A Collocation-based Method for Addressing Challenges in Word-level Metric Differential Privacy

Stephen Meisenbacher, Maulik Chevli, and Florian Matthes
Technical University of Munich
School of Computation, Information and Technology
Department of Computer Science
Garching, Germany
{stephen.meisenbacher,maulikk.chevli,matthes}@tum.de

Abstract

Applications of Differential Privacy (DP) in NLP must distinguish between the syntactic level on which a proposed mechanism operates, often taking the form of *word-level* or *document-level* privatization. Recently, several word-level *Metric* Differential Privacy approaches have been proposed, which rely on this generalized DP notion for operating in word embedding spaces. These approaches, however, often fail to produce semantically coherent textual outputs, and their application at the sentence- or document-level is only possible by a basic composition of word perturbations. In this work, we strive to address these challenges by operating *between* the word and sentence levels, namely with *collocations*. By perturbing n-grams rather than single words, we devise a method where composed privatized outputs have higher semantic coherence and variable length. This is accomplished by constructing an embedding model based on frequently occurring word groups, in which unigram words co-exist with bi- and trigram collocations. We evaluate our method in utility and privacy tests, which make a clear case for tokenization strategies beyond the word level.

1 Introduction

The study of Differential Privacy (DP) in Natural Language Processing has brought about a number of innovative approaches, ranging from text rewriting to private fine-tuning of language models (Hu et al., 2024). At the core of these approaches is the goal of providing a level of quantifiable privacy protection when text is shared or used for some downstream purpose. Among other advantages, leveraging DP allows for flexibility in choice of privacy level, governed by the privacy budget, or ε.

An early form of DP in NLP comes with the notion of *word-level Metric Differential Privacy* (MLDP), the goal of which is to allow for privacy-preserving analysis on text documents by performing word-level *perturbations* (Feyisetan et al., 2020). In essence, a word is obfuscated by adding random noise to its embedding, perturbing to a (possibly different) word, and then releasing this "privatized" word (Klymenko et al., 2022). Metric DP is ensured via the implementation of *mechanisms* which add calibrated noise to text representations. While other recent advances in DP NLP have shifted towards more complex language models, the simplicity and atomicity of word-level MLDP methods make a case for its further study.

Although these works show promising results in balancing privacy and utility in the MLDP setting, a number of challenges have also been highlighted (Klymenko et al., 2022). Firstly, the design of mechanisms raises challenges when the underlying spaces, e.g., word embeddings, are both vast (large vocabularies) and complex (high dimensional) (Feyisetan et al., 2021). Moreover, applying DP at the word level and composing these results for private text generation often results in texts with grammatical errors (Mattern et al., 2022). Beyond this, composed word-level MLDP will always lead to privatized documents with the same length as the input documents, diminishing privacy protections.

In this work, we aim to address these challenges by building upon the promise of MLDP mechanisms, but rather than rely on *word-level* perturbations, we extend these mechanisms to operate on the *collocation-level*, or more generally, the *n-gram* level. *By specifically focusing on collocations, we hope to improve output text coherence, introduce generated length variability, and boost utility while also performing fewer overall perturbations, thus saving privacy budget.* In particular, we are guided by the following research question:

> *Can collocations be leveraged to improve the function of word-level Metric Differential Privacy mechanisms, and what is the effect on privacy and utility?*

Proceedings of the Fifth Workshop on Privacy in Natural Language Processing, pages 39–51

We answer this question by designing a new approach for MDLP perturbations which leverages collocation embedding models in conjunction with two proposed collocation extraction algorithms. In our conducted utility and privacy tests, we show that this simple, yet meaningful augmentation leads to improved utility and comparable privacy under a number of privatization strategies. Concretely, the contributions of our work are as follows:

1. To the best of the authors' knowledge, we are the first work to explore the use of collocations in the DP NLP space, most notably through the use of joint n-gram embedding models.
2. We demonstrate the effectiveness of using collocation-based embedding models as a basis for MLDP mechanisms, rather than previous word-level approaches.
3. We provide a blueprint for further improving MLDP mechanisms through the open-sourcing of our collocation extraction algorithms and embedding models, found at https://github.com/sjmeis/CLMLDP.

2 Foundations

2.1 Differential Privacy

Differential Privacy (DP) (Dwork, 2006) provides mathematical privacy guarantees for individual's data when their data undergoes algorithmic processing. Intuitively, it provides plausible deniability on the result about the source of input to an algorithm. An algorithm (or a *mechanism*) that is DP yields similar results irrespective of the inclusion of a single data record in the input dataset. These types of datasets that differ only in a single record are called *adjacent* or *neighboring* datasets.

Consider two adjacent datasets D and D' differing only in a single record. A randomized mechanism $\mathcal{M} : \mathcal{X}^m \rightarrow \mathcal{O}$ that takes a dataset $D \in \mathcal{X}^m$ and results in some output $O \in \mathcal{O}$ is called a (ε, δ)-DP iff for all adjacent datasets D, D' and $\forall O \subseteq \mathcal{O}$, the following holds with $\varepsilon \geq 0$ and $\delta \in [0, 1]$:

$$\mathbb{P}[\mathcal{M}(\mathcal{D}) \in O] \leq e^{\varepsilon} \cdot \mathbb{P}[\mathcal{M}(\mathcal{D}') \in O] + \delta$$

The notion of adjacency of datasets defines the element protected by DP. If adjacent datasets D and D' differ in one record, a DP mechanism provides plausible deniability about the inclusion or exclusion of a single record in the dataset. When the data records are collected at a central location and then a DP mechanism is to be applied, the adjacency notion can be defined as aforementioned and it is called *Global DP*. However, if the data collector is not trusted and the DP mechanism is applied locally before the collection of data, the notion of adjacency is defined as any two data records; this is called *Local DP* (Duchi et al., 2013).

For natural language, the unstructured nature of data brings additional challenges regarding the notion of adjacent datasets (Klymenko et al., 2022). We consider a text consisting of n-gram tokens, and define the notion of adjacency as any two tokens following Feyisetan et al. (2020). Hence, an adversary cannot determine with high probability the source token of the privatized token.

2.2 Metric Differential Privacy (MDP)

For two finite sets $\mathcal{X}$ and $\mathcal{Z}$ and a distance metric $d : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}+$ defined for the set $\mathcal{X}$, a randomized mechanism $\mathcal{M} : \mathcal{X} \rightarrow \mathcal{Z}$ satisfies metric differential privacy or $\varepsilon d_{\mathcal{X}}$-privacy *iff* $\forall x, x' \in \mathcal{X}$ and $\forall z \in \mathcal{Z}$, this condition is satisfied with $\varepsilon > 0$:

$$\frac{\mathbb{P}[\mathcal{M}(x) = z]}{\mathbb{P}[\mathcal{M}(x') = z]} \leq e^{\varepsilon d(x, x')} \tag{1}$$

Metric DP is a relaxation of DP where instead of considering the worst-case guarantees, the privacy guarantees scale according to the distance between adjacent datasets (Chatzikokolakis et al., 2013). This allows for greater utility and flexibility alongside a mathematical guarantee.

