Department of Mathematics
TUM School of Computation, Information and Technology
Technical University of Munich

Studying the Privacy-Utility Trade-off of Word Embedding Perturbations with a Focus on Sensitivity Analysis and Vector Mapping

Alisha Riecker

Thesis for the attainment of the academic degree

Master of Science

at the TUM School of Computation, Information and Technology of the Technical University of Munich

Supervisor:
Prof. Dr. Florian Matthes

Advisors:
Stephen Meisenbacher, M.Sc.

Submitted:
Munich, 14th December 2023

I hereby declare that this thesis is entirely the result of my own work except where otherwise indicated. I have only used the resources given in the list of references.

Munich, 14th December 2023 Alisha Riecker

Zusammenfassung

Bei der Anwendung von Differential Privacy (DP)-Techniken zur Perturbation von Worteinbettungen sind die Verbesserung der Privatsphäre und die Beibehaltung der Nützlichkeit zwei gegensätzliche Ziele. In dieser Arbeit wird untersucht, wie Worteinbettungen mit Hilfe von Differential Privacy effizient perturbiert werden können. Im Mittelpunkt steht dabei, die Auswirkungen verschiedener Ansätze zur Begrenzung der Sensitivität und des Vektor-Mappings auf die Privatsphäre und die Nützlichkeit zu analysieren. Wir untersuchen die theoretische Perspektive mithilfe theoretischer Privatsphäre-Garantien und führen empirische Experimente durch, um die Auswirkungen der Ansätze auf nachgelagerte Natural Language Processing (NLP)-Aufgaben zu beleuchten. Die Ergebnisse unserer Experimente zeigen, dass die Auswirkungen auf den Kompromiss zwischen Privatsphäre und Nutzen für verschiedene Datensätze, Aufgaben, DP-Mechanismen und Modelle sehr unterschiedlich sind. Die Wirkung der meisten Ansätze beschränkt sich auf eine allgemeine Ergebnisverbesserung der NLP-Aufgaben. Diese Verbesserungen sind jedoch nicht unbedingt mit einer Verbesserung des Kompromisses zwischen Privatsphäre und Nutzen verbunden.

Abstract

When applying Differential Privacy (DP) techniques to perturb word embeddings, enhancing privacy and maintaining utility are two conflicting objectives. This study examines how word embedding vectors can be efficiently perturbed using differential privacy. At the center of this, we will analyze the impact of different approaches for bounding sensitivity and vector mapping on privacy and utility. We examine the theoretical perspective through theoretical privacy guarantees and perform empirical experiments to illuminate the approaches' effects on downstream Natural Language Processing (NLP) tasks. Our experiment results show that the effect on the privacy-utility trade-off largely differs for different datasets, tasks, DP mechanisms, and models. The effect of most approaches is limited to a general improvement in performance on the tasks, which is, however, not necessarily linked to an improvement with respect to the privacy-utility trade-off.

Contents

1 Introduction

Generative Artificial Intelligence and especially Large Language Models have become omnipresent in recent years. More and more systems are based on or integrate such models. To train language models, lots of text data is required. The entity training the model is often separate from the texts' contributors or authors. As the texts might contain sensitive information, the contributors might be hesitant to share them with a third party. Thus, privacy concerns often hinder the development of new models or tailoring models to specific use cases and data. A straightforward approach to hiding private information would be sharing only vector representations, also called word embedding vectors, with a third party instead of the original texts. This method, however, is insufficient to hide the private information contained in the input texts [SR20]. Even though the vectors can mostly not be interpreted by a human without additional information, it is still possible to recover the original texts from the vector representations. Further methods to hide private information are hence required. This has led to the adoption of DP to the NLP domain with the goal of providing privacy guarantees for word vector representations.
Using DP, one can add noise to word embedding vectors and thereby perturb and privatize them. Adding more noise leads to stronger theoretical privacy guarantees [Xu+21a]. At the same time, adding more noise can impair the word's semantics encoded in the word embedding vector and can harm the performance of language models, which are trained with the perturbed word embedding vectors [Xu+21a]. Therefore, it is important for the creation of effective privatized word embeddings to choose the amount of noise in a way that provides as much privacy as possible while also harming the utility of the word embedding vectors as little as possible. Previous research has used two methods to improve this trade-off between privacy and utility. One is to limit the sensitivity of the embedding vectors before noise addition [FK21; LHL20; Mah+22]. Roughly speaking, sensitivity in this context describes the maximum distance between two embedding vectors [FK21]. The other method to improve the privacy-utility trade-off is mapping the perturbed vectors after noise addition to another close-by word embedding vector from a fixed vocabulary [Fey+20; Xu+21b]. There are different approaches for both methods. This thesis aims to explore the impact of the different approaches on the trade-off between privacy and utility.
The following research questions have been defined to guide the achievement of the goal previously specified:

1. What approaches are there to privatize word embeddings by perturbing word vector representations?
2. How can we make these privatized word embeddings more effective?
3. What is the effect of different approaches to bounding sensitivity on privacy and utility for downstream NLP tasks?
4. What are the implications on privacy and utility resulting from mapping noisy word embeddings to similar embedding vectors which are associated with real words?

The first research question will look into existing research to identify DP mechanisms, which can be applied to perturb word embedding vectors. To answer the second research question, the goal will be to find out which approaches can yield a better calibration of the DP noise or impose stronger privacy guarantees. We will specifically focus on approaches to bounding sensitivity or mapping perturbed word embedding vectors to close-by vectors associated with a word from the vocabulary. For the third and fourth research questions, we will investigate these approaches' effects on privacy and utility. Therefore, we will first shed light on this from a theoretical perspective and analyze how the different approaches influence the

amount of noise necessary to ensure DP and how their application affects theoretical privacy guarantees. Afterwards, the approaches will be implemented in practical experiments to observe the implications for NLP tasks.
After this introductory chapter, Chapter 2 of this work will elaborate on the theoretical foundations around DP, its application to the NLP domain, and the privacy-utility trade-off. In Chapter 3, existing research, that is relevant to this work, will be discussed. We will especially concentrate on DP mechanisms as well as the approaches for bounding sensitivity and vector mapping that are used. Chapter 4 will describe the methodology for the practical experiments. The results and the implications on the privacy-utility trade-off corresponding to experiments on approaches for bounding sensitivity will be presented in Chapter 5 and the results for the vector mapping approaches will be detailed in Chapter 6. Following this, Chapter 7 will discuss the main findings as well as limitations, which need to be kept in mind when interpreting the experiments' results. Furthermore, this chapter will state potential directions for future work. Finally, this work concludes with a summary in Chapter 8.

2 Foundations

Before practically applying different approaches for DP in Chapter 4, the theoretical foundations of DP and relevant mechanisms will be introduced in this chapter. This chapter will also contain an introduction to the approaches for bounding sensitivity and vector mapping, which will be compared to each other in the later experiments.

2.1 Differential Privacy

DP was first introduced by [Dwo+06] for the privatization of data stored in databases. It allows to gain insights about a population described in a database while preventing any data instance from being unambiguously linked to a specific individual whom it refers to [Dwo+06]. In the context of databases, the concept of DP is defined for two neighboring databases D and D'.

Definition 2.1.1 (Neighboring databases)**.** *Let D and D' be two databases from the set of all possible databases $\mathcal{D}$ with records v_i, v_i' for $i \in [n]$. The Hamming distance between two databases D and D' is $d_H(D, D') = \sum_{i=1}^{n} \left|v_i - v_i'\right|$.*
It represents the number of records on which D and D' differ, i.e., $d_H(D, D') = \left|i : v_i \neq v_i'\right|$.
Two databases are called neighboring or adjacent if $d_H(D, D') = 1$, i.e., they differ in only one record.

Using the above definition of neighboring databases, we can now formally define DP.

Definition 2.1.2 (ϵ-Differential Privacy (ϵ-DP))**.** *Let $\epsilon \in \mathbb{R}_0^+$. A randomized function $\mathcal{A}$ is ϵ-differentially private if for all databases D and D' differing in at most one record (i.e., with $d_H(D, D') = 1$), and possible outputs y of $\mathcal{A}$ (i.e. $y \in Range(\mathcal{A})$),*

$$\mathbb{P}[\mathcal{A}(D) = y] \leq \exp(\epsilon) \cdot \mathbb{P}[\mathcal{A}(D') = y] \tag{2.1}$$

This means that $\mathcal{A}$ fulfills ϵ-DP if its output differs at most by a multiplicative factor of $\exp(\epsilon)$ when applied to databases differing in at most one record. Thus, Definition 2.1.2 provides indistinguishability for the two inputs D and D', meaning that someone who is only presented the output of the mechanism cannot distinguish if the original input was D or D'. This explains why ϵ-DP is sometimes called ϵ-indistinguishability [Dwo+06]. We can say that the smaller the ϵ value, the stronger the indistinguishability, i.e., the harder it is to distinguish between potential inputs. If $d_H(D, D')$ is not equal to one for all pairs of inputs, then the above definition can also be applied transitively to provide a privacy guarantee [Cha+13]. The randomized function $\mathcal{A}$ is also called *(privacy) mechanism* in the context of DP and the parameter ϵ is referred to as the *privacy budget*. The privacy budget governs the amount of noise added and quantifies the strength of the privacy guarantee [Fey+20]. Privacy budget and privacy guarantee are inversely related. The smaller ϵ, the more noise is added, and the more indistinguishable and the more protected are the two inputs D and D' [Hu+23]. For $\epsilon \to 0$, the mechanism provides absolute privacy as its output becomes independent of the input. $\epsilon \to \infty$ describes the absence of privacy where $\mathcal{A}(D) = D$ [Fey+20; Xu+20]. In some cases, ϵ-DP is too strict because it demands the inequality in Definition 2.1.2 to be fulfilled even for outlier data points [Hoo+21]. The definition of ϵ-DP is sometimes relaxed to (ϵ, δ)-Differential Privacy (also *Approximate DP*):

Definition 2.1.3 ((ϵ, δ)-Differential Privacy ((ϵ, δ)-DP))**.** *Let $\epsilon \in \mathbb{R}_0^+$ and $\delta \in [0, 1] \cap \mathbb{R}$. A randomized function $\mathcal{A}$ is (ϵ, δ)-differentially private if for all databases D and D' differing in at most one record (i.e., with $d_H(D, D') = 1$), and for all possible outputs y (i.e. $y \in Range(\mathcal{A})$),*

$$\mathbb{P}[\mathcal{A}(D) = y] \leq \exp(\epsilon) \cdot \mathbb{P}[\mathcal{A}(D') = y] + \delta$$

(ϵ, δ)-DP relaxes ϵ-DP such that the mechanism outputs on two neighboring databases are allowed to additionally differ by an additive *delta* value. Hence, δ controls the strength of the relaxation. The smaller this scalar is, the stronger the relaxation. It can be interpreted as the probability of two inputs not fulfilling the privacy guarantee given by regular ϵ-DP (Definition 2.1.2) [Hu+23]. Therefore, we want to see δ as small as possible. For $\delta = 0$, approximate DP falls back to regular ϵ-DP.
When working with DP to privatize word embeddings, our inputs in the two definitions above are word embedding vectors x and x' instead of databases D and D'. A mechanism fulfilling Equation 2.1 for any two such word embedding vectors yields ϵ-DP. This means that any pair of word embeddings, that has been privatized using this mechanism, is ϵ-indistinguishable, i.e., the ratio between the probabilities that they yield the same output is bounded by $\exp(\epsilon)$. However, the above definition is less suitable for word embeddings. It enforces the same level of privacy onto every pair of inputs because the Hamming distance, and in particular the adjacency prerequisite, do not provide the possibility to adjust the privacy guarantees depending on the distance between pairs of inputs [Fey+20]. Therefore, a generalization of ϵ-DP is used to account for the distance between input pairs and additionally enable the usage of different metrics to measure this distance [Xu+20]. This generalization is called metric DP or $d_{\mathcal{X}}$-privacy. Metric-DP originates from the context of location data. The indistinguishability between two locations is scaled by their distance and, therefore, locations that are further apart are easier to distinguish [Xu+20]. The same holds for the application of metric-DP to word embeddings instead of locations. Words that are more distant in the embedding space will have higher indistinguishability and, thus, be easier to distinguish, compared to words that are closer [Car+23]. Metric-DP can be defined as follows:

Definition 2.1.4 (Metric Differential Privacy)**.** *Let $\epsilon \in \mathbb{R}_0^+$, $\mathcal{X}$ be a set with a metric $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_0$, and $f : \mathcal{X} \to \mathbb{R}^d$ a function, whose output is to be privatized. A randomized function $\mathcal{A}_f : \mathcal{X} \to \mathcal{Y}$ is $\epsilon d_{\mathcal{X}}$-differentially private if for any $x, x' \in \mathcal{X}$ the distributions over outputs of $\mathcal{A}_f(x)$ and $\mathcal{A}_f(x')$ satisfy the following bound: for all possible output $y \in \mathcal{Y}$ we have*

$$\mathbb{P}[\mathcal{A}_f(x) = y] \leq \exp(\epsilon d_{\mathcal{X}}(x, x')) \cdot \mathbb{P}[\mathcal{A}_f(x') = y]$$

The above definition shows that the privacy guarantee in metric-DP also depends on a privacy budget ϵ just as in regular DP. However, this ϵ does not represent the same privacy level as the ϵ in regular ϵ-DP [Alv+18]. Therefore, privacy guarantees need to be carefully assessed. In metric-DP, the privacy guarantee additionally depends on the metric d. When Hamming distance is used as the metric d, metric-DP reduces to regular ϵ-DP. Since this work considers word embeddings in Euclidean space, we follow [FK21] and focus on the Euclidean metric. This metric works well to capture semantic similarity between words. For the later experiments, we will only use DP mechanisms, which use Euclidean distance to allow for an easier comparison between privacy guarantees.
DP comes with some useful properties that are valid for all randomized functions that fulfill ϵ-DP, (ϵ, δ)-DP, or metric-DP. One of these properties, which will also be central to this work, is its closure under post-processing. This property guarantees that the output of a differentially private algorithm will stay differentially private if further post-processing is applied and its DP-guarantees will not incur any further privacy loss [Dwo+06]. This can be formalized by the following proposition:

Proposition 2.1.1 (Closure under post-processing)**.** *Let g be a randomized function.*
If $\mathcal{A}$ is an ϵ-differentially private mechanism, then the mechanism $\mathcal{A}' = g \circ \mathcal{A}$ is ϵ-differentially private.

This property allows to map word embedding vectors, privatized through a differentially private perturbation, to other, close-by embedding vectors without deteriorating the theoretical privacy guarantees.

2.1.1 Differential Privacy Mechanisms

Multiple randomized functions or mechanisms have been proposed in the literature to achieve metric-DP. In the context of privatizing word embedding vectors, the randomized functions perturb input words or word embedding vectors to privatized words or word embedding vectors and guarantee indistinguishability for the original input [ZC22]. Typically, these functions are parametrized by a probability distribution

[Hab21], which gives the mechanism its name. Random noise is drawn from this distribution to perturb the input. This subsection will now provide details for some selected DP mechanisms relevant to this work.

Multivariate Laplace Mechanism

One of the most frequently used DP mechanisms is the Laplace mechanism [ZC22], where noise is sampled from a Laplace distribution and added to the input. For the Laplace mechanism, we need to distinguish between the Univariate and the Multivariate Laplace mechanism. The Univariate Laplace mechanism adds noise drawn from a Univariate Laplace distribution to each element of an input vector to provide ϵ-DP. The Multivariate Laplace mechanism instead draws Laplace noise from an d-dimensional Laplace distribution and adds it to the input to yield ϵ-metric DP. Since ϵ-metric DP is more suitable for privatizing word embedding vectors, this work focuses on the Multivariate Laplace mechanism, which will later be applied as part of the experiments. Its definition can be formalized as follows:

Definition 2.1.5 (Multivariate Laplace Mechanism)**.** *Let $\epsilon \in \mathbb{R}_0^+$ and $f : \mathcal{X} \to \mathbb{R}^d$. The Multivariate Laplace mechanism is defined as $\mathcal{A}(x) = f(x) + \eta$, where η is sampled from the d-dimensional Laplace distribution with density $p(z) \propto \exp(-\epsilon\|z\|_2)$.*

Theorem 2.1.1. *The Multivariate Laplace mechanism in Definition 2.1.5 satisfies ϵ-metric DP with respect to the Euclidean distance.*

Proof. To prove the theorem, we need to bound

$$\frac{\mathbb{P}[\mathcal{A}(x) = y]}{\mathbb{P}[\mathcal{A}(x') = y]}$$

By definition of the Laplace mechanism, $\mathbb{P}[\mathcal{A}(x) = y] = \mathbb{P}[f(x) + \eta = y]$. Since η is a random variable distributed according to the density $p_\eta(z) \propto \exp(-\epsilon\|z\|_2)$, by the theorem on linear transformations of random variables, we have $p_{f(x)+\eta}(z) \propto \exp(-\epsilon\|z - f(x)\|_2)$. Thus,

$$\begin{aligned}
\frac{\mathbb{P}[f(x) + \eta = y]}{\mathbb{P}[f(x') + \eta = y]} &= \frac{\exp(-\epsilon\|z - f(x)\|_2)}{\exp(-\epsilon\|z - f(x')\|_2)} \\
&= \exp(\epsilon \cdot (\|z - f(x')\|_2 - \|z - f(x)\|_2)) \\
&\leq \exp(\epsilon \cdot \|z - f(x') - z + f(x)\|_2) \\
&\leq \exp(\epsilon \cdot \|f(x) - f(x')\|_2)
\end{aligned}$$

Now, for the metric d being defined as $d(x, x') = \|f(x) - f(x')\|_2$, we can continue as

$$\exp(\epsilon \cdot \|f(x) - f(x')\|_2) = \exp(\epsilon \cdot d(x, x'))$$

□

In the scenario considered in this work, we would like to privatize word embedding vectors. Thus, we consider the embedding model Φ as our function f, which we aim to privatize. The distance between two words will be described via the Euclidean distance between the corresponding word embedding vectors, i.e., $d(x, x') = \|\Phi(x) - \Phi(x')\|_2$. This choice for the distance metric is also reflected in the proof above so that the proof is directly transferable to the privatization of word embeddings.
To sample from the d-dimensional Laplacian, one can follow the procedure described in [Wu+17]. First, a vector v' is sampled from a d-dimensional normal distribution

$$p(x) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x - \mu)^T \Sigma^{-1}(x - \mu)\right) \tag{2.2}$$

where the mean μ is the d-dimensional zero vector to yield a zero-centered distribution. The covariance matrix Σ is the identity matrix. The vector v' is then normalized to unit length to yield a vector $v = \frac{v'}{\|v\|_2}$

following the uniform distribution on the d-dimensional unit sphere. Additionally, a noise magnitude l is sampled from a Gamma distribution with density p:

$$p(x) = \frac{x^{d-1} \exp(-\frac{x}{\theta})}{\Gamma(d)\theta^d}$$

In the above density function, $\theta = 1/\epsilon$. The final Laplacian noise can then be constructed by multiplying $\eta = lv$.

