberdiev and Ivan Habernal. 2023. DP-BART for privatized text rewriting under local differential privacy. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13914–13934, Toronto, Canada. Association for Computational Linguistics.

Paweł Kamocki and Jim O'Regan. 2016. Privacy issues in online machine translation services-european perspective. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4458–4462.

Shiva Prasad Kasiviswanathan, Homin K Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. 2011. What can we learn privately? *SIAM Journal on Computing*, 40(3):793–826.

Oleksandra Klymenko, Stephen Meisenbacher, and Florian Matthes. 2022. Differential privacy in natural language processing the story so far. In *Proceedings of the Fourth Workshop on Privacy in Natural Language Processing*, pages 1–11, Seattle, United States. Association for Computational Linguistics.

Jaewoo Lee and Chris Clifton. 2011. How Much Is Enough? Choosing ϵ for Differential Privacy. In *Proceedings of the 14th Information Security Conference (ISC 2011)*, pages 325–340, Xi'an, China. Springer Berlin / Heidelberg.

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis,

Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matussière, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander Rush, and Thomas Wolf. 2021. Datasets: A community library for natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–184, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xuechen Li, Florian Tramer, Percy Liang, and Tatsunori Hashimoto. 2022. Large language models can be strong differentially private learners. In *International Conference on Learning Representations*.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Justus Mattern, Zhijing Jin, Benjamin Weggenmann, Bernhard Schoelkopf, and Mrinmaya Sachan. 2022. Differentially private language models for secure data sharing. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4860–4873, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR.

Mariana Neves, Antonio Jimeno Yepes, Amy Siu, Roland Roller, Philippe Thomas, Maika Vicente Navarro, Lana Yeganova, Dina Wiemann, Giorgio Maria Di Nunzio, Federica Vezzani, et al. 2022. Findings of the wmt 2022 biomedical translation shared task: Monolingual clinical case reports. In *WMT22-Seventh Conference on Machine Translation*, pages 694–723.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Peyman Passban, Tanya Roosta, Rahul Gupta, Ankit Chadha, and Clement Chung. 2022. Training mixed-domain translation models via federated learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2576–2586.

Natalia Ponomareva, Jasmijn Bastings, and Sergei Vassilvitskii. 2022. Training text-to-text transformers with privacy guarantees. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2182–2193, Dublin, Ireland. Association for Computational Linguistics.

Natalia Ponomareva, Hussein Hazimeh, Alex Kurakin, Zheng Xu, Carson Denison, H Brendan McMahan, Sergei Vassilvitskii, Steve Chien, and Abhradeep Guha Thakurta. 2023. How to dp-fy ml: A practical guide to machine learning with differential privacy. *Journal of Artificial Intelligence Research*, 77:1113–1201.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Matīss Rikters, Ryokan Ri, Tong Li, and Toshiaki Nakazawa. 2019. Designing the business conversation corpus. In *Proceedings of the 6th Workshop on Asian Translation*, pages 54–61, Hong Kong, China. Association for Computational Linguistics.

Tanya Roosta, Peyman Passban, and Ankit Chadha. 2021. Communication-efficient federated learning for neural machine translation. *arXiv preprint arXiv:2112.06135*.

Manuel Senge, Timour Igamberdiev, and Ivan 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*, pages 7340–7353, Abu Dhabi, UAE.

Weiyan Shi, Aiqi Cui, Evan Li, Ruoxi Jia, and Zhou Yu. 2022. Selective differential privacy for language modeling. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2848–2859, Seattle, United States. Association for Computational Linguistics.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pages 3–18. IEEE.

Pranav Subramani, Nicholas Vadivelu, and Gautam Kamath. 2021. Enabling Fast Differentially Private SGD via Just-in-Time Compilation and Vectorization. In *Advances in Neural Information Processing Systems*, volume 34, pages 26409–26421. Curran Associates, Inc.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008, Long Beach, CA, USA. Curran Associates, Inc.