2.3 MDP for a Sentence

We assume a vocabulary set consisting of all the tokens in $\mathcal{V}$, with the tokens as points in the embedding space. The embedding function $\Phi : \mathcal{V} \rightarrow \mathbb{R}^d$ gives the position of the tokens in the space. Additionally, we assume that the space $\mathcal{V}$ is equipped with a distance metric $d_{\mathcal{V}} : \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{R}+$ that gives us the distance between two tokens w and w' as

$$d_{\mathcal{V}}(w, w') = ||\Phi(w) - \Phi(w')||_2 \tag{2}$$

If a mechanism $\mathcal{M}$ satisfies MDP for two tokens for $\varepsilon > 0$, it satisfies Equation 1 $\forall w, w' \in \mathcal{V}$, and thus, we have the following inequality:

$$\frac{\mathbb{P}[\mathcal{M}(w) = x]}{\mathbb{P}[\mathcal{M}(w') = x]} \leq e^{\varepsilon \cdot d_{\mathcal{V}}(w, w')} \tag{3}$$

This guarantee can be extended to the whole sentence consisting of n tokens, i.e., $s = w_1 \cdot w_2 \cdots w_n$. Following Feyisetan et al. (2020), a token-level mechanism can be applied to each token independently and a privatized sentence can

be generated by concatenating these privatized tokens, i.e., $z = x_1 \cdot x_2 \cdots x_n$. If the distance function that takes sentences of the same token length $D : \mathcal{V}^n \times \mathcal{V}^n \rightarrow \mathbb{R}+$ is defined as $\mathcal{D} = \sum_{i=1}^{n} d_{\mathcal{V}}(w_i, x_i)$, the privacy guarantees of applying mechanism $\mathcal{M}$ to the sentence can be derived as follows:

$$\begin{aligned}\frac{\mathbb{P}[\mathcal{M}(s) = z]}{\mathbb{P}[\mathcal{M}(s') = z]} &= \prod_{i=1}^{n} \frac{\mathbb{P}[\mathcal{M}(w_i) = x]}{\mathbb{P}[\mathcal{M}(w_i') = x]} \\ &\leq \prod_{i=1}^{n} \exp\left(\varepsilon \cdot d_{\mathcal{V}}(w_i, w_i')\right) \\ &= \exp\left(\varepsilon \cdot \sum_{i=1}^{n} d_{\mathcal{V}}(w_i, w_i')\right) \\ &= \exp\left(\varepsilon \cdot \mathcal{D}(s, s')\right)\end{aligned}$$

It should be noted that while we use the term "sentence" here, the above can be generalized to text "documents".

2.4 The Theory of Collocations

In linguistics, *collocations* are defined as groupings of words that often appear together in language. More specifically, collocations are word groups ("multi-word expressions") existing in the space between idioms and free word groups (McKeown and Radev, 2000), where the meaning of idioms cannot be understood by their individual words. Intuitively, collocations can be defined as groupings of words that appear in predictable patterns (*good morning*), without being as rigid as idioms (*sleep like a baby*) (McKeown and Radev, 2000).

An important concept is the *Contextual Theory of Meaning* of John Rupert Firth (Léon, 2005; Manning and Schutze, 1999), famously summarized by "a word is characterized by the company it keeps". The meaning of a given collocation only takes form when viewing the group as a whole, and not by examining the meaning of each word individually.

Looking to the notion of differentially private text rewriting via the composition of word-level replacements, one may imagine that the theory of collocations sheds light on the potential pitfalls of isolated word substitutions. As highlighted by Mattern et al. (2022), word-level DP disregards context, which results in semantically disjoint replacements as well as frequent grammatical incongruities. In this light, we posit that collocations may improve both of these challenges, as collocations represent groups of words with *bundled* meaning, and within a collocation, proper grammar must be upheld.

3 Related Work

3.1 Word-level MLDP

While Fernandes et al. (2019) proposed an early implementation of metric DP, (Feyisetan et al., 2020) were the first to design a word-level MLDP mechanism for static word embeddings. Ensuing works aim to improve word-level methods through various means, including differing metrics (Xu et al., 2020), nearest neighbor mapping (Xu et al., 2021b; Meisenbacher et al., 2024a), or noise mechanism (Xu et al., 2021a; Carvalho et al., 2023). Other works focus on the selection of words to privatize (Yue et al., 2021; Chen et al., 2022).

We aim to build upon this body of work, while also addressing the known challenges of semantic coherence, grammatical correctness, and output text length variability. In particular, we tackle these challenges in the word-level MLDP setting by leveraging *collocations* and *n-gram embeddings*.

3.2 Collocation Extraction and Evaluation

Several computational approaches for automatic collocation extraction have been explored. Pecina (2005) surveys an extensive list of early collocation extraction methods, and later explores the combination of different metrics (Pecina and Schlesinger, 2006). Other works improve on classic association measures (Bouma, 2010; Brezina et al., 2015), and more recent work has focused on evaluating end-to-end solutions (Bhalla and Klimcikova, 2019; Espinosa Anke et al., 2021). More on the theoretical underpinnings and our motivation for the use of collocations can be found in Section 4.

3.3 N-gram Embeddings

Extending static embedding models beyond the word level often takes the form of *n-gram* embeddings or *phrase* embeddings (Poliak et al., 2017; Yin and Schütze, 2014). Works have explored different methods of embedding n-grams, notably the use of Pointwise Mutual Information (PMI) (Zhao et al., 2017) or BERT-based models for more contextual phrase embeddings (Wang et al., 2021).

In a study of n-gram embeddings, Gupta et al. (2019) find that the joint training process improves the quality of single-word embeddings. In other works, it is shown that n-gram embeddings can improve a variety of NLP tasks Bai et al. (2018); Zhang et al. (2014); Yin and Schütze (2015).

With these works as motivation, we investigate whether n-gram embeddings can serve to improve

Method Text:	*I think, therefore I am*	Tokens	PMI	Token Budget (ε = 10)
S1: Word Tokenization	i · think · , · therefore · i · am	6	--	1.67
S2: GST + Word-level Guarantee	i_think · , · therefore_i · am	4	7.53	1.67
S3: Collocation Tokenization (GST)	i_think · , · therefore_i · am	4	7.53	2.5
S4: Collocation Tokenization (MST)	i_think · , · therefore · i_am	4	12.43	2.5

Figure 1: An example of word tokenization versus collocation tokenization. Collocation tokenization will often result in fewer tokens, as collocations frequently occur in natural language. *Token budget* denotes the privacy budget assigned to each token given an example document-level budget (e.g., $\varepsilon = 10$) and assuming basic composition.

DP text privatization approaches previously relying on word embeddings. In particular, we explore the usefulness of embedding *collocations* as the underlying embedding model of MLDP mechanisms.

4 A Collocation-based MLDP Method

In this section, we describe our proposed method, which differs from word-level MLDP methods in that it sets the underlying metric space to that of a *jointly trained* model of unigrams, bigram collocations, and trigram collocations. We outline a method to extract collocations, the training of the abovementioned embedding model, and the augmentation of existing MLDP mechanisms.

4.1 Extracting Collocations

The first challenge of dealing with collocations is the reliable extraction of meaningful multi-word expressions that uphold the definition of a collocation. Several methods have been proposed by the literature, ranging from simple frequency-based approaches, methods looking at syntactic co-occurrences, to *hypothesis testing* methods or *association measures* such as mutual information (Evert, 2009; Manning and Schutze, 1999).

In this work, we focus on the extraction of bigram and trigram collocations via the use of *Pointwise Mutual Information* (PMI) (Church and Hanks, 1990). Essentially, PMI indicates how much one point (word) tells us about another. In other words, if the presence of one word *decreases* the uncertainty of the presence of another word, these two words have a high PMI. In the case of bigrams, two words x and y have a PMI as follows:

$$PMI(x, y) = \log_2 \frac{P(x|y)}{P(x)} = \log_2 \frac{P(y|x)}{P(y)} \quad (4)$$

Given a corpus of N words, we can empirically measure the bigram PMI of xy as defined in Equation 4 by the following:

$$PMI(x, y) = \log_2 \frac{N \cdot c(xy)}{c(x) \cdot c(y)} \quad (5)$$

Note that in Equation 5, the order of the unigrams matters, and c denotes the raw frequency count of a given unigram or bigram. For trigram collocations, a simple modification can be made:

$$PMI(x, y, z) = \log_2 \frac{N^2 \cdot c(xyz)}{c(x) \cdot c(y) \cdot c(z)} \quad (6)$$

4.1.1 Empirical Collocations

For the extraction of *empirical* collocations (Evert, 2009), i.e., those that can be derived via empirical means, we measure the PMI of bigrams and trigrams from a selected random sample of 2.5 million texts of the publicly available large-scale text corpus C4 (Colossal Cleaned Common Crawl) (Raffel et al., 2020). After counting the frequency of all unigrams, bigrams, and trigrams, we calculate the bigram and trigram PMI values using Equations 5 and 6, respectively. We filter the results for all values with a PMI score of 2.0 or higher *and* not containing any English connector words (e.g., *a*, *an*, *the*, *and*, *or*, etc.)[1]. This process results in a set of 3.02 million bigrams and 1.31 million trigrams[2].