Truncated Gumbel Mechanism

The Truncated Gumbel mechanism is a density-aware word substitution mechanism, which employs a truncated Gumbel random noise for selecting amongst a list of perturbed word embeddings [Xu+21a]. It was proposed by [Xu+21a] to limit the number of nearby words to be considered as compared to the Multivariate Laplace mechanism. The procedure is motivated by the observation that with the Multivariate Laplace mechanism, there is a high probability for words in densely populated regions within the embedding space being substituted with close but irrelevant words [Xu+21a]. Thus, the Truncated Gumbel mechanism contains another sampling step, which leads to a narrower selection of potential substitute words with similar meanings as the original word. This, in turn, yields better preservation of semantic meaning during word perturbations and better utility of models trained on those perturbed words in downstream NLP tasks compared to the Multivariate Laplace mechanism [Xu+21a]. The complete procedure for perturbing a single input word is described in Algorithm 1. Algorithm 1 first sets the number of candidate

Algorithm 1 Truncated Gumbel Mechanism

Input: Word $x \in \mathcal{W}$, privacy budget ϵ, word set $\mathcal{W}$

Let $\Delta_{max} = \max_{x,x' \in \mathcal{W}} \|\Phi(x) - \Phi(x')\|_2$ be the maximum inter-word distance,
and $\Delta_{min} = \min_{\substack{x,x' \in \mathcal{W}, \\ x \neq x'}} \|\Phi(x) - \Phi(x')\|_2$ be the minimum inter-word distance within the embedding space.
Let $b = 2\Delta_{max} \cdot \frac{1}{min\{W(2\alpha\Delta_{max}), \ln(\alpha\Delta_{min})\}}$, where $\alpha = \frac{1}{3}\left(\epsilon - \frac{2(1+ln|\mathcal{W}|)}{\Delta_{min}}\right)$ and W denotes the principal branch of the Lambert-W function.

Sample $k \sim TruncatedPoisson(ln|\mathcal{W}|; 1, |\mathcal{W}|)$
Find the top k closest words to x as $u = [u_1, ..., u_k]$, where $u_1 = x$
Compute the distances $d = [d_1, ..., d_k]$, where $d_j = \|x - u_j\|_2$ for all $j \in [k]$
Sample $g_1, ..., g_k \overset{i.i.d.}{\sim} TruncatedGumbel(0, b, \Delta_{max})$
Set $\tilde{x} = u_j$, where $j = argmin\{d_1 + g_1, d_2 + g_2, ..., d_k + g_k\}$
Return $\tilde{x}$

substitution words based on a value k sampled from a Truncated Poisson distribution. Then, the $k-1$ nearest neighbors to the input word x with respect to the Euclidean distance are determined. Their respective distances to x are also saved for subsequent steps. Next, the distances to those candidate substitutions are perturbed using i.i.d. random variables, sampled from a truncated Gumbel distribution. This distribution is scaled using the privacy budget ϵ, and the maximum and minimum inter-word distance Δ_{max} and Δ_{min} within the embedding space. Additionally, the random variable drawn from this distribution is clipped according to a truncation parameter C, which is set to be the maximum inter-word distance Δ_{max}. The selection from the set of candidate embeddings is made based on the smallest perturbed distance to the original input word x. The algorithm yields a perturbed word embedding vector and associated real word $\tilde{x}$.

Theorem 2.1.2. *The Truncated Gumbel mechanism (Algorithm 1) satisfies ϵ-metric DP with respect to the Euclidean distance.*

The proof for the above theorem can be found in [Xu+21a].

2.1.2 Sensitivity

For mechanisms like the Laplace mechanism, its privacy-utility trade-off can be improved by calibrating the amount of noise added by the mechanism to the input sensitivity [Dwo+06], which is defined as follows

Definition 2.1.6 (L_p-Sensitivity)**.** *The (global) sensitivity of a function $f : \mathcal{X} \to \mathbb{R}^d$ is defined as*

$$\Delta_f = \max_{x,x' \in \mathcal{X}} \frac{\|f(x) - f(x')\|_p}{\|x - x'\|_p} \tag{2.3}$$

If the norm used in equation (2.3) is the Euclidean norm (L_2-norm), then we also speak of L_2-sensitivity.

For the Univariate Laplace mechanism, it is crucial to bound the L_1-sensitivity as we can otherwise not ensure ϵ-DP. For the Multivariate version of the mechanism, a bounded L_2-sensitivity provides the possibility to use this additional knowledge about our embedding space to calibrate the noise added by the mechanism accordingly. This means that instead of sampling the noise η from the distribution defined in Definition 2.1.5, we sample from a d-dimensional Laplace distribution with density $p(z) \propto \exp\left(-\frac{\epsilon}{\Delta_f}\|z\|_2\right)$. This provides an alternative, calibrated version of the Multivariate Laplace mechanism. This mechanism is for example used by [FK21], and satisfies ϵ-metric DP. However, working with the calibrated Multivariate Laplace mechanism limits the comparability of results. Even though it also provides ϵ-metric DP with respect to the same metric as the regular Multivariate Laplace mechanism, the ϵ values in their guarantees are not comparable since the amount of noise used by the mechanisms are different. Therefore, for this thesis, we choose not to calibrate the noise to the sensitivity even for the cases where it would be possible due to bounded sensitivity. Consequently, we can compare theoretical privacy guarantees more straightforwardly. For example, it allows us to compare versions of the mechanism working without bounded sensitivity to versions working with bounded sensitivity. In cases where we have bounded sensitivity but do not calibrate the noise accordingly, the sensitivity is reflected in the theoretical privacy bound. This can be illustrated by the following example:

Example 1. Consider a set of word embedding vectors in $\mathbb{R}^d$, generated using an embedding model Φ. We would now like to apply some transformation function f to these embeddings before privatizing them using the Multivariate Laplace mechanism $\mathcal{A}$ and training some NLP model on those embedding vectors. Such a transformation could for example be a normalization or dimensionality reduction. While we will show the concrete sensitivity bounds for such transformations in Chapter 4, we will assume $\Delta_f \leq C$, where $C \in \mathbb{R}$, for the purpose of this illustration. Using this bound and the Multivariate Laplace mechanism without calibrating the noise (η is sampled from a distribution with density $p_\eta(z) \propto \exp(-\epsilon\|z\|_2)$), we can provide $\epsilon \cdot C$-metric DP. Let $\Phi(x)$ and $\Phi(x')$ be two word embedding vectors and y a perturbed word embedding vector output by the Multivariate Laplace mechanism. Then,

$$\begin{aligned}\frac{\mathbb{P}[f(\Phi(x)) + \eta = y]}{\mathbb{P}[f(\Phi(x')) + \eta = y]} &\leq \exp\left(\epsilon \cdot \|f(x) - f(x')\|_2\right) \\ &\leq \exp\left(\epsilon \cdot C \cdot \|\Phi(x) - \Phi(x')\|_2\right)\end{aligned}$$

The second inequality follows from the bound on sensitivity. Since the noise in this example has not been calibrated to the sensitivity bound we could directly compare this theoretical privacy guarantee to others achieved in scenarios where there was no bound on sensitivity and we would not even have had the option to calibrate the noise.

However, bounding sensitivity does not only have beneficial consequences for the Laplace mechanism. Since the sensitivity largely depends on the possible distances between arbitrary pairs of inputs, the problems of determining a bound on sensitivity and on the maximum and minimum inter-word distances are related. Thus, a bound on the sensitivity also affects the calibration of noise in the Truncated Gumbel mechanism. This work examines different possibilities for such a transformation function f to artificially bound sensitivity before input to a DP mechanism. This will provide further options for targeted calibration of the mechanism's noise and is expected to affect privacy and utility in downstream NLP tasks. The extent of these effects will be examined from a theoretical perspective as well as through practical experiments.

2.2 Privacy

DP techniques are often loosely described as privacy-providing. However, to better understand the guarantees that DP brings with it, a more differentiated view of the term privacy is necessary. Privacy is a multifaceted concept, which [Boj+17] divide into three key components: transparency and consent, data minimization, and anonymization of released aggregates [Bon+22]. The principle of transparency and consent refers to users of a product or service understanding the usage of their data and approving it. The objective of data minimization is to limit access to raw data at all stages throughout a computation. Data anonymization entails that a released computation output does not reveal anything unique to an individual. Out of these three components, DP addresses data anonymization [Pon+23]. It allows to reason about data anonymization in a formal, quantitative way and to provide anonymization guarantees. In this work, we will use the term privacy to refer to the anonymization guarantees provided for text data used in training models on NLP tasks. Since this work applies DP on a word-level to perturb word embedding vectors, it provides anonymization on a word-level. Following the alternative description of DP, it grants indistinguishability to individual words, meaning that, after perturbation, one cannot be sure which word was the original input word. This is also often described as plausible deniability about the original input.
Privacy guarantees provided by DP can be assessed from two perspectives. On the one hand, one can formally prove a bound like in Equation 2.1 for a specific mechanism to reason about its theoretical privacy guarantees. Since this bound is defined via the privacy budget ϵ, this parameter is used to discuss and related theoretical privacy guarantees. On the other hand, privacy guarantees can be practically assessed by applying the respective mechanism to a concrete task that one would attribute to an adversary. In the context of NLP, such an adversarial task can, for example, be the identification of pseudo-private information from texts. Pseudo-private means that the information is not actually sensitive for the concrete task at hand, however, it is easy to imagine that the same type of information could be sensitive in a different context. Thereby, one can simulate an adversarial setting to practically test the privacy guarantees without actually endangering sensitive information. Many researchers use empirical privacy to quantify these privacy guarantees. [CNC18; LHL20] use $1 - X$, where X is a performance measure of the simulated adversary, to measure empirical privacy. A higher value signals better empirical privacy.

2.2.1 Privacy-Utility Trade-off

When evaluating the effectiveness of DP methods, it is not sufficient to only consider the methods' privacy levels. Using DP methods comes with an inherent trade-off between privacy and utility such that stronger privacy guarantees entail reduced utility [Pon+23]. Thus, in addition to privacy, one needs to monitor the effects of DP methods on utility. In the best case, utility and privacy are both improved at the same time. However, as we expect utility to decrease as privacy increases, it is desirable to see a larger increase in privacy than a reduction in utility. This characterizes a favorable privacy-utility trade-off from a perspective focused on improving privacy. Most researches evaluate this trade-off by qualitatively comparing metrics for empirical privacy and utility [CNC18; LHL20]. If privacy is increased at about the same extent as utility is reduced, it can be assumed that an approach leads to a general perturbation of data and does not specifically target an improvement in the privacy-utility trade-off. To characterize the change in privacy and utility, we are comparing against baseline models.
Achieving a good trade-off is a challenging task. One of the reasons is that effects on privacy and utility in NLP tasks can only be assessed after training a model on the respective task [Pon+23]. Therefore, other influencing factors introduced during training implicate that the trade-off cannot be considered in isolation. The privacy-utility trade-off is not only influenced by the dataset size, the amount of computation used during training [Pon+23], and the design of the DP mechanism but also by its parameters such as the privacy budget ϵ. While the first three factors are usually fixed per experiment, we can vary ϵ to balance privacy and utility. This can help to assess the privacy level from a theoretic perspective if all other influencing factors are kept constant. It needs to be noted that the ϵ value is not suitable to compare different mechanisms but it can be used to assess the privacy guarantees for the same mechanism. During the later experiments, we will evaluate empirical utility and privacy on two different datasets. We use a specific

task for each dataset to determine utility and another task to determine privacy. This will help to delineate the effects of the approaches that we are testing from effects, which might originate from specificities of the different datasets or tasks.

3 Related Work

Several works among existing literature consider the application of DP to perturb vector representations of data from different domains. The following chapter will give a summary of the ones that are most relevant to this thesis. We will reference works that apply DP through embedding vector perturbations. Our focus will be on the ones that use different approaches for bounding sensitivity and vector mappings. Most of the mentioned works stem from the NLP domain but we also include works from other domains that use relevant approaches to bounding sensitivity or vector mapping.

3.1 Differential Privacy in NLP

While most existing works on DP deal with structured data, in recent years, the application of DP to unstructured text data has received increasingly more attention. Since this thesis will also apply DP to text data, we look into works from this context to learn about the characteristics of such applications. [KMM22] discuss how DP can be adapted to NLP methods and shine a light on peculiarities that arise from an application to this domain. In their work, they identify several challenges, including the need to carefully balance privacy and utility and poor explainability, i.e., explaining if the text is private. As [KMM22] elaborate, a core question in this context is what information needs to be privatized in a text. The answer to this question also influences the level at which DP is applied in NLP [Hu+23]. While some DP mechanisms can be employed on a word-, sentence- or document-level, it makes sense to focus on one particular level for more targeted privacy guarantees. For example, [FK21] or [Mah+22] use word-level DP and aim at protecting individual words. Applying DP on the word-level allows for better interpretability of the perturbed text since the influence of the perturbation can be inspected for each word. Additionally, one can theoretically provide different levels of privacy for each word depending on its individual privacy requirements. While this specific question is out of the scope of this thesis, we will consider word-level DP for its greater versatility.
Another work that focuses on DP in NLP is the work by [Hu+23]. They also look at the specificities of DP in NLP and categorize its applications into two classes. In gradient perturbation based methods, DP is provided by adding calibrated noise to the gradients of the loss during model training. In embedding vector perturbation based methods, noise is added to embedding vectors for individual tokens to guarantee DP. Since the focus of this work lies on the latter, we will take a closer look at existing methods from this category in the following section.
Apart from the aforementioned theoretical perspectives on DP in NLP, several works practically apply DP to different NLP tasks to investigate the corresponding privacy guarantees and performance. The works that are of particular relevance to this thesis are listed in Table 3.1 and will be discussed in more detail in Section 3.2.

3.2 Differential Privacy Through Embedding Vector Perturbation

As stated above, this thesis focuses on methods, that ensure DP through embedding vector perturbation, i.e., adding calibrated noise to embedding vectors. We choose this type of method because of its advantages compared to gradient perturbation based methods: they are less computationally expensive and the mechanisms can be straight-forwardly applied, independent of the dataset [MWK22]. This section will now look at the different mechanisms, which have been applied in this context in existing research.
One of the most commonly applied mechanisms is the Laplace mechanism. The works of [KGD22; Mah+22; Pan+20; PGG21] use its univariate version and add Laplace noise for each individual dimension of word-,

respectively sentence embedding vectors to achieve ϵ-DP. [FDM18; FK21; Fey+20; Qu+21; Xu+21b] make use of the Multivariate Laplace mechanism to achieve ϵ-metric DP. Due to its popularity and the fact that the latter privacy guarantee is specifically suitable for text data, we will use this mechanism during our later experiments. The definitions of the Laplace mechanism in the aforementioned works differ not only with respect to the dimensionality of the sampled noise but also with respect to how noise is calibrated and what happens before and after the application of noise. In [FK21] the amount of noise added depends not only on the privacy budget ϵ but also on the input sensitivity while in [Fey+20] noise is calibrated only to ϵ. Adding sensitivity to the calibration allows for a more targeted construction of noise. However, one first needs to bound sensitivity to be able to integrate it. Section 3.3.1 will elucidate different approaches that can be applied before adding the noise to ensure bounded sensitivity. Some works additionally map perturbed embedding vectors to other close-by embedding vectors. For example, [Fey+20] define their Laplace mechanism to contain the addition of Laplace noise as well as mapping perturbed embedding vectors to their nearest neighbor. Our notation slightly deviates from their definition as we will only refer to the noise addition without the additional vector mapping when talking about DP mechanisms. The different approaches for vector mapping will be considered separately and will be discussed in Section 3.3.2.
The Mahalanobis mechanism, which [Xu+20] introduce, enhances the Multivariate Laplace mechanism [Fey+20] by choosing an elliptical instead of a spherical noise distribution. This is achieved by calibrating the noise using Regularized Mahalanobis instead of Euclidean distance. Consequently, the mechanism satisfies ϵ-metric DP with respect to the Regularized Mahalanobis. In contrast, the Multivariate Laplace mechanism's privacy guarantee in [Fey+20] is with respect to the Euclidean distance. [Xu+20] compare their mechanism to the latter in empirical experiments and find that it yields better privacy statistics while utility stays consistent. Because of its similarity to the Multivariate Laplace mechanism, we will not include the Mahalanobis mechanism in our experiments.
In their 2019 work, [FDD19] consider the perturbation of word vector representations in Hyperbolic space as they hypothesize that it is better suited to capture hierarchical and semantic information than the Euclidean space. Therefore, they first transform words to Poincaré word embeddings, which lie in Hyperbolic space. This is where they apply DP by adding noise sampled from a Hyperbolic distribution.
Other works use versions of the exponential mechanism [MT07]. [Car+23] present a Truncated Exponential mechanism. It is defined for an arbitrary distance metric and provides ϵ-metric DP on a word-level with respect to the chosen metric. For their empirical experiments, [Car+23] elect Euclidean distance to ensure comparability to the Multivariate Laplace mechanism. Further, [MMC22] suggest a DP mechanism based on the exponential mechanism to provide ϵ-DP for documents.
[Xu+21a] present the Truncated Gumbel mechanism in their work, which yields ϵ-metric DP with respect to the Euclidean distance on a word-level. This mechanism's workings are fundamentally different from those of, for example, the Multivariate Laplace mechanism. It does not perturb embedding vectors by directly adding noise to them but instead can be described as a perturbed nearest neighbor search. Therefore, we select this mechanism as the second one to be used in our later experiments.
There are also some general limitations to embedding vector perturbation based methods on a word-level which need to be noted. As [MWK22] mention, a privatized text will always have the same length (in tokens) as the original input text. This can be critical since the length of a text could potentially also give away information about the original text. Also, the longer an input text, the higher the privacy budget required to perturb each of its words to privatize the whole text [MWK22]. Alternatively, if the same privacy budget is to be used independent of the input text's length, one needs to accept weaker privacy guarantees for longer input texts. Since each word is perturbed individually, the changes incurred from DP are predominantly semantic in nature and rarely syntactic [MWK22]. This further limits the leeway of privatization and may result in grammatical errors in the output. These limitations should be kept in mind since they will also be limitations of this thesis, which works with perturbations on a word-level.
Despite these limitations, there have already been reasonable performance results achieved [Fey+20; Xu+21a], which, together with the favorable computational aspects, warrant the focus on embedding vector perturbation based methods in this thesis.