Jianzong Wang, Zhangcheng Huang, Lingwei Kong, Denghao Li, and Jing Xiao. 2021. Modeling without sharing privacy: Federated neural machine translation. In *Web Information Systems Engineering–WISE 2021: 22nd International Conference on Web Information Systems Engineering, WISE 2021, Melbourne, VIC, Australia, October 26–29, 2021, Proceedings, Part I 22*, pages 216–223. Springer.

Yu-Xiang Wang, Borja Balle, and Shiva Prasad Kasiviswanathan. 2019. Subsampled rényi differential privacy and analytical moments accountant. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1226–1235. PMLR.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

Christopher Weiss, Frauke Kreuter, and Ivan Habernal. 2023. To share or not to share: What risks would laypeople accept to give sensitive data to differentially-private NLP systems? *arXiv preprint*.

Xinwei Wu, Li Gong, and Deyi Xiong. 2022. Adaptive differential privacy for language model training. In *Proceedings of the First Workshop on Federated Learning for Natural Language Processing (FL4NLP 2022)*, pages 21–26, Dublin, Ireland. Association for Computational Linguistics.

Chang Xu, Jun Wang, Francisco Guzmán, Benjamin Rubinstein, and Trevor Cohn. 2021. Mitigating data poisoning in text classification with differential privacy. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4348–4356, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Ying Yin and Ivan Habernal. 2022. Privacy-preserving models for legal natural language processing. In *Proceedings of the Natural Legal Language Processing Workshop 2022*, pages 172–183, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Ashkan Yousefpour, Igor Shilov, Alexandre Sablayrolles, Davide Testuggine, Karthik Prasad, Mani Malek, John Nguyen, Sayan Ghosh, Akash Bharadwaj, Jessica Zhao, Graham Cormode, and Ilya Mironov. 2021. Opacus: User-Friendly Differential Privacy Library in PyTorch. *arXiv preprint*.

Da Yu, Saurabh Naik, Arturs Backurs, Sivakanth Gopi, Huseyin A Inan, Gautam Kamath, Janardhan Kulkarni, Yin Tat Lee, Andre Manoel, Lukas Wutschitz, et al. 2021. Differentially private fine-tuning of language models. In *International Conference on Learning Representations*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating Text Generation with BERT. In *International Conference on Learning Representations*.

A Background on Differential Privacy and DP-SGD

Differential Privacy Differential privacy (DP) is a mathematical framework which formally guarantees that the output of a randomized algorithm $\mathcal{M} : \mathcal{X} \rightarrow \mathcal{Y}$ abides by the following inequality in Eqn. 1, for all *neighboring* datasets $x, x' \in \mathcal{X}$, i.e. datasets which are identical to one another, with the exception of one data point (Dwork and Roth, 2013)

$$\Pr[\mathcal{M}(x) \in S] \leq e^{\varepsilon} \Pr[\mathcal{M}(x') \in S] + \delta, \quad (1)$$

for all $S \subseteq \mathcal{Y}$.

We refer to the algorithm $\mathcal{M}$ as being (ε, δ)-differentially private, where $\varepsilon \in [0, \infty)$, also known as the *privacy budget*, represents the strength of the privacy guarantee. A lower ε value represents an exponentially stronger privacy protection. $\delta \in [0, 1]$ is a very small constant which relaxes the pure differential privacy of $(\varepsilon, 0)$-DP, providing better composition when iteratively applying multiple DP mechanisms to a given dataset.

In order to transform a non-private algorithm $f : \mathcal{X} \rightarrow \mathcal{Y}$ into one satisfying an (ε, δ)-DP guarantee, we generally add Gaussian noise to the output of f. Overall, the whole process restricts the degree to which any single data point can stand out when applying algorithm $\mathcal{M}$ on a dataset.