4.1.2 Collocation-level Tokenization

We design an extraction algorithm that will tokenize a given input text into its unigram, bigram, and trigram counterparts based upon the empirically derived PMI scores of the collocations. To do this, we define two scoring methods (pseudocode found in Appendix Algorithms 1 and 2):

- **Greedy Sequential Tokenization (GST)**: a text is tokenized *greedily* by processing the tokens in order, with trigrams being prioritized. This is described in Algorithm 1.
- **Max Score Tokenization (MST)**: a text is tokenized in a way that maximizes the overall PMI score of the resulting tokenized text. This is described in Algorithm 2.

[1] As defined by the Python GENSIM package.
[2] Can be found in the data folder of our code repository.

Algorithm 1
Greedy Sequential Tokenization (GST)

Require: scored bigrams B, scored trigrams T, input $text$
  tkns $\leftarrow$ $word\_tokenize(text)$
  bigram_cands $\leftarrow$ *get_bigrams*(tkns).*intersect*(B)
  trigram_cands $\leftarrow$ *get_trigrams*(tkns).*intersect*(T)
  n $\leftarrow$ *length*(tkns)
  output $\leftarrow$ []
  for $idx \in 1...n$ **do**
    cand $\leftarrow$ trigram_cands.*find*(tkns[idx: $idx+2$])
    if !cand **then**
      cand $\leftarrow$ bigram_cands.*find*(tkns[idx: $idx+1$])
    end if
    if !cand **then**
      output.$append(text[idx])$ ▷ unigram
    else
      output.$append$(cand)
    end if
    bigram_cands.$delete$(cand)
    trigram_cands.$delete$(cand)
    if cand $\in B$ **then** ▷ advance to next unmatched word
      idx $+= 2$
    else
      idx $+= 3$
    end if
  end for
  return output

Algorithm 2
Max Score Tokenization (MST)

Require: scored bigrams B, scored trigrams T, input $text$
  unigrams $\leftarrow$ $word\_tokenize(text)$
  bigram_cands $\leftarrow$ *get_bigrams*($text$).*intersect*(B)
  trigram_cands $\leftarrow$ *get_trigrams*($text$).*intersect*(T)
  cands $\leftarrow$ *sorted*(unigrams + bigram_cands + trigram_cands)
  n $\leftarrow$ *length*(cands)
  matched $\leftarrow$ []
  output $\leftarrow$ []
  for $idx \in 1...n$ **do**
    if all(cands.$tokens$!$\in$ matched) **then**
      output.$append$(cand[idx])
      matched.add(cands.$tokens$)
    end if
  end for
  return output

GST and MST output a list of "tokens", which can be either unigrams, bigram collocations, or trigram collocations. In its application, we tokenize documents at the *sentence-level*, so as not to detect collocations across sentence boundaries. Note that this method can be extended to an arbitrary n-gram level. As a result, there are collocation tokens less than or equal to the number of word tokens.

4.2 A Collocation Embedding Space

We train an embedding model in which unigram words, bigram collocations, and trigram collocations co-exist in a single embedding space. In particular, we train a 300-dimension WORD2VEC model (Mikolov et al., 2013) using the GENSIM package (Řehůřek and Sojka, 2010).

To train the model, we leverage a large subset of the C4 Corpus, namely 250 million text samples, or roughly 500GB. As inputs to the GENSIM trainer, we give the text samples as tokenized by our two algorithms, namely GST and MST, thus resulting in two trained embedding models. The models were trained on a six-core Intel Xeon CPU, with the entire training process (extraction + embedding) taking roughly 90 hours per model. These models are made available in our code repository.

4.3 Augmenting MLDP Mechanisms

With the two collocation embedding models, we can now make a simple augmentation to existing word-level MLDP mechanisms. As these mechanisms typically operate on strictly word (unigram) spaces, we first swap out these models with our trained embedding models. Then, inputs to the mechanisms are tokenized by our collocation extraction algorithms, rather than word tokenization.

The returned tokens can be of word length 1-3. However, the MLDP privacy guarantees are not affected, as the embedding space consists of these variable word-length tokens. Hence, the mechanisms can operate as usual, with the outputs being perturbed uni-, bi-, *or* trigrams. Mathematically, the privacy guarantees for any tokens w, w' in our embedding space remain as defined in Section 2.3.

5 Experimental Setup and Results

In our experiments to test our collocation-based method, we focus on evaluating the effect that can be observed by using collocations rather than pure words. In particular, we perform a two-part evaluation: utility experiments and privacy experiments.

5.1 Mechanism Selection

We center our evaluation around the fundamental MLDP mechanism proposed by Feyisetan et al. (2020), often referred to as MADLIB (Algorithm 3), which typically operates on word embeddings in Euclidean space by adding calibrated multivariate noise. Our goal is to experiment using this mechanism across a range of ε values, with the hopes of generalizing to mechanisms that build on top of MADLIB. Specifically, we choose the values $\varepsilon \in \{0.1, 0.5, 1, 5, 10, 15, 25, 50\}$.

5.2 Utility Experiments

Our utility experiments follow the example set by several previous DP NLP works (Mattern et al.,

Algorithm 3
MADLIB (Feyisetan et al., 2020)

Require: String $x = w_1 w_2 \dots w_n$, privacy parameter $\epsilon > 0$, word set $\mathcal{W}$, embedding function φ
Ensure: Privatized string $\hat{x}$
  for $i \in \{1, \dots, n\}$ **do**
    Compute embedding $\varphi_i = \varphi(w_i)$
    Perturb embedding to obtain $\hat{\varphi}_i = \varphi_i + \mathcal{N}$ with noise density $p_{\mathcal{N}}(z) \propto \exp(-\epsilon \|z\|)$
    Obtain perturbed word $\hat{w}_i = \arg\min_{u \in \mathcal{W}} |\varphi(u) - \hat{\varphi}_i|$
    Insert $\hat{w}_i$ in i^{th} position of $\hat{x}$
  end for
  return $\hat{x}$

2022; Utpala et al., 2023; Igamberdiev and Habernal, 2023), that is to evaluate how well DP generated text can preserve the original utility of the dataset. In particular, texts that are generated by a mechanism are compared against a non-privatized baseline, and the utility (loss) is measured.

To ensure a greater practical relevance, we perform utility experiments for our chosen mechanism at a *document level*, where privatized documents are achieved via the composition of token-level perturbations. For this, we set a *dataset specific privacy budget*, where our "base" ε values introduced above are scaled by the average word length of each dataset. Thus, each text is perturbed with an overall budget of $\varepsilon * \textit{avg\_word\_len}(dataset)$. This ensures that all texts, regardless of length, are offered the same privacy guarantee.

We note here that in this budget calculation, our goal is to provide an equal guarantee for each document to be privatized. However, we do not take into account the effect of the distance function in the Metric DP guarantee; thus, the document level budget is calculated according to pure DP composition, namely with basic composition of ε values.

We evaluate five privatization strategies, which are described below and illustrated in Figure 1:

1. **Non-private**: no DP is applied to a given text.
2. **Word-level** (S1): a text is tokenized by *word*, and the document budget is distributed evenly to each word to be perturbed. For embeddings, we use WORD2VEC-GOOGLE-NEWS-300[3]. Since this model contains three billion tokens, we filter the vocabulary down to that of the DEBERTA-V3-BASE (see next section). In S1, stopwords are not privatized.
3. **Collocation-level, word-level guarantee** (S2): a text is tokenized using our GST collocation extraction algorithm, but each resulting *token* is given the same budget as in the **word-level** scenario (see Figure 1).
4. **Collocation-level (GST)** (S3): a text is tokenized by GST, and the document budget is distributed evenly to all resulting collocations.
5. **Collocation-level (MST)** (S4): same as above, but with the MST algorithm.