3.3 Enhancing the Privacy-Utility Trade-off

The central set screw to influence the trade-off between privacy and utility in DP is the amount of noise added [Dwo+06]. While DP mechanisms require fixed noise distributions, their parameters can be chosen in dependence on the privacy budget ϵ. For some mechanisms, parameters are additionally based on characteristics of the embedding space, e.g., sensitivity [Dwo+06; FK21; Wu+17] or diameter [Xu+21a]. This is also one of the starting points for our investigations on enhancing the privacy-utility trade-off. We will discuss the occurrences of relevant approaches in existing research in Subsection 3.3.1. Also, one can make use of the post-processing property of DP to improve the utility of perturbed embeddings for downstream applications without affecting privacy guarantees. One method that follows this path is vector mapping approaches [Dwo+06]. Subsection 3.3.2 will outline different vector mapping approaches that this thesis will consider.

3.3.1 Approaches for Bounding Sensitivity

In existing research, different approaches for bounding sensitivity have been used in various experiment setups. Most works only consider one approach as a means to an end to achieve bounded sensitivity and, for example, earn the possibility for a more targeted calibration of noise in DP mechanisms. However, they do not compare different approaches with respect to their suitability and different approaches used across works are not comparable because of the differences in their experiment setups. This thesis applies them within a uniform experiment setup for proper comparison.

[LHL20] work with Univariate Laplace noise. They calibrate the noise to sensitivity after bounding it. Therefore, they apply a normalization step before noise addition, which restricts the range of each entry in the input vector to the interval $[0, 1]$. This is one of the approaches for bounding sensitivity, which we will transfer to the Multivariate Laplace mechanism for our experiments. [LHL20] state that due to this normalization step, the sensitivity of the input to the DP mechanism is 1. However, they overlook that this value only reflects the sensitivity per dimension and needs to be aggregated across all of the input's n dimensions. This yields a true input sensitivity of n. [Mah+22] have later pointed out this error in the sensitivity analysis.

[Mah+22] privatize document embedding vectors through a combination of a DP mechanism and adversarial training. They also bound sensitivity before applying the Univariate Laplace mechanism. Since this mechanism adds noise to each dimension of the input individually and subject to the L_1-sensitivity, they bound the length of the input with respect to L_1-norm. Therefore, they normalize the embedding vectors to unit length with respect to the L_1-norm. The effect of this normalization is equivalent to bounding the range of values for each dimension of the vector to the interval $[0, 1]$. [PGG21] use the same approach to bound input sensitivity in their work. We will follow their example for one of our approaches to bounding sensitivity. However, transferring it to the Multivariate Laplace mechanism requires normalization with respect to the L_2-norm.

[KGD22] train an encoder to transform words to latent space representations. During training, clipping is applied to restrict the representations to a hyper-sphere of fixed radius. The latent space representations are then perturbed by adding Univariate Laplacian or Gaussian noise. Due to the clipping, the representations come with bounded sensitivity. A similar clipping approach will be examined in this thesis. [Hab21] later point out an error in the privacy analysis of [KGD22] and prove that the actual sensitivity value is higher. [Hab21] further propose how the clipping approach can be modified such that the sensitivity as originally stated by [KGD22] is true. Their proposal involves normalization to unit length with respect to L_1-norm. This would, however, result in a much larger amount of noise being required and would, therefore, hurt utility [Hab21].

[LC21] examine differentially private image generation. Similar to the NLP domain, the sensitivity of images' vector representations is difficult to bound and can lead to a large amount of noise necessary to provide DP. Therefore, [LC21] clip the values of the vector representations to the maximum observed range of values based on the training data to bound the sensitivity before applying the Univariate Laplace mechanism to each of a vector's components. This allows them to provide ϵ-DP on an image-level. We

are interested if this approach can also be beneficial for NLP applications and, thus, we adopt it as another one of our approaches to bounding sensitivity.
Sensitivity cannot only be bounded by limiting the maximum range of values but also by reducing the input's dimensionality. This is the approach taken by [FK21]. They investigate randomly projecting vector representations of words to a lower-dimensional space by making use of the Johnson-Lindenstrauss (JL) lemma before adding Multivariate Laplace noise. Thereby, the sensitivity of the vectors is bounded, and calibrating the noise distribution to this sensitivity allows to alleviate the issue that the magnitude of noise necessary to guarantee DP usually grows with the input's dimensionality. The approach, which [FK21] describe, provides (ϵ, δ)-metric DP. While their theoretical considerations are applicable to vector representations in general without specifying a certain DP level, in their practical implementations they test the approach to achieve word- as well as sentence-level DP. Through utility analysis as well as experimental evaluations on NLP datasets [FK21] show that their approach outperforms the approach by [Fey+20], which does not bound sensitivity. This makes the approach interesting for our later experiments.
[Pan+20] use a workaround to finding a general bound for input sensitivity. They instead estimate sensitivity from 10,000 randomly generated pairs of embedding vectors. This sensitivity value is then used to add calibrated Univariate Laplace noise and achieve ϵ-DP on a sentence-level. Similarly, for their Truncated Gumbel mechanism, [Xu+21a], use concrete values to estimate the maximum and minimum inter-word distance of their embedding space. Truncated Gumbel noise is then calibrated based on these values. Since these measures are related to the definition of sensitivity, bounding those would also affect sensitivity. This justifies why bounding sensitivity can also help to enhance the Truncated Gumbel mechanism. The authors, however, do not implement any such bounding measures.
In the case of the exponential mechanism, instead of input sensitivity, one would bound sensitivity of the utility function, which is a central element in constructing any variant of this type of mechanism. This is usually achieved by choosing the utility function accordingly. Examples of such utility functions with bounded sensitivity can be found in [MMC22] and [Car+23].

3.3.2 Approaches for Vector Mapping

Similarly to the situation with approaches to bounding sensitivity, different approaches for vector mapping are scattered across different existing works. Most works only consider one specific approach at a time. One of those is the research by [Fey+20]. While they do not take any specific measures to bound sensitivity of the mechanism input, [Fey+20] use a vector mapping approach to end up with a real word from their dictionary as the final output of their procedure. After perturbation with the Multivariate Laplace noise, they map the perturbed embedding to its nearest neighbor embedding vector. In this work, we will use the shorthand formulation of *vector mapping* to refer to mapping embedding vectors to other close-by embedding vectors. Such vector mapping approaches do not affect the theoretical privacy guarantees because they can be seen as a post-processing step to the actual application of DP [Dwo+06]. This makes such approaches particularly interesting for improving the utility side of the privacy-utility trade-off without hurting privacy. [FDD19] employ the same vector mapping approach to map perturbed Poincaré embeddings to their nearest neighbor embedding. Similarly, [Xu+20] map their perturbed embedding vectors to their nearest neighbor with respect to the Regularized Mahalanobis distance. [Qu+21] explicitly inspect the effect of using a vector mapping approach of mapping to the nearest neighbor in combination with the Multivariate Laplace mechanism. They observe that including this vector mapping approach leads to significantly improved utility as compared to not using any vector mapping approach. The gain in utility is especially prevalent for smaller amounts of noise added.
An alternative to this vector mapping approach is introduced by [Xu+21b]. Instead of always mapping to the nearest neighbor embedding after noise addition, they choose a balanced random selection between the first and second nearest neighbor. We will adopt this approach for our experiments as an alternative to mapping to the nearest neighbor only. For a potential further enhancement of this vector mapping approach, [Xu+21b] continue to look into generalizing the vector mapping to selecting from the $k \leq 2$ nearest neighbor embeddings. They find that the biggest improvement in performance can be achieved by choosing from the two nearest neighbors instead of only choosing the nearest neighbor. This improvement

gradually levels off if we choose from more than the two nearest neighbors. Therefore, we will set $k = 2$ when applying this vector mapping approach. While the study by [Xu+21b], thus, contains a comparison of two different vector mapping approaches, they only perform this comparison in one specific experiment setup. They do not consider any DP mechanisms except for the Multivariate Laplace mechanism. Therefore, it remains unclear if their results would be reproducible in different experiment setups. This thesis will expand on this by providing a comparison of approaches in further experiment setups.
The Truncated Gumbel mechanism by [Xu+21a] does not need to be combined with an additional vector mapping approach because a noisy nearest neighbor selection is inherent to the procedure. The mechanism does not perturb embedding vectors by directly adding noise to them but instead determines a random number of nearest neighbors, adds truncated Gumbel noise to the distances between these neighbors and the original embedding vector, and chooses the neighbor with the smallest noisy distance as output. Therefore, the outputs of the mechanism are already real words. Similarly, the exponential mechanism used by [Car+23] and [MMC22] makes vector mapping redundant because choosing embedding vectors from a specified vocabulary is inherent to the mechanism. [KGD22] use the decoder part of their model's auto-encoder architecture to generate real words from perturbed word representations. This step is similar in purpose to a vector mapping approach.

Publication	Domain	DP Definition	Noise Distribution	DP Level	Approach for bounding sensitivity	Approach for vector mapping
[FK21]	NLP	(ϵ, δ)-metric DP w.r.t. Euclidean distance	Multivariate Laplace	Word-level	Dimensionality Reduction (JL lemma)	-
[LC21]	Image	ϵ-DP	Univariate Laplace	Image-Level	Clipping to observed range	-
[MMC22]	NLP	ϵ-DP	Exponential	Sentence-Level	By definition of the utility function	Selection from candidate embeddings
[PGG21]	NLP	ϵ-DP	Univariate Laplace	Word-level	Normalizing to unit length (w.r.t. L1-norm)	-
[Mah+22]	NLP	ϵ-DP	Univariate Laplace	Word-level	Normalizing to unit length (w.r.t. L1-norm)	-
[LHL20]	NLP	ϵ-DP	Univariate Laplace	Word-level	Normalizing to $[0, 1]^d$	-
[Fey+20]	NLP	ϵ-metric DP w.r.t. Euclidean distance	Multivariate Laplace	Word-level	-	Mapping to nearest neighbor
[Xu+21b]	NLP	ϵ-metric DP w.r.t. Euclidean distance	Multivariate Laplace	Word-level	-	Randomly choosing first or second nearest neighbor
[Pan+20]	NLP	ϵ-DP	Univariate Laplace	Sentence-Level	Estimation	-
[FDD19]	NLP	ϵ-metric DP w.r.t. hyperbolic distance	Hyperbolic	Sentence-Level	-	Mapping to nearest neighbor
[Qu+21]	NLP	ϵ-metric DP w.r.t. Euclidean distance	Multivariate Laplace	Token-Level	-	Mapping to nearest neighbor
[Xu+20]	NLP	ϵ-metric DP w.r.t. Regularized Mahalanobis distance	Mahalanobis	Word-Level	-	Mapping to nearest neighbor
[FDM18]	NLP	ϵ-metric DP w.r.t. Earth Mover's distance	Multivariate Laplace	Document-Level	-	Mapping to nearest neighbor
[Car+23]	NLP	ϵ-metric DP w.r.t. any metric d	Exponential	Word-Level	Defining a noisy utility function	Selection from candidate embeddings
[KGD22]	NLP	ϵ-DP	Univariate Laplace/Gaussian	Word-Level	Clipping to the interval $[-C, C]^d$	Decoder
[Xu+21a]	NLP	ϵ-metric DP w.r.t. Euclidean distance	Truncated Gumbel	Word-Level	-	Noisy nearest neighbor selection

Table 3.1 Overview of works on DP using embedding vector perturbation

4 Methodology

In this section, the setup for the empirical experiments will be described. The objective of the experiments is to examine the effect of different approaches for bounding sensitivity and vector mapping on the privacy-utility trade-off. First, details on the datasets, their preprocessing, and the tasks, that we evaluate on, are given (Section 4.1). After that, we will talk about the models used in the experiments (Section4.3) and the metrics, which will be used to evaluate the models' performance with respect to privacy and utility (Section 4.2). The complete code base corresponding to the experiments is available in a GitHub repository[1].

4.1 Tasks and Datasets

The experiments are performed on two different datasets to reduce the influence of the dataset on the results: the Trustpilot dataset [HS15] and the AG News corpus[2] [CGR05]. Since the effects on privacy and utility are to be tested, two types of experiments will be carried out. One type will be used to evaluate utility and the other to evaluate privacy. Utility is evaluated using sensitivity analysis on the Trustpilot dataset and topic classification on the AG News corpus. These tasks will also be called the *main tasks* or *utility tasks*. We will mainly examine how the utility of models on these tasks changes for different approaches and DP mechanisms with the goal of achieving as small as possible utility losses. Privacy is evaluated on simulated adversarial tasks. One task will be the classification according to a semi-private variable (gender) on the Trustpilot dataset and one will be the identification of selected named entities in texts from the AG News corpus. We strive for as large of a privacy gain as possible for the privacy experiments. All tasks are framed as classification problems. Sentiment analysis is performed as a classification of a sentiment rating on a scale of 1-5 and topic classification uses six target classes. In the context of the privacy experiments, gender identification is a binary classification problem. The identification of named entities aims at identifying the presence of each of five selected named entities in the texts. The presence of each one of these named entities is considered a separate attribute with binary outcome (*1* for present and *0* for not present) so that classification is carried out as multi-label classification where instances are classified into multiple classes simultaneously in a single classification output. These classes are non-exclusive since, in one text instance, several named entities might be present.
The two Trustpilot dataset and the AG News corpus are preprocessed as follows:

- **Trustpilot Dataset:** The Trustpilot dataset [HS15] contains text reviews. Each review is labelled with a sentiment rating on a scale of 1-5 as well as the attributes gender, and location, which have been self-provided by the authors of the reviews. On the sentiment scale, 1 represents the worst sentiment and 5 represents very good sentiment. There are five subcorpora to the Trustpilot dataset, which contain data from different regions. Since this research focuses on the English language, only the UK and the US subcorpus are used since they predominantly consist of English-language reviews. As done by [CNC18], the data is filtered for the instances with gender and birth year information. All other data instances are excluded. The attribute *gender* will be considered private information and will be used as a basis for the privacy experiments. It is transformed to a binary variable, where 1 corresponds to "male" and 0 corresponds to "female" sex. As [PGG21] mention, it is important to note that a binary gender representation may not be generally comprehensive and is only due to the dataset's structure. After that, the reviews are filtered for the English-language ones using the Python package *langid* [LB12] and only the reviews with language classified as English are kept.

[1] `https://github.com/AlishaRiecker/master-thesis.git`
[2] `http://groups.di.unipi.it/~gulli/AG_corpus_of_news_articles.html`

- **AG News Corpus:** The AG News corpus [CGR05] is a collection of news articles from different sources. The news articles are categorized by the topic they deal with. During preprocessing of the dataset, the categories that are especially infrequent, faulty, or hard to distinguish from other categories (e.g., *Top News* or *World*) are excluded. This narrows the dataset down to the categories *Business* (class 0), *Sci/Tech* (class 1), *Software and Development* (class 2), *Entertainment* (class 3), *Sports* (class 4), and *Health* (class 5). Further, data entries with missing news text are neglected. Following [CNC18], the fields *title* and *description* are concatenated to form the text which is later used as input to the classifier. As private information, named entities in each news text will be considered as it has also been done by [CNC18] and [LHL20]. Following their example, the named entities are first identified through named entity recognition using the *NLTK* package [BLK09]. This work focuses on five of the most frequently appearing named entities, which are *iraq*, *china*, *united states*, *bush*, and *british*. We proceeded with only those data instances that contain at least one of these named entities. This allows us to frame the named entity identification problem during the experiments as a multi-label classification problem, where there is one class for each named entity and the goal is to predict if a text instance contains a named entity simultaneously for all five classes.

Both datasets are scaled down to 70% of their size after preprocessing to accommodate limited computational resources during the experiments. Afterwards, the datasets are randomly split into a training set (80%), a development set (10%), and a test set (10%). Table 4.1 provides a summary of the dataset sizes.

Dataset	Train Set	Dev Set	Test Set
Trustpilot	104,055	13,006	13,007
AG News	140,510	17,563	17,564

Table 4.1 Dataset sizes (in number of data instances)

The distributions of classes among the attributes that will be classified vary depending on the respective attribute. Most of them are skewed towards a majority class. Table 4.2 provides the individual attributes' distributions.

Table 4.2 shows that for the sentiment attribute, class 4, which represents the category of the most positive sentiment, occurs much more frequently than the other classes. About 75% of all texts are very positive. Class 0 (negative sentiment) is the second most frequent class, which is still much less frequent than class 4. The distribution of the attribute gender is more balanced. The majority of texts stem from male authors (about 60%). The most frequent topic in the AG News corpus is class 3 (Entertainment). While the topics *Sports* (class 4), *Business* (class 0), and *Science/Tech* (class 1) do occur with relatively similar frequencies in the dataset, the topics *Health* (class 5) and *Software and Development* (class 2) are less frequent. Only about 0.93% of texts are on the topic *Software and Development*. For each of the five selected named entities, class 1 signals the presence of a named entity, and class 0 stands for its absence. For all of the entities, there are many more texts that do not contain the respective named entity than there are texts where it is contained. Each named entity on average only occurs in about 1.67% of texts. These class distributions should be kept in mind for the evaluation and interpretation of experiment performance. However, we choose not to apply any sampling techniques to balance the class distributions to stay closer to a real-world scenario. Experiments to investigate the influence of data distribution on privacy and utility are left to future research.