DP-SGD A popular method for applying DP to the domain of machine learning is through differentially private stochastic gradient descent (DP-SGD) (Abadi et al., 2016b). The core of the methodology relies on adding two extra steps to the original stochastic gradient descent algorithm. For any input data point x_i, we first calculate the gradient of the loss function for a model with parameters θ, $\mathcal{L}(\theta)$, at training iteration t. Hence, $g_t(x_i) = \nabla_{\theta_t} \mathcal{L}(\theta_t, x_i)$.

We then incorporate a *clipping* step, in which the ℓ_2-norm of $g_t(x_i)$ is clipped with clipping constant

C, as in Eqn. 2, in order to constrain the range of possible values. This is followed by a *perturbation* step, adding Gaussian noise to the clipped gradients, as in Eqn. 3.

$$\bar{g}_t(x_i) = \frac{g_t(x_i)}{\max\left(1, \frac{||g_t(x_i)||_2}{C}\right)} \tag{2}$$

$$\hat{g}_t = \frac{1}{L}\sum_{i \in L}\left(\bar{g}_t(x_i) + \mathcal{N}(0, \sigma^2 C^2 \mathbf{I})\right) \tag{3}$$

Importantly, L represents the *lot size*, being a group of data points that are randomly drawn from the full training dataset at each iteration. The final gradient descent step is then taken with respect to this noisy gradient $\hat{g}_t$. We outline the DP-SGD algorithm in more detail in Algorithm 1.

Algorithm 1 DP-SGD

function DP-SGD($f(\boldsymbol{x}; \Theta)$, $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$, $|L|$ — 'lot' size, T — # of steps)
  for $t \in (1, 2, \ldots, T)$ **do**
    Add each training example to a 'lot' L_t with probability $|L|/N$
    for each example in the 'lot' $\boldsymbol{x}_i \in L_t$ **do**
      $\boldsymbol{g}(\boldsymbol{x}_i) \leftarrow \nabla\mathcal{L}(\theta_t, \boldsymbol{x}_i)$ ▷ Compute gradient
      $\bar{\boldsymbol{g}}(\boldsymbol{x}_i) \leftarrow \boldsymbol{g}(\boldsymbol{x}_i)/\max\left(1, \|\boldsymbol{g}(\boldsymbol{x}_i)\|/C\right)$ ▷ Clip gradient
      $\tilde{\boldsymbol{g}}(\boldsymbol{x}_i) \leftarrow \bar{\boldsymbol{g}}(\boldsymbol{x}_i) + \mathcal{N}(0, \sigma^2 C^2 \boldsymbol{I})$ ▷ Add noise
    $\hat{\boldsymbol{g}} \leftarrow \frac{1}{|L|}\sum_{k=1}^{|L|} \tilde{\boldsymbol{g}}(\boldsymbol{x}_k)$ ▷ Gradient estimate of 'lot' by averaging
    $\Theta_{t+1} \leftarrow \Theta_t - \eta_t \hat{\boldsymbol{g}}$ ▷ Update parameters by gradient descent
  return Θ

B Hyperparameters

We present our hyperparameter search space as follows. We experiment with learning rates in the range $[10^{-5}, 0.01]$ and maximum sequence lengths in $[8, 64]$. Following previous work, we utilize large batch and lot sizes for our experiments, finding $1,048,576$ to be the best for WMT-16, $2,048$ for BSD, and 256 for ClinSPEn-CC. We build up these batch sizes using gradient accumulation with a physical batch size of 16. In the case of Poisson sampling, we first sample using large lot sizes and build the resulting drawn batch using gradient accumulation, as described in Section 4. We train models for up to 25 epochs, using the same definition for *epochs* as in Abadi et al. (2016b) in the Poisson sampling setting, being $\frac{N}{L}$. We take the ceiling in case of L not cleanly dividing into N. Each configuration is run using 5 seeds for the BSD and ClinSPEn-CC datasets and 3 seeds for WMT-16, reporting the mean and standard deviation of results.

We additionally present our computational runtimes in Table 2. All experiments are run on up to two 80GB NVIDIA A100 Tensor Core GPUs.