Thus, for each given input text, we receive five resulting outputs: the original (baseline) text and four privatized variants. These serve as the basis for our utility (and privacy) experiments.

5.3 Training and Evaluation

Datasets To measure utility, we choose four datasets from the GLUE benchmark (Wang et al., 2018), a standard benchmark representing a variety of language understanding tasks. Specifically, we utilize the COLA, MRPC, RTE, and SST2 datasets. ForSST2, we use a 10k random sample.

We first perturb each dataset according to the strategies outlined above. Note that we privatize both the train and validation sets, as this presents the strictest test of utility preservation in which all data is perturbed. For datasets with two sentences (RTE, MRPC), we only perturb the first sentence.

Model Training The preservation of utility is measured by fine-tuning a language model on all dataset variants (i.e., baseline or perturbed), and measuring the effect on utility. For this, we fine-tune all datasets on a DEBERTA-V3-BASE model with input size of 256, for one epoch and otherwise default HuggingFace Trainer parameters. All training is performed on one NVIDIA A6000 GPU. For stability in the results, we run each training procedure three times on different random shuffles of the data, reporting the average metrics of all runs.

Metrics We report the (micro) F1 score of all trained models on the validation sets. This metric aims to capture the effect of privatization on the ability for a model with good utility to be trained.

In addition, we report the *cosine similarity* (CS) between each (*original*, *private*) dataset pair. This metric can be used to measure the degree to which semantic similarity is preserved in perturbation (Meisenbacher et al., 2024b). For this, we utilize the SENTENCE-TRANSFORMERS/ALL-MINILM-L6-V2 model (Reimers and Gurevych, 2019).

We also use *perplexity* to measure the semantic coherence privatized texts. As perplexity aims to

[3] https://code.google.com/archive/p/word2vec/

measure the ability of a language model to predict a given text, a better (lower) perplexity would imply a text is more "natural" or "predictable". Although this metric has been used in recent DP NLP works (Yue et al., 2023; Singh et al., 2024), its use directly on privatized texts has not been explored widely with the exception of Weggenmann et al. (2022). We report *average perplexity* (AP) of all sequences in a dataset, using GPT-2 (Radford et al., 2019).

5.4 Privacy Experiments

Our privacy experiments take the form of *empirical privacy* measurement, where we use two tasks as proxies for privacy preservation, which also allow for measures of *relative gain* (discussed below):

1. **Yelp Reviews** (Zhang et al., 2015): we utilize the same dataset used by Utpala et al. (2023), which contains a subset of reviews authored by 10 frequent reviewers. From this, we model an *authorship identification* task. We take a random subset of 10k rows.
2. **Trustpilot Reviews** (Hovy et al., 2015): each review includes the gender of the original reviewer (M/F). This creates an *gender identification* task, for which we use a 10k sample.

As with the utility experiments, all texts in the two datasets are privatized according to the five perturbation strategies. The resulting datasets are then divided into a 90-10 train-validation split[4].

Evaluation Both datasets are labeled for sentiment (positive/negative), allowing for a binary classification task, which is carried out in a similar manner as the utility experiments. Macro F1 is reported, as the labels are positive-biased.

Next, empirical privacy is measured. To do this, an adversarial classifier is trained to predict the sensitive attribute (author ID or gender) given the corresponding text. We use the same DEBERTA-V3-BASE fine-tuning process for the creation of this classifier. For evaluation, we follow two adversarial archetypes as proposed in the recent literature (Mattern et al., 2022; Utpala et al., 2023): the *static* and *adaptive* attackers. The static attacker is only able to train on the non-privatized train split and must evaluate on privatized validation splits. The adaptive attacker, a much more capable adversary, is able to train on the privatized train splits.

[4]A random seed of *42* is used throughout this work.

For adversarial performance, we report macro F1 scores. Using both the utility and privacy measurements, we calculate the *relative gain* (RG) of privatization (Mattern et al., 2022), namely whether the gains in privacy outweigh potential losses in utility. This metric is given by the following formula, where P_p, U_p, P_o, U_o are the measured privacy (P) and utility (U) scores of the privatized ($_p$) or original ($_o$) data: $RG = (U_p/U_o) - (P_p/P_o)$.

5.5 Results

The results of the utility experiments are given in Tables 1, 2, and 3, and are illustrated in Figure 2, whereas the privacy results are shown in Table 4.

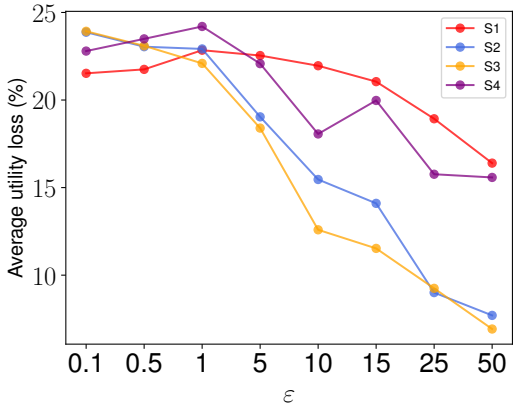

Figure 2: Average Utility Loss. This graph depicts the average utility loss (in F1) for a given base ε value across four GLUE tasks and our four privatization strategies.

ε	0.1	0.5	1	5	10	15	25	50
S1	0.13	0.13	0.13	0.14	0.18	0.25	0.38	0.63
S2	0.16	0.16	0.18	0.42	0.65	0.78	0.88	0.94
S3	0.16	**0.17**	**0.20**	**0.51**	**0.74**	**0.85**	**0.92**	**0.96**
S4	**0.17**	0.15	0.19	0.33	0.45	0.52	0.60	0.68

Table 1: Average cosine similarity between original and privatized texts across all four utility datasets.

Baseline	622							
ε	0.1	0.5	1	5	10	15	25	50
S1	**1731**	**1967**	**2325**	**3593**	5150	5525	5978	3987
S2	3913	4135	4774	4037	2953	2239	1714	1582
S3	3848	4237	4960	3609	**2418**	**1925**	**1632**	**1547**
S4	4855	5456	6103	5429	4673	3056	2574	2302

Table 2: Average perplexity of the privatized texts across all four utility datasets, where lower scores are better.

6 Discussion

Utility Analysis An analysis of the results begins with the strong utility performance of collocation-based perturbation strategies across all tested datasets and ε values. This effect is especially

Baseline	$84.97_{0.4}$							
ε	0.1	0.5	1	5	10	15	25	50
S1	$69.13_{0.0}$	$69.13_{0.0}$	$69.13_{0.0}$	$69.13_{0.0}$	$69.13_{0.0}$	$69.13_{0.0}$	$69.13_{0.0}$	$69.13_{0.0}$
S2	$69.13_{0.0}$	$69.13_{0.0}$	$69.13_{0.0}$	$69.13_{0.0}$	$72.83_{1.3}$	$74.11_{1.0}$	$78.17_{0.0}$	$79.42_{0.2}$
S3	$69.13_{0.0}$	$69.13_{0.0}$	$69.13_{0.0}$	$69.13_{0.0}$	$\mathbf{73.27_{1.9}}$	$\mathbf{75.01_{1.1}}$	$\mathbf{80.22_{0.9}}$	$\mathbf{81.85_{0.4}}$
S4	$69.13_{0.0}$	$69.13_{0.0}$	$69.13_{0.0}$	$69.13_{0.0}$	$69.13_{0.0}$	$69.13_{0.0}$	$69.16_{0.0}$	$69.13_{0.0}$