4.2 Evaluation Metrics

The basis for evaluating the performance of the different models will be the F1 score. It measures the exactness of a classification and is particularly suitable for skewed data distributions where one class is considerably larger than the others [SMR08]. This corresponds to the situation as it exists for the tasks

Dataset	Task	Attribute	Class	Ratio of Data Instances
Trustpilot	Sentiment Analysis	Sentiment label	1	9.55%
			2	2.77%
			3	2.93%
			4	9.42%
			5	75.34%
	Gender Identification	Gender	0	39.71%
			1	60.29%
AG News	Topic Classification	Topic	0	21.67%
			1	16.40%
			2	0.93%
			3	28.38%
			4	24.67%
			5	7.95%
	NE Identification	NE 1 (iraq)	0	98.10%
			1	1.90%
		NE 2 (china)	0	98.48%
			1	1.52%
		NE 3 (united states)	0	98.05%
			1	1.95%
		NE 4 (bush)	0	98.52%
			1	1.48%
		NE 5 (british)	0	98.49%
			1	1.51%

Table 4.2 Class distributions

that this work considers. The F1 score is first calculated for each individual class and then aggregated as the unweighted mean across classes to a Macro F1 score. For brevity, this work will use the term *F1 score* to refer to the Macro F1 score. Whenever the individual F1 scores on a class-level are discussed, they will be explicitly referred to as *F1 scores on a class-level*. These class-level scores will predominantly be considered in the context of the named entity identification, which is conducted as a multi-label classification problem. The value of the F1 score lies in the interval between 0 and 1, with the maximum value 1 corresponding to perfect classification [SMR08]. Since we evaluate the models with respect to empirical privacy and utility, the usage of the F1 score will be different for the utility and privacy tasks. To assess the empirical utility of a model, we will apply the model to the two main tasks, sentiment analysis and topic classification, and use the F1 scores achieved on these tasks as empirical utility metric. Higher empirical utility indicates better performance, i.e., higher utility. Empirical privacy of a model will be evaluated based on the two privacy tasks, gender and named entity identification. We follow [CNC18] and define the empirical privacy $1 - F$, where F is the F1 score, as our privacy measure. Thus, empirical privacy corresponds to the inverse F1 score on the adversarial tasks. This means that the higher the empirical privacy, the worse the simulated attacker's performance on the privacy tasks, i.e., the higher is privacy. We will use the metrics empirical privacy and utility to assess the performance of individual approaches and also compare the metrics to those achieved on the baseline models to determine the absolute change in privacy and utility with respect to the F1 score. These changes will tell us how applying a specific approach affects privacy and utility. For an easier comparison of the change in utility and the change in privacy, we model the trade-off between the two values as their sum. This is grounded on the assumption that DP leads to a decrease in utility and an increase in privacy. Thus, whenever the sum is positive we gain more than we lose. Either the gain in privacy is larger than the decrease in utility or the gain in utility is larger than the decrease in privacy. This characterizes a favorable privacy-utility trade-off where privacy and utility are weighted equally. If

the sum is negative, we either see a decrease in both, utility and privacy, or the decrease in utility is larger than our gain in privacy. These scenarios correspond to an adverse privacy-utility trade-off if we assume an equal weighting of privacy and utility. However, it needs to be noted that this is likely not the exact weighting, which one would aim for in a real-world scenario. Since the reason for applying DP is likely the goal to increase privacy, one would rather put a heavier weight on privacy. The weighting of utility depends on how much of a decrease in utility the respective application can cope with. However, this thesis focuses on a general examination of the privacy-utility trade-off. Thus, we deem the sum of the change in utility and the change in privacy a suitable heuristic for assessing the trade-off. In addition to privacy and utility, we consider the running times of the individual models to compare different methods from the perspective of the computational effort they require.

4.3 Baselines for Experiments

Since both, utility and privacy, will later be quantified by comparing performance before and after applying any approaches for limiting sensitivity, vector mapping, or adding differentially private noise, we first need to train the baseline models for each task without the application of any approaches or noise.

4.3.1 Model Architecture

Two different types of models are employed in this work to explore if the effects of different approaches also vary depending on the type of model used. We use a Long Short-Term Memory (LSTM) network and a pre-trained version of a Bidirectional Encoder Representations from Transformers (BERT) model.

- **LSTM:** The LSTM model is a type of Recurrent Neural Network (RNN) and was originally introduced by [HS97]. LSTMs contain memory units, which allow to control the information flow through the network and to remember information for longer as compared to the basic RNN [HS97]. Consequently, it can learn long-term dependencies in sequential data, which makes it especially suitable for processing text data. The first layer in the LSTM model is an embedding layer, which is constructed from 300-dimensional GloVe embeddings trained on a 6 billion token corpus [PSM14]. The embedding layer is frozen so input words are always mapped to the same vectors during the training process, which essentially corresponds to a simple look-up. Thereby, we can simulate a scenario where the input words are mapped to their corresponding embedding vector on the side of the data contributor or author of the text before the third party training a model on the data gets access [Qu+21]. It lays the foundation for the later experiments on perturbed embeddings, where the words will also be transformed to their embedding representations and perturbed on the side of the data contributor. The output of the embedding layer in the LSTM model is fed to an LSTM layer, before being propagated through a dropout layer, and on to a linear layer for the final classification.
- **BERT:** For BERT, we use the pre-trained model *BERT base (uncased)*, which has been introduced by [Dev+19]. The model follows a transformers architecture and has been pre-trained on large English-language corpora and two different unsupervised tasks [Dev+19]. It has been shown that it can achieve state-of-the-art performance on a variety of tasks by fine-tuning with one additional output layer [Dev+19]. We add a simple classifier layer on top of the pre-trained BERT model. As with the LSTM, the pre-trained model's embedding layer is frozen to mimic the implementation of the embedding look-up on the side of the data provider. However, it needs to be noted that the pre-trained model requires a text sequence, consisting of actual words, as input and constructs its own, distinct embeddings from it. BERT does not embed texts word by word but instead on a more granular word-piece level. Additionally, BERT works with positional embeddings, which carry the position of a token within an input sequence, and segment embeddings, which signal the segment or sentence to which a token belongs. Thus, we cannot use the same GloVe embeddings as for the LSTM. Also, since we decided not to amend the pre-trained model architecture, the later experiments on perturbed embeddings will use text-to-text perturbation, which will happen entirely before model input. We will provide more details on this setup in Section 4.4.

4.3.2 Baseline Training

To create the baseline models, the LSTM and the pre-trained BERT models are then trained, respectively pre-trained, on the training data set. Therefore, hyperparameter tuning is performed to achieve reasonable performance on the different tasks. However, one should note here, that the focus of this research is not to achieve a particularly strong performance of the baseline models. Thus, hyperparameter tuning was only performed for the less computationally expensive LSTM model to keep the amount of required resources within a reasonable frame. For the LSTM, different sets of hyperparameters are tried out through a grid search and evaluated on the development data set. For each task, the hyperparameters with the best performance with respect to the F1 score were chosen. The search values for the maximum length of the input in number of tokens were 100, 500, or *None*. *None* represents the option of not imposing any maximum length and instead feeding the complete input to the model. The size of the hidden layer was chosen from $\{64, 128\}$, for dropout it was $\{0, 0.1, 0.3, 0.5\}$, and for the learning rate value from the set {1e-3, 1e-4, 1e-5, 1e-6} were tried out. The batch size was set to 32 for all training runs since this value had shown to yield good results in preliminary tests. Each LSTM model was trained three times for three epochs. The number of epochs had been determined through preliminary tests, which showed that for most tasks, model performance did either not improve beyond the third epoch or the model was learning too slowly for the performance to catch up on results achieved with different hyperparameters. The choice of three training runs allows observing the standard deviation in the model's performance to assess the robustness and reproducibility of performance. A single T4 GPU is used for training the LSTM models. For the BERT model, instead of performing hyperparameter tuning, the best-performing hyperparameters were chosen based on experiences shared in existing research. The reason for this is the higher amount of computational resources required for fine-tuning BERT as compared to the LSTM. The maximum input length of the pre-trained BERT model in use is 512 [Dev+19] so this value is adopted for fine-tuning performed here. There is no need to specify values for hidden size and dropout since these values are already inherent to the pre-trained model, thus, they will not be actively changed for fine-tuning. The learning rate value is chosen to be $2e-5$, which is the same or similar to existing research fine-tuning BERT [Dev+19], [Qu+21], [Che+23], [Du+23]. Just as with the LSTM, the batch size was set to 32 following the examples for BERT fine-tuning by [Dev+19] and [Qu+21]. In order to further minimize computational resource consumption each BERT model was only trained one time, for one epoch. Preliminary tests had shown that the increase in performance with respect to the F1 score on the development set only increased marginally beyond the first epoch. We used a single A100 GPU for those training runs. An overview of the best-performing hyperparameters for each task and model can be found in Appendix A.1 as well as an overview of the corresponding models' performance over the different training runs on the development set (Table A.2).

4.3.3 Baseline Performance

The performance of the baseline models on the test dataset with respect to empirical utility and privacy as well as their average running times are stated in Table 4.3. The empirical utility values correspond to the F1 scores achieved on the main tasks of the respective datasets. The empirical privacy values are the inverse of the F1 scores reached on the adversarial tasks. The running times are averages over the different training runs and across the two tasks evaluated for each dataset and model. As mentioned above, these results will serve as a baseline to evaluate different approaches using perturbed data during the later experiments. Across all different tasks, the fine-tuned BERT model outperforms the LSTM model with respect to the F1 score. Consequently, BERT's utility is higher while its privacy is lower. The most significant difference exists for the main task on the AG News corpus where BERT achieves an empirical utility value which is about 0.18 higher than that of the corresponding LSTM. The differences measured by empirical privacy are rather small for the adversarial tasks. The best performance of both models with respect to the F1 score has been attained on the named entity identification task. Therefore, we get small values for empirical privacy on the AG News corpus. This task is performed as a multi-label classification so it makes sense to additionally inspect the performance of the individual classes. For both models, class 4 which corresponds

Dataset	Model	Empirical Utility	Empirical Privacy	Avg. Running Time (in sec.)
Trustpilot	LSTM	0.464	0.333	35.66
	BERT	0.593	0.331	2270.72
AG News	LSTM	0.608	0.041	47.25
	BERT	0.783	0.031	3054.35

Table 4.3 Results from baseline model training on unperturbed data (empirical utility and privacy are based on the test sets; running time is averaged over the different training runs and across main and adversarial tasks)

to the named entity *bush*, performs worse than the other classes. One reason for that might be the double meaning of the term. While it is a synonym of *shrub*, in the AG News corpus, the term mostly refers to the former president of the United States. This might lead to the term's slightly worse identification. Since for BERT only one training run is used to fine-tune, the average running times stated in Table 4.3 are essentially the averages across the respective main and the adversarial tasks for one training run each. For the LSTM models, the running times are additionally averaged over the three training runs performed. One can see that there is a significant difference between the time needed to train an LSTM versus fine-tuning BERT. While the running time for an LSTM is about 35-50 seconds, it is about 40-50 minutes for BERT. That means the latter takes about 65 times as long. We also observe that the average running times for the AG News corpus are slightly higher (about 1.3 times higher) than for the Trustpilot dataset. This is due to the fact that our training data set for AG News is larger (see Table 4.1).

4.4 Experiment Setup

The training setup during our experiments will be similar to the one used for training the baseline models. For each task and dataset, we will adopt the hyperparameters, which have been found to perform best during hyperparameter tuning of the baseline models. This enables a direct comparison of the models trained on perturbed data versus the corresponding baseline models. At the same time, this probably does not correspond to how it would be handled in practice. In practice, it would be more realistic for an entity to perform hyperparameter tuning on the perturbed data since it probably does not even have access to the original, unperturbed data. Thus, it would make the results of our experiments more realistic if we also performed hyperparameter tuning for the perturbed data. We decided against doing this because of the computational overhead that would arise from this. Saving this allows to perform a larger number of experiments instead.
For each of our experiments, we compose an experiment setting by varying its components along different dimensions. An overview of these different dimensions and the possible choices for each dimension is given in Table 4.4.

We use the Multivariate Laplace and the Truncated Gumbel mechanism to perturb the input data while guaranteeing DP. Then, we train an LSTM or BERT model on this perturbed data for the different NLP tasks from two datasets. This forms the basic setup for our experiments. In addition to that, we vary different components of the perturbation process to examine the effect of the different approaches to bounding sensitivity and vector mapping. Figure 4.1 provides an overview of the perturbation process for the LSTM and for BERT. The dashed elements in the process represent optional steps. We will gradually include these in our experiments in order to investigate the associated effects on privacy and utility. For the LSTM, we can in- or exclude the steps for bounding sensitivity as well as vector mapping to map to a word embedding vector corresponding to a real word. For BERT, we can in- or exclude only the step for bounding sensitivity. A vector mapping step always needs to be included since BERT expects text sequences consisting of real words as input. Thus, the output of the perturbation pipeline for BERT is always such text sequences. These are then fed to BERT's embedding layer, which embeds them using its specific BERT embeddings. The perturbation pipeline for the LSTM outputs a word embedding vector. In case a vector mapping is used,

Dataset	Model	DP Mechanism	Sensitivity Approach	Vector Mapping	ϵ
• Trustpilot • AG News	• LSTM • BERT	• Truncated Gumbel • Mulitvar. Laplace	• Normalization to unit length • Normalization to $[-1, 1]^d$ • Normalization to observed range • Clipping to observed range • Dimensionality reduction (JL lemma)	• Map to nearest neighbor • Randomly mapping to first or second nearest neighbor • No mapping	• 0.1 • 1 • 10 • 50 • 100 • 150

Table 4.4 Experiment dimensions for composing different experiment settings

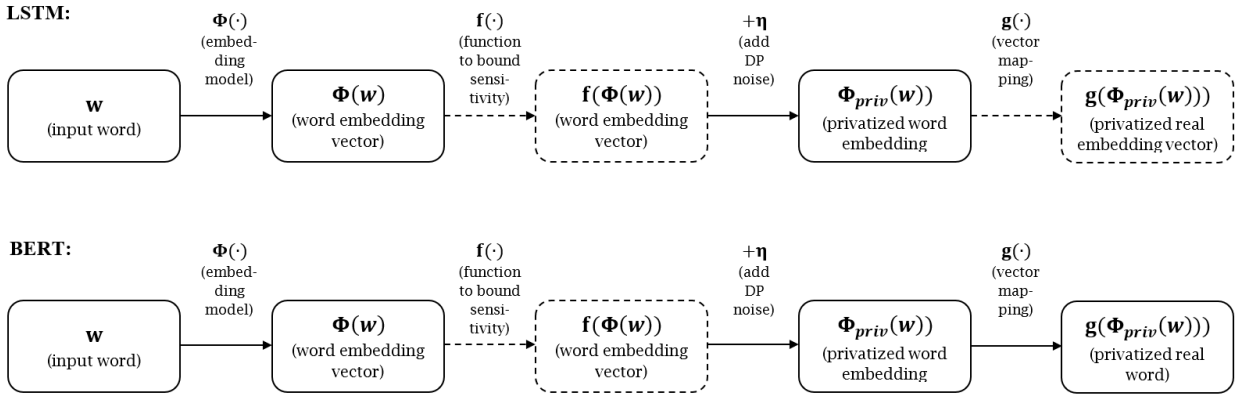

Figure 4.1 Input perturbation for experiments for LSTM and BERT

this embedding vector corresponds to a real word from a fixed vocabulary. Within the model architecture, the perturbation is built around the embedding layer such that perturbed word embeddings are directly fed to the LSTM layer of the model.

Independent of the model used, we use GloVe embeddings to transform the original input words to word embedding vectors $\Phi(x)$. The different approaches for bounding sensitivity, described by the function f are described in the next section. How to choose the noise η that we add to ensure DP, depends on the respective mechanism used. The different approaches, which will be used for the vector mapping step represented by the function g, will be detailed in Section 4.6. In the case of the Truncated Gumbel mechanism the last and the second to last step are combined into a single step. For each experiment setup, we vary the privacy budget ϵ to examine the privacy-utility trade-off for different amounts of noise added. We choose $\epsilon \in \{0.1, 1, 10, 50, 100, 150\}$. Just as for baseline training, we utilize a single T4 GPU for the experiments using an LSTM and an A100 GPU for the experiments using BERT.

4.5 Bounding Sensitivity

Bounding sensitivity (see Equation 2.3) has been identified as a promising approach for increasing the utility of differentially privatized word embedding vectors while preserving their privacy guarantees. Different researchers have identified different approaches to achieve this as outlined in Chapter 3. One of the

central questions that this work aims to answer is how these different approaches for estimating sensitivity affect privacy and utility. To explore this question, the effects will be examined and compared theoretically and through practical experiments. Which approaches have been selected for this endeavor is described in the following. It is important to note that limiting sensitivity does not per se provide DP. It only assures that the input to a DP mechanism has bounded sensitivity and thus, thereby it either affects the amount of noise required by that mechanism or the privacy guarantees, which can be achieved. In the following, we will describe the different approaches to bounding sensitivity and their implementation during the later experiments in detail and deduce their effects on sensitivity. The effects on privacy and utility will then be discussed in Chapter 5.