Dataset	ε	Iteration Method	Epoch Time
WMT-16	∞	Random shuffling	2 h 45 m 08 s
WMT-16	1000	Random shuffling	2 h 59 m 15 s
WMT-16	1000	Poisson sampling	4 h 08 m 01 s
WMT-16	5	Random shuffling	1 h 30 m 03 s
WMT-16	5	Poisson sampling	4 h 02 m 35 s
WMT-16	1	Random shuffling	1 h 29 m 49 s
WMT-16	1	Poisson sampling	4 h 09 m 02 s
BSD	∞	Random shuffling	0 h 01 m 17 s
BSD	1000	Random shuffling	0 h 01 m 59 s
BSD	1000	Poisson sampling	0 h 01 m 49 s
BSD	5	Random shuffling	0 h 00 m 52 s
BSD	5	Poisson sampling	0 h 01 m 49 s
BSD	1	Random shuffling	0 h 01 m 09 s
BSD	1	Poisson sampling	0 h 02 m 15 s
ClinSPEn-CC	∞	Random shuffling	0 h 00 m 09 s
ClinSPEn-CC	1000	Random shuffling	0 h 00 m 05 s
ClinSPEn-CC	1000	Poisson sampling	0 h 00 m 28 s
ClinSPEn-CC	5	Random shuffling	0 h 00 m 10 s
ClinSPEn-CC	5	Poisson sampling	0 h 00 m 27 s
ClinSPEn-CC	1	Random shuffling	0 h 00 m 15 s
ClinSPEn-CC	1	Poisson sampling	0 h 00 m 27 s

Table 2: Sample epoch runtimes for each configuration. Some differences between configurations arise due to different optimal hyperparameters, with larger sequence lengths leading to longer epoch times.

C Detailed Results

Dataset	ε	Iteration Method	Test BLEU	Test BERTScore
WMT-16	∞	Random shuffling	36.19 (0.13)	0.95 (0.00)
WMT-16	1000	Random shuffling	20.86 (0.56)	0.92 (0.00)
WMT-16	1000	Poisson sampling	15.12 (0.08)	0.91 (0.00)
WMT-16	5	Random shuffling	19.24 (0.52)	0.92 (0.00)
WMT-16	5	Poisson sampling	7.23 (0.21)	0.89 (0.00)
WMT-16	1	Random shuffling	19.83 (0.64)	0.92 (0.00)
WMT-16	1	Poisson sampling	2.35 (0.07)	0.84 (0.00)
BSD	∞	Random shuffling	10.09 (2.75)	0.90 (0.01)
BSD	1000	Random shuffling	1.36 (0.67)	0.87 (0.01)
BSD	1000	Poisson sampling	1.01 (0.07)	0.87 (0.00)
BSD	5	Random shuffling	0.06 (0.05)	0.85 (0.01)
BSD	5	Poisson sampling	0.06 (0.06)	0.84 (0.02)
BSD	1	Random shuffling	0.00 (0.01)	0.45 (0.22)
BSD	1	Poisson sampling	0.00 (0.00)	0.65 (0.15)
ClinSPEn-CC	∞	Random shuffling	5.42 (2.41)	0.86 (0.02)
ClinSPEn-CC	1000	Random shuffling	0.03 (0.02)	0.75 (0.01)
ClinSPEn-CC	1000	Poisson sampling	0.70 (0.19)	0.78 (0.00)
ClinSPEn-CC	5	Random shuffling	0.80 (0.56)	0.79 (0.00)
ClinSPEn-CC	5	Poisson sampling	0.83 (0.27)	0.79 (0.00)
ClinSPEn-CC	1	Random shuffling	0.50 (0.20)	0.78 (0.00)
ClinSPEn-CC	1	Poisson sampling	0.54 (0.22)	0.78 (0.00)

Table 3: Detailed results of each experimental configuration. Scores shown as "mean (standard deviation)". Results show the average over 3 seeds for the WMT-16 dataset, and 5 seeds for BSD and ClinSPEn-CC.