(a) CoLA (Avg. Words/Text: 7.80)

Baseline	$94.33_{0.2}$							
ε	0.1	0.5	1	5	10	15	25	50
S1	$\mathbf{58.75_{1.9}}$	$\mathbf{56.0_{3.6}}$	$53.94_{2.9}$	$56.8_{0.6}$	$56.73_{5.1}$	$58.87_{2.4}$	$67.78_{1.8}$	$76.11_{0.8}$
S2	$50.76_{0.2}$	$50.92_{0.0}$	$53.25_{1.7}$	$68.0_{0.4}$	$79.05_{0.9}$	$82.76_{0.4}$	$91.67_{0.5}$	$\mathbf{93.16_{0.7}}$
S3	$50.92_{0.0}$	$52.22_{1.8}$	$\mathbf{56.15_{0.7}}$	$\mathbf{71.18_{0.6}}$	$\mathbf{84.56_{1.0}}$	$\mathbf{87.69_{0.4}}$	$\mathbf{92.51_{0.4}}$	$92.78_{0.4}$
S4	$51.61_{0.3}$	$50.92_{0.0}$	$52.68_{2.5}$	$57.11_{4.8}$	$71.25_{0.5}$	$65.9_{0.8}$	$80.2_{2.1}$	$80.24_{0.4}$

(b) SST2 (Avg. Words/Text: 8.82)

Baseline	$85.34_{1.0}$							
ε	0.1	0.5	1	5	10	15	25	50
S1	$69.28_{0.8}$	$69.93_{1.2}$	$\mathbf{70.02_{0.5}}$	$68.38_{0.0}$	$69.69_{0.6}$	$70.1_{0.2}$	$70.75_{0.1}$	$70.75_{0.5}$
S2	$69.93_{1.2}$	$\mathbf{70.67_{0.4}}$	$69.2_{1.2}$	$\mathbf{69.85_{1.1}}$	$70.26_{1.3}$	$71.08_{2.3}$	$\mathbf{76.8_{4.5}}$	$80.72_{1.7}$
S3	$69.2_{1.2}$	$69.53_{1.6}$	$69.61_{1.0}$	$69.12_{1.0}$	$\mathbf{74.35_{1.6}}$	$\mathbf{73.37_{2.5}}$	$74.75_{4.6}$	$\mathbf{81.29_{1.0}}$
S4	$\mathbf{70.26_{1.3}}$	$69.12_{1.0}$	$68.38_{0.0}$	$69.44_{1.2}$	$71.24_{0.1}$	$70.02_{1.2}$	$72.06_{1.1}$	$71.81_{2.1}$

(c) MRPC (Avg. Words/Text: 19.54)

Baseline	$70.97_{2.0}$							
ε	0.1	0.5	1	5	10	15	25	50
S1	$52.35_{0.5}$	$\mathbf{53.55_{0.7}}$	$51.14_{3.0}$	$51.14_{2.2}$	$52.23_{0.5}$	$53.31_{1.4}$	$52.23_{0.9}$	$\mathbf{54.03_{1.5}}$
S2	$50.3_{3.4}$	$52.71_{0.0}$	$52.35_{0.5}$	$52.47_{0.6}$	$51.62_{3.5}$	$51.26_{1.2}$	$\mathbf{52.95_{2.1}}$	$51.5_{1.2}$
S3	$50.66_{2.9}$	$52.35_{0.8}$	$\mathbf{52.35_{0.3}}$	$\mathbf{52.59_{0.2}}$	$\mathbf{53.07_{3.1}}$	$\mathbf{53.43_{2.3}}$	$51.14_{1.3}$	$51.99_{0.8}$
S4	$\mathbf{53.43_{1.0}}$	$52.47_{0.3}$	$48.62_{3.0}$	$51.62_{1.5}$	$51.74_{1.9}$	$50.66_{2.2}$	$51.14_{3.0}$	$52.11_{3.4}$

(d) RTE (Avg. Words/Text: 44.48)

Table 3: Utility Experiment Results. All results represent average micro F1 scores over three training runs, with the standard deviation reported as a subscript. Scores in **bold** denote the highest result for a given dataset and ε value.

		ε							
	Yelp	0.1	0.5	1	5	10	15	25	50
	Baseline	Utility: $81.76_{0.8}$ / Adversary: 90.60							
S1	Utility F1	$48.1_{0.0}$	$48.1_{0.0}$	$48.1_{0.0}$	$48.1_{0.0}$	$48.1_{0.0}$	$48.1_{0.0}$	$48.1_{0.0}$	$48.1_{0.0}$
	Static F1	16.4	15.9	14.4	11.7	13.4	15.4	19.6	30.4
	Adaptive F1	$56.4_{3.6}$	$58.9_{1.6}$	$59.7_{3.0}$	$59.6_{1.2}$	$59.0_{2.5}$	$62.1_{2.1}$	$60.4_{1.3}$	$59.2_{1.5}$
	Relative Gain	-0.03	-0.06	-0.07	-0.07	-0.06	-0.10	-0.08	-0.07
S2	Utility F1	$48.1_{0.0}$	$48.1_{0.0}$	$48.1_{0.0}$	$48.1_{0.0}$	$48.1_{0.0}$	$50.3_{3.2}$	$76.5_{4.2}$	$79.4_{0.3}$
	Static F1	8.7	9.4	9.7	19.8	32.8	42.3	55.8	63.3
	Adaptive F1	$44.1_{3.4}$	$44.0_{4.4}$	$42.9_{2.0}$	$50.6_{2.3}$	$55.0_{1.8}$	$63.6_{0.6}$	$71.6_{3.2}$	$82.2_{2.7}$
	Relative Gain	0.10	**0.10**	0.11	**0.03**	-0.02	-0.09	**0.15**	**0.06**
S3	Utility F1	$48.1_{0.0}$	$48.1_{0.0}$	$48.1_{0.0}$	$48.1_{0.0}$	$55.2_{10.0}$	$58.8_{15.2}$	$69.1_{14.9}$	$79.4_{1.1}$
	Static F1	8.9	9.4	11.0	24.8	40.9	52.2	61.2	64.3
	Adaptive F1	$40.9_{5.4}$	$45.5_{4.1}$	$39.2_{3.3}$	$54.9_{0.8}$	$60.9_{3.8}$	$67.4_{2.6}$	$77.5_{3.2}$	$82.8_{0.8}$
	Relative Gain	**0.14**	0.09	**0.16**	-0.02	**0.00**	**-0.02**	-0.01	**0.06**
S4	Utility F1	$48.1_{0.0}$	$48.1_{0.0}$	$48.1_{0.0}$	$48.1_{0.0}$	$48.1_{0.0}$	$48.1_{0.0}$	$48.1_{0.0}$	$53.1_{3.7}$
	Static F1	9.3	9.6	10.6	17.2	21.3	24.4	31.2	40.5
	Adaptive F1	$42.5_{3.7}$	$45.0_{2.1}$	$42.0_{7.5}$	$52.6_{0.5}$	$56.8_{1.6}$	$57.4_{2.4}$	$61.7_{2.2}$	$66.9_{0.2}$
	Relative Gain	0.12	0.09	0.12	0.01	-0.04	-0.05	-0.09	-0.09