4.5.1 Normalizing to Unit Length

Normalizing embedding vectors to unit length before inputting them to a DP mechanism for noise addition is a common practice in applications of DP as Chapter 3 outlines. The most straightforward way to achieve this normalization is by applying a function $f_{norm}(z) = \frac{z}{\|z\|_2}$ to an embedding vector z. This function maps a vector to the unit sphere such that every output vector has length 1 with respect to the Euclidean norm. Consequently, two outputs of f_{norm} can at the maximum be located at a distance of 2 from each other if they are located opposite of each other on the unit sphere. Therefore, the distance between any two outputs of f_{norm} can be upper bounded by 2 as follows:

$$\begin{aligned} \|f_{norm}(\Phi(x)) - f_{norm}(\Phi(y))\|_2 &\leq \|f_{norm}(\Phi(x)) + f_{norm}(\Phi(y))\|_2 \\ &\leq \|f_{norm}(\Phi(x))\|_2 + \|f_{norm}(\Phi(y))\|_2 \\ &= \left\| \frac{\Phi(x)}{\|\Phi(x)\|_2} \right\|_2 + \left\| \frac{\Phi(y)}{\|\Phi(y)\|_2} \right\|_2 \\ &= 1 + 1 \\ &= 2 \end{aligned} \tag{4.1}$$

This inequality provides an upper bound to the numerator of f_{norm}'s sensitivity. However, for the complete sensitivity term, a dependence on the minimum distance between two arbitrary embedding vectors $\Phi(x)$ and $\Phi(y)$ persists as Equation 4.2 shows.

$$\begin{aligned} \Delta_{f_{norm}} &= \max_{\Phi(x),\Phi(y)} \frac{\|f_{norm}(\Phi(x)) - f_{norm}(\Phi(y)))\|_2}{\|\Phi(x) - \Phi(y)\|_2} \\ &\leq \max_{\Phi(x),\Phi(y)} \frac{2}{\|\Phi(x) - \Phi(y)\|_2} \\ &= \frac{2}{\min_{x,y} \|\Phi(x) - \Phi(y)\|_2} \end{aligned} \tag{4.2}$$

Thus, using the function f_{norm} to normalize embedding vectors to unit length, does not readily provide a bound for the whole sensitivity term but only for the maximum distance between embeddings. We are still going to examine this approach as part of our experiments and will use an estimate for the minimum distance to reach an actual bound for the complete sensitivity term. We approximate the minimum distance in the embedding space by the minimum distance observed. Inspired by [LC21] who determine the maximum distance between any two inputs to their DP mechanism based on the observed values from their training data set, this work inserts the minimum distance between embedding vectors observed from the training data to stand in for the range of possible values. While for GloVe embeddings, one could have also determined the minimum distance based on the whole GloVe vocabulary instead of only the training set, we decided against this because this would not work for other embedding models such as FastText [Boj+17], where there it is difficult to determine a fixed vocabulary because embeddings are generated based on word-pieces. Also, using the training dataset as a baseline for the estimate allows to speed up the computationally expensive calculations of pairwise distances. Finally, for our GloVe embeddings, we

determine the minimum distance to be approximately 0.7890 across both datasets, the Trustpilot and the AG News dataset. Consequently, the sensitivity can be bounded as

$$\begin{aligned}\Delta_{f_{norm}} &\lesssim \frac{2}{0.7890} \\ &\approx 2.5349.\end{aligned}$$

This work additionally considers a slightly modified version of the normalization function f_{norm}, $\tilde{f}_{norm}$, which provides a real bound on sensitivity without requiring any approximations. This function $\tilde{f}_{norm}$ only maps those vectors to the unit sphere, which have length larger than 1 with respect to the Euclidean norm. Input vectors with a length smaller or equal to one are not modified. The function $\tilde{f}_{norm}$ can be formally expressed as

$$\tilde{f}_{norm}(x) = \begin{cases} x & \|x\|_2 \leq 1 \\ \frac{x}{\|x\|_2} & \|x\|_2 > 1 \end{cases} \tag{4.3}$$

Using $\tilde{f}_{norm}$ instead of f_{norm} allows to upper bound the Euclidean distance between any two outputs of $\tilde{f}_{norm}$ by the distance between any two inputs to the function:

$$\left\| \tilde{f}_{norm}(x) - \tilde{f}_{norm}(y) \right\|_2^2 \leq \|x - y\|_2^2 \tag{4.4}$$

Equation 4.4 further allows to bound the sensitivity of $\tilde{f}_{norm}$ as $\Delta_{\tilde{f}_{norm}} \leq 1$. This follows from the fact that Equation 4.4 holds for all pairs x and y and thus also for those, for which the ratio between the left- and the right-hand side is maximal, which, in turn, corresponds to the definition of sensitivity.

4.5.2 Normalizing to the Interval $[-1, 1]^d$

Another approach to bounding sensitivity of the input to a DP mechanism that will be considered as part of this work is normalizing embedding vectors such that all its entries are bounded by the interval between -1 and 1. The idea is inspired by the normalization used by [LHL20] and [PGG21], who normalize to the interval $[0, 1]^d$. However, using an interval $[-1, 1]^d$, which is symmetric around zero, can be assumed to be more suitable when working with GloVe embeddings as they also allow both positive and negative entries. This hypothesis has been confirmed through preliminary experiments. This normalization approach, which is also called *min-max normalization* or *feature scaling*, preserves the relationship between the original input values [CAP18]. As outliers are also scaled to the interval $[-1, 1]^d$, their impact will be dampened. The normalization can be expressed by the function $g_{norm}(x) = 2 \cdot \frac{x - \min(x)}{\max(x) - \min(x)} - 1$ for an input vector $x \in \mathbf{R}^d$.
Similar to normalization to unit length, this approach allows to upper bound the distance between any two outputs of g_{norm}:

$$\begin{aligned}\|g_{norm}(x) - g_{norm}(y)\|_2 &= \sqrt{\sum_{i=1}^{d} |(g_{norm}(x))_i - (g_{norm}(y))_i|^2} \\ &\leq \sqrt{\sum_{i=1}^{d} 4} \\ &\leq 2\sqrt{d}\end{aligned} \tag{4.5}$$

Inequality 4.5, provides an upper bound to the numerator of the sensitivity term while retaining sensitivity's dependency on the minimum distance between two arbitrary embedding vectors. Thus, analogously to Inequality 4.2, sensitivity is bounded as

$$\Delta_{g_{norm}} = \frac{2\sqrt{d}}{\min_{x,y} \|\Phi(x) - \Phi(y)\|_2}. \tag{4.6}$$

Since we use 300-dimensional GloVe embeddings in this work, d is equal to 300 in this bound. The minimum distance in the denominator is again approximated based on the embeddings in the training dataset, which yields

$$\begin{aligned}\Delta_{g_{norm}} &\lesssim \frac{2 \cdot \sqrt{300}}{0.7890} \\ &\approx 43.9050.\end{aligned}$$

It should again be noted that the approximation used to reach this bound limits the generality of the bound. As opposed to the approach using normalization to unit length, there is no straightforward modified version of the normalizing function g_{norm} to circumvent this.

4.5.3 Normalizing to Observed Range

Similar to the normalization to the interval $[-1, 1]^d$ discussed before, we can also normalize embedding vectors to an interval determined based on the observed range of values to bound sensitivity. We call the function which formalizes this normalization h_{norm}. The observed range of values is defined by an interval $[v_{min}, v_{max}]$ and can be estimated by setting the interval bounds to the minimum and maximum values of embedding vectors as observed from the training data set's tokens. This is also how [LC21] determine an estimate for the range of possible values. The main advantage of applying h_{norm} to an embedding vector before inputting it to a DP mechanism is that the range of the inputs to the mechanism is known and thus, allows to bound the distance between two arbitrary embedding vectors x and y in $\mathbf{R}^d$:

$$\begin{aligned}\|h_{norm}(x) - h_{norm}(y)\|_2 &= \sqrt{\sum_{i=1}^{d} |(h_{norm}(x))_i - (h_{norm}(y))_i|^2} \\ &\leq \sqrt{\sum_{i=1}^{d} (v_{max} - v_{min})^2} \\ &= \sqrt{d \cdot (v_{max} - v_{min})^2} \\ &= \sqrt{d} \cdot (v_{max} - v_{min})\end{aligned} \tag{4.7}$$

As described in the previous sections, Equation 4.7 allows to bound the numerator of the sensitivity term while retaining the sensitivity's dependence on the minimum distance between two embedding vectors:

$$\Delta_{h_{norm}} = \frac{\sqrt{d} \cdot (v_{max} - v_{min})}{\min_{x,y} \|\Phi(x) - \Phi(y)\|_2}$$

Again, one can use the training data as a basis to determine approximations for the variables contained in this bound. We find that the observed range can be bounded by the interval $[-3.0639, 2.6668]$ for GloVe embeddings. Setting $d = 300$ to match the 300-dimensional GloVe embeddings used in this work, we can calculate the final bound on sensitivity for this approach of

$$\begin{aligned}\Delta_{h_{norm}} &\lesssim \frac{\sqrt{300} \cdot (2.6668 + 3.0639)}{0.7890} \\ &\approx 125.8031.\end{aligned}$$

4.5.4 Clipping to Observed Range

Instead of normalizing to the observed range, we can alternatively clip the values in the embedding vectors to this range. Depending on the two bounding values v_{min} and v_{max}, which determine the observed range,

entries of embedding vectors are clipped to the closer bounding value whenever they fall outside of the observed range. This procedure can be formalized by the following clipping function f_{clip}

$$f_{clip}(x_i) = \begin{cases} x_i & x_i \in [v_{min}, v_{max}] \\ v_{min} & x_i < v_{min} \\ v_{max} & x_i > v_{max} \end{cases}$$

As with normalization to the observed range, the values v_{min} and v_{max} are approximated based on the GloVe embedding vectors contained in the training data. Applying f_{clip} allows to upper bound the sensitivity $\Delta_{f_{clip}}$ by 1. This follows from the fact that we can bound the distance between two outputs of f_{clip} by the distance between the two corresponding inputs x and y:

$$\begin{aligned} \|f_{clip}(x) - f_{clip}(y)\|_2^2 &= \|f_{clip}(x)\|_2^2 - 2\|f_{clip}(x)\|_2\|f_{clip}(y)\|_2 + \|f_{clip}(y)\|_2^2 \\ &\leq \|x\|_2^2 - 2\|x\|_2\|y\|_2 + \|y\|_2^2 \\ &= \|x - y\|_2^2 \end{aligned} \tag{4.8}$$

Inequality 4.8 follows from the fact that the absolute value of each entry $|(f_{clip}(x))_i|$ is smaller than or equal to the corresponding entry of the input vector x_i.

4.5.5 Dimensionality Reduction Using JL Lemma

One of the issues with adding noise according to a DP mechanism is that the amount of noise also grows with the dimensionality of the input vector. For the Multivariate Laplace mechanism, the dimensionality d determines the shape parameter of the Gamma distribution, which the noise magnitude is sampled from. Thus, the larger d is, the flatter the distribution's density function becomes and the higher the probability to draw larger noise magnitudes. In the case of the Truncated Gumbel mechanism, the dimensionality d influences the maximum and minimum inter-word distances and through those parameters also the amount of noise added. To address this issue of the noise being larger for larger input dimensionality, [FK21] perform dimensionality reduction through random projection before feeding the projected vector to the DP mechanism. Generally speaking, through dimensionality reduction, a high-dimensional data set can be embedded into a lower-dimensional space. The method that [FK21] use is based on the Johnson-Lindenstrauss Lemma, which grants that the distances between vectors are approximately preserved during dimensionality reduction with a function f_Φ. This characteristic of the dimensionality reduction also directly provides a probabilistic bound on sensitivity [FK21]. Before formalizing this bound on the dimensionality reduction function f_Φ, one first needs to define a set's Gaussian width [FK21]:

Definition 4.5.1 (Gaussian Width)**.** *Given a closed set $\mathcal{X} \subset \mathbb{R}^d$, its Gaussian width $\omega(\mathcal{X})$ is defined as:*

$$\omega(\mathcal{X}) = \mathbb{E}_{g \in \mathcal{N}(0,1)^d}\left[\sup_{x \in \mathcal{X}} \langle x, g \rangle\right] \tag{4.9}$$

Using this definition, one can now formalize the probabilistic bound on sensitivity provided by applying the JL lemma for dimensionality reduction:

Lemma 4.5.1 (JL Lemma)**.** *Let Φ be an $d \times m$ matrix with i.i.d. entries from $\mathcal{N}(0, 1/m)$. Let $\beta \in (0, 1)$. If $m = \Omega\left(\frac{\left(\omega(Range(M)) + \sqrt{log(1/\delta)}\right)^2}{\beta^2}\right)$, where $Range(M) \subset \mathbb{R}^d$ denotes the range of the embedding model M and $\omega(Range(M))$ its Gaussian width. Then,*

$$\mathbb{P}\left[\Delta_{f_\Phi} \leq 1 + \beta\right] \geq 1 - \delta \tag{4.10}$$

Since this bound is probabilistic in nature, it is possible that there exist two word embedding vectors x and x' for which this bound fails. This is also relevant when analyzing the privacy guarantees for the

combination of dimensionality reduction with a DP mechanism.
To use Lemma 4.5.1 for the practical experiments, it is crucial to determine the target dimensionality m, which the embedding vectors can be mapped to. Therefore, it is necessary to bound the Gaussian width of the embedding model's range $\omega(Range(M))$. An additional requirement to this bound is that the resulting value for the target dimension m needs to be smaller than the original embeddings' dimension d. This source dimension d is equal to 300 in the case of this work. To fulfill this requirement, the following bound on the Gaussian width for a finite subset of the Euclidean ball is used:

Lemma 4.5.2. *Let $\mathcal{X}$ be a finite subset of the Euclidean ball of unit radius in $\mathbb{R}^d$. Then,*

$$\omega(\mathcal{X}) \leq \sqrt{2 \cdot \ln |\mathcal{X}|}. \tag{4.11}$$

Proof. Let $z = \mathbb{E}\left[\sup_{x \in \mathcal{X}} \langle x, g \rangle\right]$ and $\lambda > 0$.
Then,

$$\begin{aligned}
\exp(\lambda \cdot z) &\leq \mathbb{E}\left[\exp\left(\lambda \cdot \sup_{x \in \mathcal{X}} \langle x, g \rangle\right)\right] && (4.12)\\
&= \mathbb{E}\left[\sup_{x \in \mathcal{X}} \left(\exp\left(\lambda \cdot \langle x, g \rangle\right)\right)\right] && (4.13)\\
&\leq \sum_{i=1}^{|\mathcal{X}|} \mathbb{E}\left[\exp\left(\lambda \cdot \langle x, g \rangle\right)\right] && (4.14)\\
&= \sum_{i=1}^{|\mathcal{X}|} \exp \frac{\lambda^2}{2} && (4.15)\\
&= |\mathcal{X}| \cdot \exp \frac{\lambda^2}{2} && (4.16)
\end{aligned}$$

The first inequality follows from Jensen inequality and the penultimate step follows by definition of the Gaussian moment-generating function.
Therefore, it holds that

$$\begin{aligned}
&\exp(\lambda \cdot z) \leq |\mathcal{X}| \cdot \exp \frac{\lambda^2}{2} && (4.17)\\
\Leftrightarrow \quad &\lambda \cdot x \leq \ln\left(|\mathcal{X}| \cdot \exp\left(\frac{\lambda^2}{2}\right)\right) && (4.18)\\
\Leftrightarrow \quad &x \leq \frac{\ln |\mathcal{X}|}{\lambda} + \frac{\lambda}{2} && (4.19)
\end{aligned}$$

Minimizing the right-hand side with respect to λ leads to choosing $\lambda = \sqrt{2 \cdot \ln |\mathcal{X}|}$, which concludes the proof. □

In our practical experiments, we first need to map the embeddings to the Euclidean ball of unit radius to profit from the guarantees of Lemma 4.5.2. This can be achieved by normalizing the embedding vectors to unit length. Even though this will incur an additional distortion to the embedding space, it enables us to project to a reasonable target dimension. To transfer the theoretical guarantees to our practical experiments, we use $\mathcal{X}$ to denote the set of unique word embedding vectors normalized to unit length. It is important to note that the bound in Lemma 4.5.2 is independent of the source dimension d and only depends on the cardinality of $\mathcal{X}$, i.e., on the size of the dataset. Since the dependence is logarithmical, the relative reduction that can be achieved is especially large for datasets with larger cardinality of $\mathcal{X}$. In the experiments, $|\mathcal{X}|$ is determined from the cardinality of the whole dataset in terms of the number of unique tokens. The reason for using the whole dataset as a point of reference is to get the most conservative bound possible. For the Trustpilot dataset, this yields $|\mathcal{X}| = 72,600$, and for the AG News dataset, it is $|\mathcal{X}| = 75,378$. The Gaussian widths of the two datasets can thus both be bounded by 4.74 making use of

Lemma 4.5.2.
Following Lemma 4.5.1, the target dimension m for the experiments needs to be chosen at least as large as $\frac{\left(\omega(|\mathcal{X}|)+\sqrt{\ln(1/\delta)}\right)^2}{\beta^2}$. The factor δ controls the probability with which the bound on the sensitivity holds. While it is desirable for this probability to be high, a large δ counteracts this. At the same time, a larger δ provides the possibility to choose a smaller value for the target dimension m, and therefore, achieve a greater reduction in dimensionality. The experiments in the upcoming section make use of a value for δ of 1e-6 which is also used by [FK21] in all of their experiments. Since the parameter β affects the privacy guarantee, it is preferred to choose β as small as possible to increase indistinguishability. On the other hand, a small β will limit the choices for the target dimension m to larger values, which is less welcome. Through these relationships, β also has a non-negligible influence on the privacy-utility trade-off. [FK21] examine this influence by varying β in their experiments. From their results, it is to be assumed that larger values of β are beneficial in practice. Based on this deduction [FK21] choose $\beta = 0.9$ for their experiments on NLP datasets. Following this example, the same value is chosen for the experiments. All in all, these choices for the parameters allow to choose the target dimension as $m \geq 89$ and bound sensitivity as

$$\mathbb{P}\left[\Delta_{f_\Phi} \leq 1.9\right] \geq 0.999999$$

4.6 Vector Mapping

As discussed in Chapter 3, there are several works that use vector mapping approaches as a post-processing step to their DP mechanism in order to map perturbed words back to real words from a vocabulary or the corresponding embedding vectors. The main benefit of employing such a vector mapping is that it makes the output of a DP mechanism more interpretable by relating the output with real words from a known vocabulary. This additionally improves the usability of the perturbed embeddings because they can then also be used as input to models that have been trained on non-perturbed words. Without the vector mapping a possibility to directly input the perturbed embedding vectors to the model would need to be given. Using vector mapping approaches to post-process the outputs of a DP mechanism is especially attractive since it does not interfere with the theoretical privacy guarantees provided by the mechanism. This is ensured by the post-processing property of DP (Proposition 2.1.1). In the upcoming sections, we will provide details on two different vector mapping approaches, which will then be examined in our experiments.

4.6.1 Mapping to Nearest Neighbor

The most commonly used vector mapping approach maps a perturbed embedding vector to its nearest neighbor embedding vector. As detailed in Chapter 3, this approach was first introduced by [Fey+20] who use it as a post-processing step to the Multivariate Laplace mechanism. The nearest neighbor search is usually carried out on a predefined set of vectors which is equivalent to the embedding vectors associated with a fixed vocabulary. For this vocabulary, one could use the set of all word embeddings vectors, which appear in the training dataset. In this work, we use a limited vocabulary consisting of 34,573 tokens which is based on the vocabulary of the BERT model and different sets of popular words. This leads to a smaller vocabulary than if we would construct it based on the training dataset and helps to speed up the nearest neighbor search. Nearest neighbor search is generally a computationally expensive task to perform. To find the nearest neighbor of only one specific vector the pairwise distances to all other vectors in the vocabulary are required. However, there are sophisticated approximation methods to speed up the computation. We also make use of such methods for this thesis and choose the Python package *faiss* [Joh+19] to perform the nearest neighbor search. The package provides efficient algorithms for similarity search on vectors with GPU support. It should be noted that the methods internally use compressed representations, which might make results less precise. However, our preliminary results showed that computations can be greatly sped up using this package so that we accept the potentially reduced precision.

4.6.2 Random Choice Between First and Second Nearest Neighbor

Instead of deterministically mapping to the nearest neighbor embedding vector, we can use the approach introduced by [Xu+21b] and randomly choose to map either to the first or the second nearest neighbor. This approach follows from the motivation to make the reconstruction of an input word harder as it has been observed that when only mapping to the first nearest neighbor, perturbed embedding vectors might be mapped back to the original unperturbed embedding [Xu+21b]. This is especially the case for rare words being located in sparse regions within the embedding space, which might still be closest to the original word even after perturbation. Including the second nearest neighbor in the mapping procedure allows to reduce the risk of reconstructing the original word, and therefore, secures privacy [Xu+21b]. The approach is implemented such that as a first step, both the first and the second nearest neighbors of a perturbed embedding vector are determined. Then, one of them is randomly selected. The probability for this selection depends on the perturbed embedding vectors distances to the first and second nearest neighbor as well as on a tuning parameter $t \in [0, 1]$ [Xu+21b]. For $t \to 1$, the second nearest neighbor is favored in the selection process while for $t \to 0$ the first nearest neighbor is favored. In the experiments performed by [Xu+21b], they find that for $t \leq 0.75$, a higher value for t results in better empirical privacy for the same privacy budget ϵ. However, choosing $t = 1$ does not enhance privacy any further. Therefore, we choose to set $t = 0.75$ in our experiments. [Xu+21b] also tested if it makes sense to generalize this approach to selecting from the $k \leq 2$ nearest neighbor embeddings and found that performance can be improved the most if we choose $k = 2$ instead of $k = 1$. This is why we also adopt the approach of selecting from the $k = 2$ nearest neighbors for our experiments. For the implementation of this approach, we again use the nearest neighbor search from the *faiss* package [Joh+19] and map to vectors from our predefined, limited vocabulary.

5 Sensitivity Analysis

This chapter will discuss our experiments' results for different approaches to bound sensitivity of the input to a DP mechanism. Based on the respective bound deduced in the previous chapter, we will first examine the effects of these bounds on privacy and utility for each approach. Specifically, we will consider the effects of the bounds on the amount of noise required by a mechanism and the privacy guarantees achieved. Then, we will analyze the empirical performance on the two datasets.

5.1 Preliminary Experiments with Unbounded Sensitivity

To determine the effect of each approach on the privacy-utility trade-off, we will now look at the results of preliminary experiments, which consider the application of DP mechanisms only, without any approach to bound sensitivity or vector mapping. The detailed results can be found in Table 5.1. We consider only the LSTM model for the Multivariate Laplace mechanism since BERT would require an additional vector mapping step to achieve text-to-text perturbation before model input. For the Truncated Gumbel mechanism, we consider both, LSTM and BERT. Since the vector mapping is inherent to the Truncated Gumbel mechanism, it allows the output of real words which can then be fed to BERT without any issues. Note that for smaller ϵ values on the AG News corpus, the model's performance is very low with respect to the F1 score and resembles that of random guessing. It seems that in these cases the amount of noise added is too big such that the model fails to provide reasonable results. As this is generally not desirable, we exclude the corresponding results from our analysis as well as from Table 5.1. We highlight favorable privacy-utility trade-offs where the privacy gain outweighs the utility loss in Table 5.1 in bold. The preliminary results show that the Multivariate Laplace mechanism leads to smaller utility losses for increasing ϵ values while the privacy gain simultaneously becomes smaller. This is due to less noise being added by the mechanism as ϵ increases. While it helps the performance on the main task, it also helps a potential adversary, leading to smaller privacy enhancements. Across the different ϵ values and the two datasets we mostly get negative values for our privacy-utility trade-off heuristic. This signals an unfavorable trade-off, where we either lose more utility than we gain in privacy or we lose more privacy than we gain in utility. Only for $\epsilon = 10$ on the AG News corpus, we get a positive result for the trade-off since we see a larger gain in privacy which compensates for the utility loss. The addition of Laplace noise increases the running time for training a single model by about three minutes compared to the baseline scenario without any noise added. This corresponds to an increase in running time of about 5%. The running time increases by the same factor when using the Truncated Gumbel mechanism instead. This shows that the calculation and addition of noise take the same amount of time for both mechanisms. For the Truncated Gumbel mechanism, empirical utility and privacy values are more consistent across different ϵ values. We achieve relatively high absolute utility and privacy values even for small privacy budgets. Using an LSTM, we see small utility decreases across all privacy budgets ϵ. With respect to the change in privacy, the picture is different for the two datasets. While the Truncated Gumbel mechanism decreases privacy on the Trustpilot dataset, it enhances it for the AG News corpus. Consequently, the privacy-utility trade-off shows negative values on the former and mostly positive values on the latter. We also see different results for the privacy-utility trade-off for BERT depending on the dataset. We notice an unfavorable trade-off between privacy and utility on the Trustpilot dataset while results are favorable on the AG News corpus. On the AG News corpus, using BERT after perturbation from the Truncated Gumbel mechanism even leads to gains in both, privacy and utility, for some ϵ values. This corresponds to a favorable privacy-utility trade-off independent of how we weigh privacy and utility. The perturbation of words before they are input into the model adds about 10 minutes to the total training time for a BERT model.

DP Mechanism	Model	Dataset	ϵ	Change in Utility	Change in Privacy	Trade-off
Multivariate Laplace	LSTM	Trustpilot	0.1	-0.287	+0.259	-0.029
			1	-0.287	+0.262	-0.025
			10	-0.272	+0.261	-0.011
			50	-0.090	+0.041	-0.049
			100	-0.095	-0.004	-0.098
			150	-0.029	+0.007	-0.022
		AG News	0.1	-*	-*	-*
			1	-*	-*	-*
			10	-0.416	+0.536	**+0.120**
			50	-0.054	+0.009	-0.045
			100	-0.022	+0.014	-0.008
			150	-0.014	+0.005	-0.008
Truncated Gumbel	LSTM	Trustpilot	0.1	-0.014	-0.001	-0.015
			1	-0.013	-0.002	-0.015
			10	-0.014	-0.006	-0.020
			50	-0.017	-0.004	-0.021
			100	-0.019	+0.000	-0.019
			150	-0.018	-0.003	-0.021
		AG News	0.1	-0.034	+0.017	-0.017
			1	-0.014	+0.017	**+0.004**
			10	-0.016	+0.014	-0.002
			50	-0.013	+0.014	**+0.001**
			100	-0.015	+0.016	**+0.001**
			150	-0.015	+0.016	**+0.002**
Truncated Gumbel	BERT	Trustpilot	0.1	-0.014	-0.017	-0.031
			1	-0.014	-0.028	-0.042
			10	-0.005	-0.028	-0.033
			50	-0.006	-0.028	-0.034
			100	-0.006	-0.036	-0.042
			150	-0.004	-0.026	-0.030
		AG News	0.1	-0.007	+0.024	**+0.018**
			1	-0.001	+0.008	**+0.007**
			10	+0.000	+0.008	**+0.008**
			50	-0.023	+0.008	-0.016
			100	+0.017	+0.013	**+0.029**
			150	+0.003	+0.001	**+0.003**

* We have left out results where the model behaves like a random guesser.

Table 5.1 Performance of preliminary experiments using only DP mechanisms (without bounding sensitivity and vector mapping) with respect to privacy and utility. Favorable trade-offs are indicated in bold.

Sensitivity Approach	Sensitivity*	Mechanism	Noise Parameters	Privacy Guarantee
None	$\Delta = \infty$ (unbounded)	Multivariate Laplace	$d = 300$ $\theta = 1/\epsilon$	ϵ-metric DP
		Truncated Gumbel	$\Delta_{max} = 18.4198$ $\Delta_{min} = 0.7890$	ϵ-metric DP
Normalizing to Unit Length	$\Delta \leq 2.5349$	Multivariate Laplace	$d = 300$ $\theta = 1/\epsilon$	$\epsilon \cdot 2.5349$-metric DP
		Truncated Gumbel	$\Delta_{max} = 1.8962$ $\Delta_{min} = 0.1298$	ϵ-metric DP
Normalizing to Unit Length (adapted)	$\Delta \leq 1$	Multivariate Laplace	$d = 300$ $\theta = 1/\epsilon$	ϵ-metric DP
		Truncated Gumbel	$\Delta_{max} = 1.8962$ $\Delta_{min} = 0.1298$	ϵ-metric DP
Normalizing to the Interval $[-1, 1]^d$	$\Delta \leq 43.9050$	Multivariate Laplace	$d = 300$ $\theta = 1/\epsilon$	$\epsilon \cdot 43.9050$-metric DP
		Truncated Gumbel	$\Delta_{max} = 18.8451$ $\Delta_{min} = 0.4773$	ϵ-metric DP
Normalizing to Observed Range	$\Delta \leq 125.8031$	Multivariate Laplace	$d = 300$ $\theta = 1/\epsilon$	$\epsilon \cdot 125.8031$-metric DP
		Truncated Gumbel	$\Delta_{max} = 53.9977$ $\Delta_{min} = 1.3676$	ϵ-metric DP
Clipping to Observed Range	$\Delta \leq 1$	Multivariate Laplace	$d = 300$ $\theta = 1/\epsilon$	ϵ-metric DP
		Truncated Gumbel	$\Delta_{max} = 18.4198$ $\Delta_{min} = 0.7890$	ϵ-metric DP
Dimensionality Reduction using JL lemma	$\Delta \leq 1.9$ with prob. 0.999999	Multivariate Laplace	$d = 89$ $\theta = 1/\epsilon$	$(1.9\epsilon, (1\text{e-}6)\delta)$-metric DP
		Truncated Gumbel	$\Delta_{max} = 2.2047$ $\Delta_{min} = 0.0800$	(ϵ, δ)-metric DP

* The sensitivity bounds for the different approaches have been derived in Chapter 4. For clarity, we leave out the subscripts referring to the respective functions in this table (e.g., for normalization to unit length, the sensitivity corresponds to $\Delta_{f_{norm}}$).

Table 5.2 Theoretical Privacy Guarantees for Sensitivity Approaches

5.2 Bounding Sensitivity

We will now analyze the different approaches for bounding input sensitivity and their effects on privacy and utility from a theoretical perspective as well as based on the results achieved in the practical experiments. From the theoretical perspective, we will mainly consider the amount of noise added as part of the mechanisms and the theoretical privacy guarantees. Table 5.2 provides an overview of the sensitivity bounds, noise parameters, and theoretical privacy guarantees for the different approaches to bounding sensitivity. The practical implications will be deducted from the approaches' performance on the test dataset in downstream NLP tasks as described in Chapter 4.

5.2.1 Normalizing to Unit Length

As pointed out in Subsection 4.5.1, normalizing embedding vectors to unit length does not readily provide a bound for the whole sensitivity term and so we use an estimate for the minimum distance of embeddings in order to achieve a bound to sensitivity of 2.5349. For the Multivariate Laplace mechanism, this approach does not influence the amount of noise added since the parameters in the multivariate Gaussian and the Gamma distribution are not affected. However, its theoretical privacy guarantee for this approach is scaled by the bound on sensitivity. We can show this by considering a mechanism $\mathcal{A}_1 = \mathcal{A} \circ f_{norm}$, which concatenates the Multivariate Laplace mechanism $\mathcal{A}$ with f_{norm}:

$$\begin{aligned} \frac{\mathbb{P}\left[\mathcal{A}_1(x) = y\right]}{\mathbb{P}\left[\mathcal{A}_1(x') = y\right]} &\leq \exp\left(\epsilon \cdot \|f_{norm}(\Phi(x)) - f_{norm}(\Phi(x'))\|_2\right) \\ &\leq \exp\left(\epsilon \cdot \Delta_{f_{norm}} \|\Phi(x) - \Phi(x')\|_2\right) \\ &= \exp\left(\epsilon \cdot 2.5349 \cdot \|\Phi(x) - \Phi(x')\|_2\right) \end{aligned} \tag{5.1}$$

Thus, the mechanism $\mathcal{A}_1$ satisfies $\epsilon \cdot 2.5349$-metric DP. When interpreting this theoretical privacy guarantee, one should mind the approximation used. Since it has been determined based on the training data set, there is a chance that it does not hold for some embedding vectors in the development or test dataset. Thus, the true theoretical privacy guarantee could be weaker. For the truncated Gumbel mechanism the amount of noise added to perturb the distances is affected through the maximum and minimum inter-word distances. As described in the previous chapter, normalization to unit length makes it possible to limit the maximum distance between words by 2. The minimum inter-word distance can also be upper bounded by 2. However, to calibrate the noise in the Truncated Gumbel mechanism more precisely, we use estimates based on the training datasets for both values instead of these bounds. For the maximum inter-word distance, we have $\Delta_{max} = 1.8962$; for the minimum inter-word distance, we get $\Delta_{min} = 0.1298$ from the above approximation. It is important to note that these values have been determined after normalizing all embedding vectors to unit length, and thus, are attributes of a transformed embedding space. Thus, the amount of noise is influenced by the normalization for the Truncated Gumbel mechanism as opposed to the Multivariate Laplace mechanism. However, this change in noise allows to achieve ϵ-metric DP as theoretical privacy guarantee.

Before looking at the results achieved in experiments using perturbed word embedding vectors, we use the approach without additional noise from a DP mechanism. Therefore, we normalize all word embedding vectors before using them to train the models on the different NLP tasks. These tests will later assist with delineating the effect of normalization to unit length from the effect of DP perturbation. It can be assumed that the distortion of the embedding space incurred by the normalization will be reflected in the performance of a downstream model without any noise already. Note, that the tests will only be performed using the LSTM model since we require real word inputs for BERT, which cannot be achieved without additionally adding a vector mapping approach. A full overview of the results for all approaches using unperturbed data can be found in Table 5.4. We notice that normalization to unit length leads to a slight decrease in utility but also to a slight increase in privacy on both datasets compared to the baseline models. It further improves the privacy-utility trade-off since the gain in privacy is larger than our loss in utility. Note that this improvement purely stems from distortions in the embedding space caused by the approach.

Sensitivity Approach	DP Mechanism	Model	Dataset	ϵ	Change in Utility	Change in Privacy	Trade-off
Normalization to unit length	Multivariate Laplace	LSTM	Trustpilot	0.1	-0.288	+0.265	-0.024
				1	-0.286	+0.273	-0.013
				10	-0.283	+0.268	-0.015
				50	-0.278	+0.270	-0.008
				100	-0.238	+0.199	-0.040
				150	-0.093	+0.155	**+0.062**
			AG News	0.1	-*	-*	-*
				1	-*	-*	-*
				10	-*	-*	-*
				50	-*	-*	-*
				100	-0.276	+0.410	**+0.134**
				150	-0.272	+0.088	-0.184
Normalization to unit length	Truncated Gumbel	LSTM	Trustpilot	0.1	-0.072	+0.033	-0.039
				1	-0.070	+0.028	-0.042
				10	-0.092	+0.073	-0.019
				50	-0.084	+0.073	-0.011
				100	-0.081	+0.075	-0.006
				150	-0.093	+0.072	-0.021
			AG News	0.1	-0.200	+0.388	**+0.188**
				1	-0.224	+0.388	**+0.164**
				10	-0.169	+0.331	**+0.162**
				50	-0.169	+0.329	**+0.161**
				100	-0.137	+0.332	**+0.195**
				150	-0.133	+0.327	**+0.194**
Normalization to unit length	Truncated Gumbel	BERT	Trustpilot	0.1	-0.014	-0.031	-0.045
				1	-0.013	-0.030	-0.043
				10	-0.010	-0.037	-0.048
				50	-0.012	-0.029	-0.040
				100	-0.011	-0.027	-0.038
				150	+0.005	-0.040	-0.035
			AG News	0.1	-0.043	+0.011	-0.032
				1	-0.005	+0.015	**+0.010**
				10	-0.076	+0.011	-0.065
				50	-0.014	+0.003	-0.011
				100	-0.024	+0.006	-0.019
				150	-0.025	+0.009	-0.017

* We have left out results where the model behaves like a random guesser.

Table 5.3 Performance of experiments on normalization to unit length (without vector mapping) with respect to privacy and utility. Favorable trade-offs are indicated in bold.

Dataset	Sensitivity Approach	Change in Utility	Change in Privacy	Trade-off
Trustpilot	Normalization to unit length	-0.102	+0.173	**+0.071**
	Normalization to unit length (adapted)	-0.124	+0.103	-0.021
	Clipping to observed range	-0.032	+0.037	**+0.004**
	Normalizing to observed range	-0.026	+0.080	**+0.053**
	Normalizing to the interval $[-1, 1]^d$	-0.021	+0.111	**+0.091**
	Dimensionality reduction (JL lemma)	-0.115	+0.218	**+0.104**
AG News	Normalization to unit length	-0.010	+0.020	**+0.010**
	Normalization to unit length (adapted)	-0.002	+0.023	**+0.021**
	Clipping to observed range	+0.010	+0.009	**+0.018**
	Normalizing to observed range	-0.027	+0.018	-0.009
	Normalizing to the interval $[-1, 1]^d$	-0.028	+0.027	-0.002
	Dimensionality reduction (JL lemma)	-0.188	+0.084	-0.105

Table 5.4 Performance of experiments using only approaches to bound sensitivity (without any DP mechanism or vector mapping) with respect to privacy and utility. Favorable trade-offs are indicated in bold.