		ε							
	Trustpilot	0.1	0.5	1	5	10	15	25	50
	Baseline	Utility: $98.49_{0.6}$ / Adversary: 68.70							
S1	Utility F1	$48.1_{0.0}$	$50.9_{3.9}$	$48.1_{0.0}$	$48.5_{0.6}$	$48.1_{0.0}$	$48.1_{0.0}$	$50.8_{3.8}$	$68.6_{4.3}$
	Static F1	58.2	58.1	57.9	57.8	57.7	58.1	58.0	60.5
	Adaptive F1	$58.1_{0.2}$	$57.9_{0.0}$	$57.9_{0.9}$	$58.7_{0.7}$	$57.6_{1.1}$	$57.6_{0.8}$	$57.1_{0.7}$	$60.5_{2.2}$
	Relative Gain	-0.36	**-0.32**	**-0.35**	-0.36	-0.35	-0.35	-0.32	-0.18
S2	Utility F1	$48.1_{0.0}$	$48.1_{0.0}$	$48.1_{0.0}$	$63.9_{12.5}$	$87.8_{0.7}$	$94.1_{0.7}$	$96.7_{0.1}$	$97.6_{0.2}$
	Static F1	57.8	57.7	58.1	59.2	62.4	64.7	67.5	67.9
	Adaptive F1	$57.9_{0.0}$	$57.6_{0.5}$	$58.5_{0.8}$	$57.9_{0.0}$	$64.1_{1.0}$	$64.3_{4.5}$	$68.2_{0.5}$	$68.7_{0.7}$
	Relative Gain	**-0.35**	-0.35	-0.36	-0.19	-0.04	**0.02**	-0.01	-0.01
S3	Utility F1	$48.1_{0.0}$	$48.1_{0.0}$	$48.1_{0.0}$	$83.4_{1.3}$	$93.1_{0.6}$	$94.9_{1.6}$	$97.8_{0.3}$	$98.4_{0.2}$
	Static F1	58.0	58.0	58.7	60.2	64.1	64.9	67.7	67.6
	Adaptive F1	$57.9_{0.0}$	$57.9_{0.0}$	$57.9_{0.0}$	$61.3_{2.0}$	$66.6_{0.5}$	$66.6_{0.2}$	$68.2_{0.1}$	$65.1_{5.2}$
	Relative Gain	**-0.35**	-0.35	**-0.35**	**-0.05**	**-0.02**	-0.01	0.00	**0.05**
S4	Utility F1	$48.1_{0.0}$	$48.1_{0.0}$	$48.1_{0.0}$	$54.3_{5.1}$	$78.4_{2.9}$	$86.3_{0.3}$	$95.4_{0.7}$	$95.1_{0.4}$
	Static F1	57.9	58.0	58.5	59.9	62.8	64.3	66.8	67.9
	Adaptive F1	$58.4_{0.7}$	$58.7_{0.6}$	$58.5_{0.7}$	$59.4_{1.0}$	$62.9_{1.3}$	$62.1_{1.5}$	$62.8_{3.5}$	$66.9_{2.0}$
	Relative Gain	-0.36	-0.37	-0.36	-0.31	-0.12	-0.02	**0.05**	-0.01

Table 4: Empirical Privacy Results. The highest *relative gains* (using *adaptive* F1) per ε are **bolded**.

prominent in the SST2 and MRPC tasks. Interestingly, the RTE task presents a challenge for all tested strategies, implying that entailment tasks are more difficult with privatized texts. Nevertheless, the utility loss is dampened with collocation-based methods, particularly at $\varepsilon \geq 1$ (Figure 2).

Another intriguing finding comes with the CoLA results, where all strategies struggle to enable any sort of "true learning" until the $\varepsilon = 10$ threshold. Upon reflection, this particular task may represent the toughest of utility tasks, as the ability to determine the *acceptability* of a given text becomes extremely challenging post-perturbation. Nevertheless, as opposed to S1 (word-level) perturbation, which can never break the worst-case (majority voting) performance, both S2 and S3 are successful in doing this for higher ε values. One can attribute this to the fact that collocation-based perturbation will still preserve traces of semantic coherence, which is crucial for the CoLA task.

Surprisingly, MST performs poorly in terms of utility as compared to GST. While the exact reason for this would require an in-depth study, we posit that two takeaways can be learned: (1) maximizing PMI might not necessarily be ideal in any case and especially for privatization, and (2) the use of PMI itself may introduce issues, due to the limitations of a frequency-based association measure.

Budget Distribution An important discussion arises out of the comparative performance demonstrated by S2 and S3/4. Despite being granted on average a (much) stricter privacy budget, S2 perturbations manage to show strong performance across all tasks, having the highest score in 5 experiment scenarios and otherwise competitive scores. In essence, texts perturbed via S2 hold tighter document-level privacy guarantees than S3/4, yet they are still able to preserve utility better on average than the pure word-level perturbations of S1.

Based on these findings, we hold that further work should be afforded to investigate best practices with budget allocation, including that beyond simple "uniform" allocation given a document budget. This becomes more interesting (and potentially complex) with collocations rather than words.

Beyond F1: Similarity and Perplexity The *CS* and *AP* metrics also tell an interesting story. On average, collocation-based perturbations always result in privatized texts with higher semantic similarity, even at lower ε values. The strength of collocations is particularly made clear at higher ε values, where the gap is quite large. In contrast, the perplexity metric is split based on ε value: at lower values, word-level perturbations (S1) achieve better scores, whereas at higher scores, S3 prevails. This

disparity is insightful, prompting the further study of metric-based evaluations in privacy-preserving NLP. Qualitatively, one can argue that collocation-based perturbations produce much more coherent and readable texts, as showcased in Appendix A.

The Effect on Privacy Analyzing the empirical privacy results also brings insights. As opposed to the disparity in perplexity measurement, a *reverse* trend can be observed with empirical privacy. At lower ε values, collocation-based perturbations achieve comparable or better privatization against adversaries, whereas this advantage begins to favor word-level approaches at higher privacy budgets. However, the strength of word-level approaches at higher budgets comes with the cost of severely limited utility, as shown by both tasks.

The *relative gain* results show that in none of the tested scenarios, a positive gain can be observed using word-level perturbations. This comes in contrast to strategies S2-4, which often show positive gains, and achieve the highest relative gain in all but one scenario. These results are promising in the way that MLDP mechanisms can be made practically feasible when leveraging collocations.

As a final analysis, we observe that collocation embedding models enable greater diversity in privatization outputs. Taking the vocabulary of DEBERTA-V3-BASE (128k tokens), we discover that while only 68,544 unigram tokens from our GST model exist in the vocabulary, *1,248,304* tokens from the model match the vocabulary, i.e., where *every* word exists in the vocabulary. This allows for a wider search space, thus presumably reducing cases where a token is perturbed to itself.

Replication on Other Mechanisms We replicate the SST2 utility experiments on two other MLDP mechanisms, the Mahalanobis Mechanism (Xu et al., 2020) and the Vickrey Mechanism (Xu et al., 2021b). These results are shown in Tables 5 and 6. The results mirror those described in this work, albeit with an interesting anomaly observed with the Vickrey Mechanism at lower ε values. We perform this extra analysis as a first step towards generalizing our results to all MLDP mechanisms, in order to investigate the advantages of multi-word rather than single word DP perturbations.

7 Conclusion

In this work, we present an alternative to word-level Metric Differential Privacy, which differs in the way that we tokenize and privatize sensitive input texts on the *collocation* level. We provide two collocation extraction algorithms and their corresponding trained embedding models, showing how word-level MLDP mechanisms can be simply augmented to operate on this higher syntactic level. In our evaluation, we demonstrate the merits of such augmentation, achieving a balance between improved utility, higher semantic coherence, and comparable privacy preservation.

The results provide researchers with two overarching insights. Using collocations, given the same *overall* budget for a document, we can achieve higher utility while still preserving privacy. At the same time, given the same *per-token* budget, perturbing collocations often outperforms word-by-word privatization. Thus, we make the case that further studies in the field of DP NLP should consider investigating linguistic units outside of the standard word- or sentence-/document-level.

The main limitations of our study come with our reliance on one particular measure for collocation extraction, namely PMI. In addition, we focus on validating our method for the MADLIB mechanism, but do not perform extensive testing on more recent methods. Finally, we base our results on the selected datasets for utility and privacy, whereas this would be well-served to be more extensively tested. As such, we propose the following paths for future work: (1) a focus on collocations and their reliable extraction for DP applications, (2) further work on validating the merits of privatization between the word and sentence level, and (3) deeper investigations into the rigorous evaluation of DP text privatization, with an emphasis on metrics.