While these distortions seem to be favorable for our tasks, the approach on its own does not provide any guarantees for enhancing privacy. The normalization of the word embedding vectors leads to a minimal increase in running time of about three seconds compared to the baseline model. This is similar to what we observe for other approaches to bounding sensitivity, which will also be discussed in the remainder of this chapter.

Next, we consider normalization to unit length in combination with the Multivariate Laplace mechanism for different values of ϵ (see Table 5.3). For this combination, we use the LSTM models because the lack of a vector mapping approach is incompatible with using BERT. On the AG News corpus, we are again omitting results where the models fail to learn and just mimic a random guesser. Across all ϵ values, the combination of normalization to unit length with the Multivariate Laplace mechanism leads to decreases in utility and increases in privacy as compared to the baselines. As ϵ increases, the utility loss as well as the privacy gain becomes smaller. This can be explained by the amount of noise added through the mechanism becoming lower as ϵ increases. As less noise is added, the absolute values for empirical utility and privacy become closer to those of the baselines. Overall, the gain in privacy cannot offset the utility loss in most cases, which leads to a negative privacy-utility trade-off. Only for larger ϵ values, can we reach a favorable trade-off. Therefore, using this combination of approaches only makes sense if one has a large privacy budget at one's disposal. This is rather unlikely in practice.

As we can see from Table 5.3, the results look different with the Truncated Gumbel mechanism and depend on which model is used. Using an LSTM model, we again notice a decrease in utility and an increase in privacy across different ϵ values. However, for this mechanism, these changes are more consistent across privacy budgets ϵ. Larger privacy budgets, i.e., smaller amounts of added noise, do not necessarily lead to a smaller decrease in utility or a smaller increase in privacy. With respect to the privacy-utility trade-off, we further need to distinguish between the two datasets. For the Trustpilot dataset, the increases in privacy are counterbalanced by larger decreases in utility, resulting in negative values for the trade-off between the two values. For the AG News, the privacy gains are more pronounced so the privacy-utility trade-off is positive. This shows that the dataset and corresponding tasks also have a noticeable influence on the privacy-utility trade-off. Examining this influence is not the subject of this work but might be of interest to future work. The values for the privacy-utility trade-off are similar for different values of ϵ. Hence, an advantage of the Truncated Gumbel mechanism is that it does not depend as much on the privacy budget and one can achieve a reasonable privacy-utility trade-off also for smaller ϵ values compared to the Multivariate Laplace mechanism. Considering this in light of the previously discussed theoretic privacy guarantees, it is reasonable to assume that the more targeted calibration of noise in the Truncated Gumbel mechanism can partially compensate for the changes in the ϵ value to provide more consistent perfor-

mance. If we use BERT instead of an LSTM, the privacy-utility trade-off is negative for the majority of experiments. On the Trustpilot dataset, privacy is even decreased when applying the approach. This can be explained by the BERT model yielding higher F1 scores on the privacy task than the baseline, which corresponds to a potential adversary achieving better results and, therefore, harms privacy. Thus, even though BERT leads to better model performance, this is not necessarily beneficial for the privacy-utility trade-off if it also helps a potential adversary.

Overall, normalization to unit length can further enhance the privacy-utility trade-off for some combinations of DP mechanism and model type. This can be recognized by comparing the just discussed results to those of the preliminary results from Section 5.1. Additionally, results can differ depending on the dataset and corresponding main and adversarial tasks. In our experiments, normalization to unit length achieves the best result with a combination of the Truncated Gumbel mechanism and an LSTM model on the AG News dataset. Since normalization to unit length itself does not add significantly to the running time of the models, the running time of its combination with a DP mechanism is mainly determined by the latter. Therefore, we end up with a running time which is increased by about three minutes as compared to the baseline model for the LSTMs. For BERT, running time is again conditioned on the pre-computation of the perturbation, which takes about 10 minutes for the training and development dataset per model.

Using $\tilde{f}_{norm}$ as a modified version of f_{norm} allows to bound sensitivity with $\Delta_{\tilde{f}_{norm}} \leq 1$ as detailed in the previous chapter. In combination with the Laplace mechanism, this approach can be used to guarantee ϵ-metric DP while the function does not affect the amount of noise added. The theoretical privacy bound can be shown analogously to the calculations in Inequality 5.1. Since the bound on sensitivity does not rely on any approximations in this case, the privacy guarantee holds for any input vectors. When we combine $\tilde{f}_{norm}$ and the Truncated Gumbel mechanism, the amount of noise required again depends on the maximum and minimum inter-word distances, similar to the non-adapted normalization function f_{norm}. Also, as with the non-adapted normalization function, maximum and minimum distances are estimated based on the training datasets to calibrate noise more pointedly. This yields the same values $\Delta_{max} = 1.8962$ and $\Delta_{max} = 0.1298$ as before. Using these values in the calibration of noise for the Truncated Gumbel mechanism provides ϵ-metric DP.

As with the non-adapted normalization, we now look at the effects of $\tilde{f}_{norm}$ without any additional noise in the results of our preliminary experiments (see Table 5.4). Since $\tilde{f}_{norm}$ does not transform all areas of the embedding space in the same way, we expect a more severe distortion of the embedding space from this adapted normalization approach. The expectation is fulfilled for the Trustpilot dataset. The adapted version of normalization to unit length leads to a larger decrease in utility and a smaller privacy gain than the non-adapted version. This results in a negative privacy-utility trade-off. The opposite is true for the AG News corpus. We get a positive trade-off value, which is even larger than for the non-adapted version of the normalization approach. The situation for this dataset fits what would be expected based on the theoretical privacy guarantees. The guarantee for the non-adapted version is larger as it is scaled by this approach's sensitivity so one would expect this approach to provide less privacy than the adapted version. However, our empirical results cannot confirm this theory-based assumption across datasets. From the results of the preliminary experiments, we can also not confirm the assumption that the adapted version of normalization to unit length introduces a stronger distortion of the embedding space.

We continue to investigate the validity of this assumption for the combination of the adapted normalization to unit length with the two DP mechanisms. The detailed results for these experiments can be found in Table 5.5. For the Multivariate Laplace mechanism, we do again see utility losses and privacy gains, which become smaller as ϵ values increase. Overall, the utility losses are larger than the gains in privacy so we end up with a negative privacy-utility trade-off for the majority of those experiments. Many of the values are also smaller than for the non-adapted version of the normalization to unit length.

For the combination of the approach with the Truncated Gumbel mechanism, we distinguish between the different types of models used. Using an LSTM, the results are again different for the two datasets. While the privacy-utility trade-off is negative for all privacy budgets ϵ on the Trustpilot dataset, it is positive for the AG News corpus. Also, we observe that similarly to the preliminary experiments, the adapted version of the normalization improves the privacy-utility trade-off on the AG News corpus compared to the

Sensitivity Approach	DP Mechanism	Model	Dataset	ϵ	Change in Utility	Change in Privacy	Trade-off
Normalization to unit length (adapted)	Multivariate Laplace	LSTM	Trustpilot	0.1	-0.289	+0.258	-0.032
				1	-0.287	+0.255	-0.032
				10	-0.286	+0.253	-0.033
				50	-0.275	+0.255	-0.020
				100	-0.217	+0.200	-0.017
				150	-0.141	+0.134	-0.007
			AG News	0.1	-*	-*	-*
				1	-*	-*	-*
				10	-*	-*	-*
				50	-*	-*	-*
				100	-0.272	+0.321	**+0.049**
				150	-0.229	+0.106	-0.123
Normalization to unit length (adapted)	Truncated Gumbel	LSTM	Trustpilot	0.1	-0.198	+0.052	-0.146
				1	-0.201	+0.052	-0.149
				10	-0.100	+0.056	-0.043
				50	-0.096	+0.053	-0.043
				100	-0.097	+0.033	-0.064
				150	-0.103	+0.027	-0.076
			AG News	0.1	-0.136	+0.342	**+0.207**
				1	-0.133	+0.380	**+0.247**
				10	-0.135	+0.353	**+0.219**
				50	-0.131	+0.396	**+0.265**
				100	-0.130	+0.329	**+0.198**
				150	-0.137	+0.319	**+0.182**
Normalization to unit length (adapted)	Truncated Gumbel	BERT	Trustpilot	0.1	-0.031	-0.031	-0.061
				1	-0.008	-0.001	-0.008
				10	-0.012	-0.043	-0.054
				50	+0.003	-0.008	-0.006
				100	+0.001	-0.032	-0.031
				150	+0.001	-0.037	-0.036
			AG News	0.1	-0.033	+0.002	-0.031
				1	-0.010	+0.001	-0.009
				10	-0.011	+0.008	-0.003
				50	-0.012	+0.010	-0.002
				100	-0.009	+0.004	-0.004
				150	-0.022	+0.002	-0.020

* We have left out results where the model behaves like a random guesser.

Table 5.5 Performance of experiments on the adapted version of normalization to unit length (without vector mapping) with respect to privacy and utility. Favorable trade-offs are indicated in bold.

non-adapted version. It seems like this approach better suits this dataset and the corresponding tasks' requirements. BERT again leads to negative privacy-utility trade-offs independent of the ϵ value. The model does not only improve performance on the target task but also yields performance gains for the simulated adversary. Independent of the model used, the values of the privacy-utility trade-off do not differ a lot for different privacy budgets ϵ of the Truncated Gumbel mechanism. This further supports our hypothesis that the results for the Truncated Gumbel mechanism are less dependent on the privacy budget and that even a small privacy budget can suffice to achieve a comparably good privacy-utility trade-off.
To answer the question if it is generally beneficial to use the adapted version of normalization to unit length, we compare against the preliminary results described in Section 5.1. One can say that it again depends on the combination of DP mechanism and model type if the privacy-utility trade-off is improved. Our experiments show that it is the combination of the Truncated Gumbel mechanism and an LSTM model on the AG News dataset that achieves the biggest improvement on the trade-off. Since this is also the combination with the most favorable privacy-utility trade-off when no approach to bounding sensitivity is applied, one could further put forward the alternative hypothesis that the adapted version of the normalization reinforces previously existing results.

5.2.2 Normalizing to the Interval $[-1, 1]^d$

By making use of an estimate for the minimum distance between arbitrary embedding vectors, we were able to upper bound sensitivity by 43.9050 in Subsection 4.5.2. Since no parameters of the Multivariate Laplace mechanism are affected by the normalization to the interval $[-1, 1]^d$, it also does not influence the amount of noise added by this mechanism. However, the theoretical privacy guarantee that this combination provides is again scaled by the bound on sensitivity. A mechanism $\mathcal{A}_2 = \mathcal{A} \circ g_{norm}$, which combines the Multivariate Laplace mechanism $\mathcal{A}$ with normalization to the interval $[-1, 1]^d$ using g_{norm} provides $\epsilon \cdot 43.9050$-metric DP:

$$\begin{aligned} \frac{\mathbb{P}\left[\mathcal{A}_2(x) = y\right]}{\mathbb{P}\left[\mathcal{A}_2(x') = y\right]} &\leq \exp\left(\epsilon \cdot \left\|g_{norm}(\Phi(x)) - g_{norm}(\Phi(x'))\right\|_2\right) \\ &\leq \exp\left(\epsilon \cdot \Delta_{g_{norm}} \left\|\Phi(x) - \Phi(x')\right\|_2\right) \\ &= \exp\left(\epsilon \cdot 43.9050 \cdot \left\|\Phi(x) - \Phi(x')\right\|_2\right) \end{aligned} \tag{5.2}$$

When interpreting this theoretical privacy guarantee, we need to again keep in mind that it is based on an approximation and consequently, might not hold on other datasets. The guarantee further suggests that normalization to the interval $[-1, 1]^d$ provides less privacy for the same values of ϵ than normalization to unit length because of the weaker privacy guarantee, which comes from the higher value for sensitivity. The later experiments will reassess if this assumption also holds in practical applications. In the case of the Truncated Gumbel mechanism, the theoretical privacy guarantee is not affected by normalization to the interval $[-1, 1]^d$ so that the mechanism provides ϵ-metric DP. Instead, the approach's effects are already considered during the noise calibration within the mechanism as it changes the maximum and minimum inter-word distances, which are parameters of the noise calibration. While the inter-word distances can generally be bounded by $2\sqrt{d}$, or by 34.64 for our 300-dimensional GloVe embeddings, we again use estimates based on the training data set for more accurate values and a more targeted noise calibration. After normalization to the interval $[-1, 1]^d$, the maximum inter-word distance of the embedding space can be estimated to be about $\Delta_{max} = 18.8451$ and the minimum inter-word distance to be about $\Delta_{min} = 0.4773$.
We use these values for first experiments without any additional noise from a DP mechanism to check how big the effect of only the normalization to the interval $[-1, 1]^d$ is, i.e., how much of an influence the distortion of the embedding space has that comes with it. As Table 5.4 shows, utility decreases to about the same extent on both datasets. However, this decrease can only be balanced out by a larger privacy gain for the Trustpilot dataset. Thus, we observe a positive privacy-utility trade-off. For the AG News corpus, the privacy-utility trade-off is hurt by normalization to the interval $[-1, 1]^d$.
We now continue with the experiments that combine the normalization to the interval $[-1, 1]^d$ with the DP mechanisms to examine how this approach for bounding sensitivity performs in this setting. An overview

Sensitivity Approach	DP Mechanism	Model	Dataset	ϵ	Change in Utility	Change in Privacy	Trade-off
Normalization to the interval $[-1, 1]^d$	Multivariate Laplace	LSTM	Trustpilot	0.1	-0.288	+0.274	-0.014
				1	-0.291	+0.268	-0.022
				10	-0.279	+0.258	-0.020
				50	-0.058	+0.069	**+0.011**
				100	-0.025	+0.048	**+0.024**
				150	-0.019	+0.052	**+0.033**
			AG News	0.1	-*	-*	-*
				1	-*	-*	-*
				10	-*	-*	-*
				50	-0.207	+0.050	-0.156
				100	-0.096	+0.024	-0.072
				150	-0.096	+0.026	-0.070
Normalization to the interval $[-1, 1]^d$	Truncated Gumbel	LSTM	Trustpilot	0.1	-0.034	+0.075	**+0.041**
				1	-0.036	+0.080	**+0.044**
				10	-0.038	+0.075	**+0.037**
				50	-0.039	+0.073	**+0.034**
				100	-0.031	+0.064	**+0.033**
				150	-0.036	+0.070	**+0.034**
			AG News	0.1	-0.195	+0.157	-0.038
				1	-0.176	+0.154	-0.022
				10	-0.193	+0.146	-0.047
				50	-0.193	+0.152	-0.041
				100	-0.194	+0.150	-0.043
				150	-0.193	+0.154	-0.039
Normalization to the interval $[-1, 1]^d$	Truncated Gumbel	BERT	Trustpilot	0.1	-0.037	-0.025	-0.062
				1	-0.036	-0.007	-0.043
				10	-0.004	+0.001	-0.003
				50	-0.001	-0.032	-0.034
				100	-0.002	-0.030	-0.032
				150	-0.002	-0.034	-0.036
			AG News	0.1	-0.016	+0.007	-0.009
				1	-0.038	+0.004	-0.034
				10	-0.013	+0.012	-0.001
				50	-0.004	+0.006	**+0.002**
				100	-0.028	+0.003	-0.025
				150	-0.006	+0.008	**+0.003**

* We have left out results where the model behaves like a random guesser.

Table 5.6 Performance of experiments on normalization to the interval $[-1, 1]^d$ (without vector mapping) with respect to privacy and utility. Favorable trade-offs are indicated in bold.

of the corresponding results is given in Table 5.6. The experiments' results show that the combination with the Multivariate Laplace mechanism yields utility losses as well as simultaneous privacy gains for all ϵ values. The privacy-utility trade-off is however only positive for $\epsilon \in \{50, 100, 150\}$ on the Trustpilot dataset. In all other experiments, this mechanism does not achieve favorable trade-offs.
For the combination of normalization to the interval $[-1, 1]^d$ with the Truncated Gumbel mechanism, it is striking that the results for the privacy-utility trade-off again seem to be connected with the dataset respectively the corresponding tasks. When using an LSTM model, we achieve positive privacy-utility trade-offs for all values of ϵ on the Trustpilot dataset while on the AG News corpus, all values are negative. Since the results on the two datasets are converse this time with Trustpilot yielding better results than AG News, we hypothesize that the dataset and tasks are one of the main factors when deciding on a specific combination of approaches and DP mechanisms. We see that we can get very different results for the privacy-utility trade-off for the same dataset depending on which methods and approaches we combine. When using BERT, the privacy-utility trade-off is negative for most of the experiments. Only for ϵ equal to 50 and 150 on the AG News corpus, the trade-off yields a positive value. However, the values are still close to zero and there are no further patterns visible in the experiment results for this model so we cannot make any assumptions about how these effects might be related to the model or the experiment setup.
Finally, to derive if normalization to the interval $[-1, 1]^d$ generally improves the privacy-utility trade-off, we come back to our preliminary experiments where no measures for bounding sensitivity were taken (see Section 5.1). We notice that the best choice for a mechanism and additional approaches strongly depends on the dataset at hand as well as the tasks that are chosen to evaluate performance on this dataset. For our experiments, it would be advisable to include normalization to the interval $[-1, 1]^d$ when working with the Trustpilot dataset and a combination of the Truncated Gumbel mechanism and an LSTM because this is the setting where it leads to an improved privacy-utility trade-off, independent of the privacy budget ϵ. If the available privacy budget is large enough it might also make sense to consider the Multivariate Laplace mechanism and an LSTM. However, this mechanism only reaches an improved trade-off if small amounts of noise are added, i.e., large ϵ's are used.