Baseline	$94.33_{0.2}$			
ε	0.1	1	10	25
S1	$\mathbf{56.0_{3.6}}$	$\mathbf{56.4_{3.9}}$	$58.7_{0.7}$	$64.6_{0.4}$
S2	$51.1_{0.3}$	$55.4_{1.7}$	$76.2_{0.8}$	$89.5_{0.4}$
S3	$50.9_{0.1}$	$54.4_{2.0}$	$\mathbf{82.6_{0.8}}$	$\mathbf{91.5_{0.3}}$
S4	$52.6_{2.4}$	$53.9_{2.2}$	$65.6_{0.2}$	$71.9_{0.7}$

Table 5: SST2 Utility Results, using the Mahalanobis Mechanism (Xu et al., 2020), with $\lambda = 0.2$.

Baseline	$94.33_{0.2}$			
ε	0.1	1	10	25
S1	$\mathbf{83.0_{1.1}}$	$\mathbf{81.3_{0.1}}$	$67.4_{0.8}$	$61.5_{7.1}$
S2	$50.9_{0.0}$	$56.1_{1.8}$	$71.8_{0.4}$	$78.7_{0.2}$
S3	$51.0_{0.1}$	$53.2_{3.2}$	$\mathbf{74.8_{1.5}}$	$\mathbf{79.8_{0.6}}$
S4	$53.0_{1.3}$	$55.2_{1.6}$	$64.7_{0.6}$	$67.3_{1.4}$

Table 6: SST2 Utility Results, using the Vickrey Mechanism (Xu et al., 2021b), using the two neighbor variant.

Acknowledgments

The authors would like to thank the anonymous reviewers for their time and feedback and Alexandra Klymenko for her valuable contributions.

Limitations

The main limitations regarding our experimental setup include the use of only one metric for automatic collocation extraction. In addition, we do not clean or filter the outputs of the collocation extraction process, outside of our set threshold of $PMI \geq 2$. The effect of performing extra cleaning steps, or by using entirely different collocation extraction methods, remains a point for future work.

Another point is the limited testing in terms of MLDP mechanisms. We decided to test extensively on one mechanism (MADLIB) rather than conduct more limited tests on a variety of mechanisms. Although we provide initial insights into the effect on other mechanisms, further testing is needed.

Finally, we acknowledge the distinction between measured results of *empirical privacy* versus true privacy preservation, and although the former is a good proxy for the latter, there is still work to be done regarding the nature of privacy in textual data.

References

Xiao Bai, Erik Ordentlich, Yuanyuan Zhang, Andy Feng, Adwait Ratnaparkhi, Reena Somvanshi, and Aldi Tjahjadi. 2018. Scalable query n-gram embedding for improving matching and relevance in sponsored search. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18, page 52–61, New York, NY, USA. Association for Computing Machinery.

Vishal Bhalla and Klara Klimcikova. 2019. Evaluation of automatic collocation extraction methods for language learning. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 264–274, Florence, Italy. Association for Computational Linguistics.

Gerlof Bouma. 2010. Collocation extraction beyond the independence assumption. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 109–114, Uppsala, Sweden. Association for Computational Linguistics.

Vaclav Brezina, Tony McEnery, and Stephen Wattam. 2015. Collocations in context: A new perspective on collocation networks. *International Journal of Corpus Linguistics*, 20(2):139–173.

Ricardo Silva Carvalho, Theodore Vasiloudis, Oluwaseyi Feyisetan, and Ke Wang. 2023. TEM: High utility metric differential privacy on text. In *Proceedings of the 2023 SIAM International Conference on Data Mining (SDM)*, pages 883–890. SIAM.

Konstantinos Chatzikokolakis, Miguel E. Andrés, Nicolás Emilio Bordenabe, and Catuscia Palamidessi. 2013. Broadening the scope of differential privacy using metrics. In *Privacy Enhancing Technologies*, pages 82–102, Berlin, Heidelberg. Springer Berlin Heidelberg.

Hui Chen, Fengran Mo, Yanhao Wang, Cen Chen, Jianyun Nie, Chengyu Wang, and Jamie Cui. 2022. A customized text sanitization mechanism with differential privacy. In *Annual Meeting of the Association for Computational Linguistics*.

Kenneth Ward Church and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1):22–29.

John C. Duchi, Michael I. Jordan, and Martin J. Wainwright. 2013. Local privacy and statistical minimax rates. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 429–438.

Cynthia Dwork. 2006. Differential privacy. In *International colloquium on automata, languages, and programming*, pages 1–12. Springer.

Luis Espinosa Anke, Joan Codina-Filba, and Leo Wanner. 2021. Evaluating language models for the retrieval and categorization of lexical collocations. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1406–1417, Online. Association for Computational Linguistics.

Stefan Evert. 2009. *58. Corpora and collocations*, pages 1212–1248. De Gruyter Mouton, Berlin, New York.

Natasha Fernandes, Mark Dras, and Annabelle McIver. 2019. Generalised differential privacy for text document processing. In *Principles of Security and Trust: 8th International Conference, POST 2019, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2019*, pages 123–148. Springer International Publishing.

Oluwaseyi Feyisetan, Abhinav Aggarwal, Zekun Xu, and Nathanael Teissier. 2021. Research challenges in designing differentially private text generation mechanisms. In *The International FLAIRS Conference Proceedings*, volume 34.

Oluwaseyi Feyisetan, Borja Balle, Thomas Drake, and Tom Diethe. 2020. Privacy- and utility-preserving textual analysis via calibrated multivariate perturbations. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, WSDM '20, page 178–186, New York, NY, USA. Association for Computing Machinery.

Prakhar Gupta, Matteo Pagliardini, and Martin Jaggi. 2019. Better word embeddings by disentangling contextual n-gram information. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 933–939, Minneapolis, Minnesota. Association for Computational Linguistics.

Dirk Hovy, Anders Johannsen, and Anders Søgaard. 2015. User review sites as a resource for large-scale sociolinguistic studies. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15, page 452–461, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

Lijie Hu, Ivan Habernal, Lei Shen, and Di Wang. 2024. Differentially private natural language models: Recent advances and future directions. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 478–499, St. Julian's, Malta. Association for Computational Linguistics.

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.

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.

Jacqueline Léon. 2005. Meaning by collocation. *History of linguistics*, pages 404–415.

Christopher Manning and Hinrich Schutze. 1999. *Foundations of statistical natural language processing*. MIT press.

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

Kathleen R McKeown and Dragomir R Radev. 2000. Collocations. *Handbook of Natural Language Processing. Marcel Dekker*, pages 1–23.

Stephen Meisenbacher, Maulik Chevli, and Florian Matthes. 2024a. 1-Diffractor: Efficient and utility-preserving text obfuscation leveraging word-level metric differential privacy. In *Proceedings of the 10th ACM International Workshop on Security and Privacy Analytics*, IWSPA '24, page 23–33, New York, NY, USA. Association for Computing Machinery.

Stephen Meisenbacher, Nihildev Nandakumar, Alexandra Klymenko, and Florian Matthes. 2024b. A comparative analysis of word-level metric differential privacy: Benchmarking the privacy-utility trade-off. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 174–185, Torino, Italia. ELRA and ICCL.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Pavel Pecina. 2005. An extensive empirical study of collocation extraction methods. In *Proceedings of the ACL Student Research Workshop*, pages 13–18, Ann Arbor, Michigan. Association for Computational Linguistics.

Pavel Pecina and Pavel Schlesinger. 2006. Combining association measures for collocation extraction. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 651–658, Sydney, Australia. Association for Computational Linguistics.

Adam Poliak, Pushpendre Rastogi, M. Patrick Martin, and Benjamin Van Durme. 2017. Efficient, compositional, order-sensitive n-gram embeddings. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 503–508, Valencia, Spain. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

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.

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta. ELRA. `http://is.muni.cz/publication/884893/en`.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Tanmay Singh, Harshvardhan Aditya, Vijay K Madisetti, and Arshdeep Bahga. 2024. Whispered tuning: Data privacy preservation in fine-tuning llms through differential privacy. *Journal of Software Engineering and Applications*, 17(1):1–22.

Saiteja Utpala, Sara Hooker, and Pin-Yu Chen. 2023. Locally differentially private document generation using zero shot prompting. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8442–8457, Singapore. Association for Computational Linguistics.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Shufan Wang, Laure Thompson, and Mohit Iyyer. 2021. Phrase-BERT: Improved phrase embeddings from BERT with an application to corpus exploration. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10837–10851, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Benjamin Weggenmann, Valentin Rublack, Michael Andrejczuk, Justus Mattern, and Florian Kerschbaum. 2022. DP-VAE: Human-readable text anonymization for online reviews with differentially private variational autoencoders. In *Proceedings of the ACM Web Conference 2022*, WWW '22, page 721–731, New York, NY, USA. Association for Computing Machinery.

Nan Xu, Oluwaseyi Feyisetan, Abhinav Aggarwal, Zekun Xu, and Nathanael Teissier. 2021a. Density-aware differentially private textual perturbations using truncated gumbel noise. In *The International FLAIRS Conference Proceedings*, volume 34.

Zekun Xu, Abhinav Aggarwal, Oluwaseyi Feyisetan, and Nathanael Teissier. 2020. A differentially private text perturbation method using regularized mahalanobis metric. In *Proceedings of the Second Workshop on Privacy in NLP*, pages 7–17.

Zekun Xu, Abhinav Aggarwal, Oluwaseyi Feyisetan, and Nathanael Teissier. 2021b. On a utilitarian approach to privacy preserving text generation. In *Proceedings of the Third Workshop on Privacy in Natural Language Processing*, pages 11–20.

Wenpeng Yin and Hinrich Schütze. 2014. An exploration of embeddings for generalized phrases. In *Proceedings of the ACL 2014 Student Research Workshop*, pages 41–47, Baltimore, Maryland, USA. Association for Computational Linguistics.

Wenpeng Yin and Hinrich Schütze. 2015. Discriminative phrase embedding for paraphrase identification. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1368–1373, Denver, Colorado. Association for Computational Linguistics.

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

Xiang Yue, Huseyin Inan, Xuechen Li, Girish Kumar, Julia McAnallen, Hoda Shajari, Huan Sun, David Levitan, and Robert Sim. 2023. Synthetic text generation with differential privacy: A simple and practical recipe. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1321–1342, Toronto, Canada. Association for Computational Linguistics.

Jiajun Zhang, Shujie Liu, Mu Li, Ming Zhou, and Chengqing Zong. 2014. Bilingually-constrained phrase embeddings for machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 111–121, Baltimore, Maryland. Association for Computational Linguistics.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Zhe Zhao, Tao Liu, Shen Li, Bofang Li, and Xiaoyong Du. 2017. Ngram2vec: Learning improved word representations from ngram co-occurrence statistics. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 244–253, Copenhagen, Denmark. Association for Computational Linguistics.

A Appendix

Collocation Examples Table 7 presents a sample of six randomly selected tokens from our GST-extracted collocation embedding model, as well as the five nearest neighbors in the space. Note that for any given token, a nearest neighbor need not be the same "length" token, i.e., a unigram's nearest neighbor may include bigrams or trigrams.

Document-level Budgets As described in Section 5.2, to utilize our selected "base" ε values, we scale the privacy budget allotted to each tested dataset. In Table 8, we tabulate all document budgets, which are calculated by multiplying the average words per text by the base ε values.

Examples Table 9 shows selected privatization outputs from two datasets using MADLIB with the privatization strategies S1-4. For readability, we strip sentence punctuation marks, and we select five ε values for illustration. Some inappropriate words have been redacted.

	Tokens					
	machinerytrader	mahatma	elise	festival_itself	wordwide_market	certificates_of_completion
Most similar tokens	crusher_aggregate_equipment	gandhiji	anna	whole_festival	global_market	course_certificate
	portable_cone_crusher	swami_vivekananda	aimee	this_festival	worldwide_markets	training_certificates
	aggregate_equipment	bapuji	julia	festival_weekend	growing_market	training_certificate
	equipmentmine	babasaheb	sarah	festival_week	this_market_segment	graduation_certificate
	bucket_crusher	savarkar	megan	festival_period	massive_market	their_certificate

Table 7: Token examples from the **GST** collocation embedding model. Shown are randomly selected tokens from the model, along with their five most similar tokens in the embedding space.

		Document Budget (ε)							
Dataset	Avg. Words/Text	0.1	0.5	1	5	10	15	25	50
CoLA	7.80	0.78	3.9	7.8	38.99	77.99	116.98	194.96	389.93
MRPC	19.54	1.95	9.77	19.54	97.72	195.44	195.44	488.6	977.21
RTE	44.48	4.45	22.24	44.48	222.41	444.82	667.23	1112.06	2224.12
SST2	8.82	0.88	4.41	8.82	44.11	88.22	132.33	220.56	441.12
Trustpilot	52.16	5.22	26.08	52.16	260.81	521.61	782.42	1304.03	2608.05
Yelp	186.87	18.69	93.43	186.87	934.34	1868.68	2803.02	4671.7	9343.41

Table 8: Document-level budgets. Given our base ε values, we scale the allocated overall budget per document based on the average token length of documents in each dataset. The resulting budgets are thus shown in the table.

		Original text:
	ε	this deal makes sense for both companies halla said in a prepared statement
S1	0.1	ridership rhp [REDACTED] hypothalamus [REDACTED] chiller rm ridership warhead ridership a cyberattacks [REDACTED]
	0.5	chiller chiller ridership lf xp chiller comeuppance [REDACTED] affections rm a [REDACTED] [REDACTED]
	1	quercetin chiller cyberattacks unsecure dropkick affections backrest [REDACTED] galaxies transcriptional a comeuppance creole
	5	ridership counselor flicker shekels fences sconces rm lidocaine aerodynamics housemates a questionnaires libretto
	10	savings hovers occasions dough photographing housemate restrictions renminbi lotion condemning a batsman genocide
S2	0.1	rbis are worthy especially true who didn animal ' knockon effect damages that up to 15 alzheimer ' particularly the case baha ' s most recent
	0.5	enjoyed every dry cleaned domino effect all u multimeter enjoyed every vicious circle vicious cycle audiences who chose marijuana use especially true
	1	up to 15 especially true potter ' s publics enjoy reading book consumers ' found your blog chain of events attempt missed i enjoyed reading forward to reading posted june
	5	extract of sample deal that was makes sense poker action both companies 154 receiving means holm shapleigh found across 09
	10	said loudly amazon which makes sense such as gym both firms le film halla said in a prepared statement
S3	0.1	true even something i could yearold has glad it particularly evident line dry later went particularly the case extremely satisfied publics machine wash change has
	0.5	captcha is if nothing true even machinewashable chilling effect nonconference static display is gluten they sleep loved every mile trail gentle cycle
	1	judged that deet belong on this mitzvot publics weather ' s blood group its traditions you woke even take especially useful california who
	5	said anna this new agreement makes sense custom construction both sectors marzi 5 responses emily rose announced " within the garden a prepared statement
	10	any deal that makes sense for both entities thats the truth halla said in a prepared statement
S4	0.1	t going t think breakfast t see click when t hesitate when i ' ve look forward is made
	0.5	he had ' d may not will not his wife t be would have t want as t get populations it
	1	filed under diameter exchange relationship between tax smaller ° c campaign master very difficult have not like
	5	its seems like for plan that seasoned instead said in an easy third floor
	10	£ 1 makes sense job search staffers clarinet brokerage firms other said in an excellent immigration and customs

Table 9: Privatization samples from MRPC.