5.2.3 Normalizing to Observed Range

The sensitivity using normalization to the observed range can be upper bounded by 125.8031 as detailed in the previous chapter. From this bound, we can deduce that combining this approach with the Multivariate Laplace mechanism provides $\epsilon \cdot 125.8031$-metric DP. To formally derive the guarantee, we consider a mechanism $\mathcal{A}_3 = \mathcal{A} \circ h_{norm}$ combining the Multivariate Laplace mechanism $\mathcal{A}$ with normalization to the observed range represented by h_{norm}:

$$\begin{aligned}\frac{\mathbb{P}\left[\mathcal{A}_3(x) = y\right]}{\mathbb{P}\left[\mathcal{A}_3(x') = y\right]} &\leq \exp\left(\epsilon \cdot \|h_{norm}(\Phi(x)) - h_{norm}(\Phi(x'))\|_2\right) \\ &\leq \exp\left(\epsilon \cdot \Delta_{h_{norm}} \|\Phi(x) - \Phi(x')\|_2\right) \\ &= \exp\left(\epsilon \cdot 125.8031 \cdot \|\Phi(x) - \Phi(x')\|_2\right)\end{aligned}$$

While the theoretical privacy guarantee is scaled by the bound on sensitivity, the amount of noise added as part of the mechanism does not incur any changes from the normalization approach. For the Truncated Gumbel mechanism, it is again the other way around. The privacy guarantee is not influenced by normalization to the observed range and can be stated as ϵ-metric DP. The noise added by the mechanism changes as the parameters maximum and minimum inter-word distance need to be adapted after the normalization. Based on estimates from our training dataset after normalizing all its embedding vectors to the maximum observed range of $[-3.0639, 2.6668]$, we set the maximum inter-word distance Δ_{max} to 53.9977 and determine the minimum inter-word distance to be $\Delta_{min} = 1.3676$. We can see that the distances between word embedding vectors are larger than before the normalization, which leads to spreading the vectors more widely within the observed range. This can also be influenced by outlier values.
To begin our discussion of the results achieved from practical experiments, we again first look at the effects of applying only normalization to the observed range without combining the approach with any DP

Sensitivity Approach	DP Mechanism	Model	Dataset	ϵ	Change in Utility	Change in Privacy	Trade-off
Normalization to observed range	Multivariate Laplace	LSTM	Trustpilot	0.1	-0.289	+0.261	-0.028
				1	-0.289	+0.262	-0.028
				10	-0.144	+0.124	-0.020
				50	-0.025	+0.039	**+0.013**
				100	-0.021	+0.037	**+0.016**
				150	-0.031	+0.041	**+0.010**
			AG News	0.1	-*	-*	-*
				1	-*	-*	-*
				10	-0.338	+0.159	-0.178
				50	-0.066	+0.019	-0.048
				100	-0.061	+0.012	-0.049
				150	-0.061	+0.012	-0.048
Normalization to observed range	Truncated Gumbel	LSTM	Trustpilot	0.1	-0.018	+0.026	**+0.008**
				1	-0.029	+0.022	-0.006
				10	+0.002	+0.025	**+0.027**
				50	-0.008	+0.024	**+0.016**
				100	-0.000	+0.027	**+0.027**
				150	-0.002	+0.025	**+0.023**
			AG News	0.1	-0.114	+0.026	-0.088
				1	-0.111	+0.023	-0.088
				10	-0.112	+0.025	-0.086
				50	-0.112	+0.026	-0.086
				100	-0.111	+0.027	-0.084
				150	-0.110	+0.023	-0.086
Normalization to observed range	Truncated Gumbel	BERT	Trustpilot	0.1	-0.016	-0.011	-0.028
				1	-0.025	-0.029	-0.054
				10	-0.024	-0.014	-0.038
				50	-0.024	-0.041	-0.065
				100	-0.026	-0.030	-0.055
				150	-0.026	-0.022	-0.048
			AG News	0.1	-0.026	+0.011	-0.014
				1	-0.019	+0.011	-0.009
				10	-0.012	+0.009	-0.003
				50	-0.017	+0.005	-0.012
				100	-0.022	+0.010	-0.012
				150	-0.010	+0.010	-0.000

* We have left out results where the model behaves like a random guesser.

Table 5.7 Performance of experiments on normalization to observed range (without vector mapping) with respect to privacy and utility. Favorable trade-offs are indicated in bold.

mechanism (see Table 5.4). We find that the privacy-utility trade-off looks different for the two datasets. While we get a positive value for the Trustpilot dataset, the trade-off is negative for the AG News corpus. This might be an indication that the performance of this approach also depends on the dataset on which it is evaluated.
We examine if this pattern persists when the normalization to the observed range is combined with the Multivariate Laplace mechanism. Indeed, Table 5.7 shows positive values for the privacy-utility trade-off for the largest three ϵ values on the Trustpilot dataset because here the privacy gain is larger than the utility loss. For all other values of ϵ, both on the Trustpilot dataset and the AG News corpus, the privacy-utility trade-off is negative because the privacy gain is too small to balance out the utility loss.
If we combine normalization to the observed range with the Truncated Gumbel mechanism, we again need to look at the results for the two different models separately. For the LSTM, we observe the same trend as for the Multivariate Laplace mechanism, where the experiments on the Trustpilot dataset perform better than those on the AG News corpus. On the Trustpilot dataset, the trade-off is positive for five out of the six ϵ values. These positive trade-offs are all due to the privacy gain being larger than the utility loss. It is also notable that these utility losses are only very minor with values close to zero. On the AG News corpus, utility losses are larger and, thus, outweigh the privacy gains leading to negative values for the privacy-utility trade-off. When using BERT with the Truncated Gumbel mechanism, we do not observe these different effects for the two datasets. Here, all values for the privacy-utility trade-off are negative independent of the dataset and the ϵ value. For the Trustpilot dataset, we even get privacy losses in addition to the utility losses. This has already been observed previously and is due to the model outperforming the baseline on the adversarial task (gender identification), which leads to lower privacy.
The decision if it is reasonable to include normalization to the observed range when working with DP mechanisms depends on the dataset used. We compare the results using the normalization to those of the preliminary experiments where no measures for bounding sensitivity were taken (see Section 5.1) and notice that normalization improves the privacy-utility trade-off only on the Trustpilot dataset while it can even be harmful in the case of the AG News dataset. On the Trustpilot dataset, we see better trade-offs for several of our experiments, especially using an LSTM. This effect exists independent of which of the mechanisms is used. On the AG News corpus, trade-offs that are positive without the normalization turn out negative if it is included. Compared to other, previously discussed approaches to bounding sensitivity, normalization to the observed range does not seem to perform worse even though the theoretical privacy guarantee, which we have calculated for this approach, is much weaker.

5.2.4 Clipping to Observed Range

With the approach of clipping embedding vectors to fall into the observed range of values, sensitivity can be bounded as $\Delta_{f_{clip}} \leq 1$ as shown in the previous chapter. As a consequence of this bound, the application of f_{clip} to the input of the Multivariate Laplace mechanism yields ϵ-metric DP. We can formally describe this privacy guarantee by considering a mechanism $\mathcal{A}_4 = \mathcal{A} \circ f_{clip}$, which combines the Multivariate Laplace mechanism $\mathcal{A}$ with the clipping function f_{clip}:

$$\begin{aligned}\frac{\mathbb{P}\left[\mathcal{A}_4(x) = y\right]}{\mathbb{P}\left[\mathcal{A}_4(x') = y\right]} &\leq \exp\left(\epsilon \cdot \left\|f_{clip}(\Phi(x)) - f_{clip}(\Phi(x'))\right\|_2\right) \\ &\leq \exp\left(\epsilon \cdot \Delta_{f_{clip}} \left\|\Phi(x) - \Phi(x')\right\|_2\right) \\ &= \exp\left(\epsilon \cdot \left\|\Phi(x) - \Phi(x')\right\|_2\right)\end{aligned}$$

Thus, for this approach to bounding sensitivity, the theoretical privacy guarantee for the Multivariate Laplace mechanism is not weakened by a perceptible scaling with the sensitivity bound. Also, none of the mechanism's noise parameters are affected by the clipping so the noise magnitude does not change due to this approach. When combining the clipping to the observed range with the Truncated Gumbel mechanism, we get the same theoretical privacy guarantee of ϵ-metric DP as for the combination with the Multivariate Laplace mechanism. However, the noise added via this mechanism is affected by changes in its parameters, the maximum and minimum inter-word distances. Because we determined our observed

Sensitivity Approach	DP Mechanism	Model	Dataset	ϵ	Change in Utility	Change in Privacy	Trade-off
Clipping to observed range	Multivariate Laplace	LSTM	Trustpilot	0.1	-0.291	+0.259	-0.032
				1	-0.288	+0.265	-0.023
				10	-0.271	+0.253	-0.018
				50	-0.062	+0.034	-0.029
				100	-0.014	+0.024	**+0.010**
				150	-0.015	-0.007	-0.022
			AG News	0.1	-*	-*	-*
				1	-*	-*	-*
				10	-0.412	+0.512	**+0.100**
				50	-0.057	+0.013	-0.045
				100	-0.033	+0.005	-0.029
				150	-0.034	+0.004	-0.030
Clipping to observed range	Truncated Gumbel	LSTM	Trustpilot	0.1	-0.052	+0.019	-0.032
				1	-0.027	-0.002	-0.029
				10	-0.033	-0.003	-0.036
				50	-0.050	-0.003	-0.053
				100	-0.042	+0.010	-0.032
				150	-0.048	+0.014	-0.034
			AG News	0.1	-0.030	+0.014	-0.016
				1	-0.030	+0.013	-0.017
				10	-0.029	+0.011	-0.018
				50	-0.030	+0.011	-0.018
				100	-0.003	+0.015	**+0.012**
				150	+0.000	+0.015	**+0.015**
Clipping to observed range	Truncated Gumbel	BERT	Trustpilot	0.1	+0.013	-0.022	-0.009
				1	+0.013	-0.028	-0.014
				10	+0.015	-0.014	+0.002
				50	+0.015	-0.031	-0.015
				100	+0.015	+0.059	**+0.074**
				150	+0.015	-0.036	-0.021
			AG News	0.1	-0.007	+0.002	-0.005
				1	-0.045	+0.009	-0.036
				10	-0.015	+0.001	-0.014
				50	+0.000	+0.004	**+0.004**
				100	-0.012	+0.010	-0.002
				150	-0.026	+0.002	-0.024

* We have left out results where the model behaves like a random guesser.

Table 5.8 Performance of experiments on clipping to observed range (without vector mapping) with respect to privacy and utility. Favorable trade-offs are indicated in bold.

range based on the values occurring in the training datasets, there are no vectors in the training dataset, which would fall outside of this range and would be clipped. Thus, estimating the maximum and minimum inter-word distance from the training data after applying the clipping yields the same values as for the original embeddings. We get a maximum inter-word distance of $\Delta_{max} = 18.4198$ and a minimum inter-word distance of $\Delta_{min} = 0.7890$.
We now examine the effects of only the clipping approach on our downstream NLP tasks to see how the potential distortion of the embedding space incurred from the clipping affects their performance. From the results in Table 5.4, we see that the clipping seems to have a beneficial effect on the privacy-utility trade-off since it yields a positive value for both datasets. In the case of the AG News corpus, it even leads to an increase in utility. This means that it improves the model's performance on the main task with respect to the F1 score. Based on these results, we expect the approach to be especially beneficial for the utility side of the trade-off and to lead to smaller utility losses.
Keeping this in mind, we now continue to look at the approach's performance if we use it together with the Multivariate Laplace mechanism. The results in Table 5.8 indicate a loss in utility as well as an increase in privacy for the majority of ϵ values. However, in most of these cases, the utility loss is larger than the privacy gain resulting in a negative privacy-utility trade-off. The only cases of positive values for the trade-off are for $\epsilon = 100$ on the Trustpilot dataset and for $\epsilon = 10$ on the AG News corpus. Thus, there does not seem to be a dependence on the dataset.
When using the Truncated Gumbel mechanism and an LSTM, most of the experiments also yield negative values for the privacy-utility trade-off. For some of the smaller ϵ values in the Trustpilot dataset, we even observe negative values for the change in privacy, i.e., privacy losses. This signals that performance on the gender identification task is better than for the baseline model, which corresponds to helping the simulated adversary. Putting this into the context of our preliminary experiments, this approach seems to generally have a beneficial influence on models' performance with respect to the F1 score. Thereby, it does, however, not only aid utility but also a potential adversary, which can lead to a decrease in privacy. For the combination of the Truncated Gumbel mechanism and the LSTM, there are only two cases of a positive privacy-utility trade-off. Those are achieved for ϵ equal to 100 and 150 on the AG News corpus. If we combine this mechanism with BERT, we also observe a positive trade-off for some of the ϵ's. A big part of those can be explained by increased utility for the respective experiments. In some cases, this occurrence is further supported by a decrease in privacy. This confirms our hypothesis that clipping to the observed range yields a general improvement in models' performance which does not only help utility but also a potential adversary. It can, hence, be harmful to the privacy-utility trade-off.
The comparison with results of the preliminary experiments without bounded sensitivity (see Section 5.1) further supports this observation. Unlike previously considered approaches, with clipping to the observed range, there is no combination of DP mechanism and model which clearly outperforms others with respect to the privacy-utility trade-off. The results rather seem to be dependent on which privacy budget ϵ is used. Also, unlike for previously considered approaches, we do not notice any relation with the dataset being used as there is no combination of mechanism and model which yields better results on only one of the two datasets. This can, however, also be seen as an advantage such that the approach might provide a favorable privacy-utility trade-off independent of the dataset if an appropriate DP mechanism and model are picked.

5.2.5 Dimensionality Reduction Using JL Lemma

As stated in the previous chapter, dimensionality reduction using the JL lemma, as done by [FK21], provides a probabilistic bound on sensitivity such that the probability that $\Delta_{f_\Phi} \leq 1.9$ is at least 0.999999. Consequently, when used before input to the Laplace mechanism, this dimensionality reduction guarantees $(1.9\epsilon, 1e-6)$-metric DP. The δ in this privacy guarantee reflects the probabilistic nature of the sensitivity's bound. Due to this, it is theoretically possible that there are two embedding vectors for which the bound on sensitivity, and as a result, this privacy guarantee does not hold. This needs to be kept in mind when working with this theoretical guarantee. For the Multivariate Laplace mechanism, we also need to note that the amount of noise the mechanism adds is affected by the dimensionality reduction. The reduced

Sensitivity Approach	DP Mechanism	Model	Dataset	ϵ	Change in Utility	Change in Privacy	Trade-off
Dimensionality reduction (JL lemma)	Multivariate Laplace	LSTM	Trustpilot	0.1	-0.287	+0.264	-0.022
				1	-0.283	+0.263	-0.021
				10	-0.282	+0.265	-0.017
				50	-0.260	+0.249	-0.011
				100	-0.193	+0.239	**+0.046**
				150	-0.225	+0.070	-0.155
			AG News	0.1	-*	-*	-*
				1	-*	-*	-*
				10	-*	-*	-*
				50	-0.336	+0.691	**+0.355**
				100	-0.344	+0.333	-0.011
				150	-0.332	+0.287	-0.045
Dimensionality reduction (JL lemma)	Truncated Gumbel	LSTM	Trustpilot	0.1	-0.158	+0.232	**+0.074**
				1	-0.157	+0.119	-0.039
				10	-0.160	+0.110	-0.050
				50	-0.159	+0.107	-0.052
				100	-0.154	+0.106	-0.048
				150	-0.158	+0.108	-0.050
			AG News	0.1	-0.352	+0.599	**+0.248**
				1	-0.332	+0.585	**+0.253**
				10	-0.320	+0.608	**+0.288**
				50	-0.290	+0.584	**+0.294**
				100	-0.293	+0.506	**+0.213**
				150	-0.290	+0.492	**+0.202**
Dimensionality reduction (JL lemma)	Truncated Gumbel	BERT	Trustpilot	0.1	-0.039	-0.016	-0.056
				1	-0.018	-0.031	-0.049
				10	-0.019	-0.028	-0.047
				50	-0.021	+0.017	-0.004
				100	-0.003	-0.035	-0.038
				150	-0.004	-0.023	-0.027
			AG News	0.1	-0.010	+0.009	-0.001
				1	-0.012	+0.010	-0.002
				10	-0.031	+0.005	-0.025
				50	-0.010	+0.008	-0.002
				100	-0.020	+0.001	-0.019
				150	-0.020	+0.021	**+0.002**

* We have left out results where the model behaves like a random guesser.

Table 5.9 Performance of experiments on dimensionality reduction using the JL lemma (without vector mapping) with respect to privacy and utility. Favorable trade-offs are indicated in bold.

dimensionality of the input vectors $m = 89$ will lead to a reduced noise magnitude. When combining the dimensionality reduction approach with the Truncated Gumbel mechanism, the amount of noise is also affected by the dimensionality reduction. After dimensionality reduction, we can estimate the maximum inter-word distance as $\Delta_{max} = 2.2047$ based on the training datasets. The minimum inter-word distance can be estimated as $\Delta_{min} = 0.0800$. As we can see, these values have been greatly reduced through dimensionality reduction. Using noise that has been calibrated using these inter-word distances, the combination of dimensionality reduction and the Truncated Gumbel mechanism provides (ϵ, δ)-metric DP.

As with the approaches considered before, we begin our discussion of experiment results by looking at the performance of models using only dimensionality reduction without any added noise from a DP mechanism (see Table 5.4). On both datasets, the approach leads to a decrease in utility and an increase in privacy. However, for the AG News corpus, the privacy gain is smaller than the utility loss so that it cannot outbalance it resulting in a negative privacy-utility trade-off. The trade-off for the Trustpilot dataset, on the other hand, is positive as we gain more privacy than what we lose with respect to utility. We note that compared to other approaches, this one shows stronger effects already in those preliminary experiments. The decrease in utility is larger than for other approaches. This also applies to the privacy gain. When looking at the absolute values for empirical privacy and utility, we notice great reductions in the performance of the downstream NLP tasks. This gives rise to the assumption that the dimensionality reduction leads to strong distortions within the embedding space, which generally hurts the performance of downstream models. At the same time, the dimensionality reduction speeds up model training so that including this approach does not lead to an increased running time for model training. The time needed to map embedding vectors to a lower dimensional embedding space is balanced out by the reduced model training time.

We next combine the dimensionality reduction with the Multivariate Laplace mechanism. This yields the expected decreases in utility and increases in privacy across all ϵ values as can be seen from Table 5.9. We note that these changes in utility and privacy are strikingly large. When added together the decrease in utility mostly outweighs the privacy gains so the privacy-utility trade-off is negative. We only achieve positive trade-offs for $\epsilon = 100$ on the Trustpilot dataset and for $\epsilon = 50$ on the AG News corpus.

The alternative combination of dimensionality reduction with the Truncated Gumbel mechanism performs more consistently across different values for the privacy budget ϵ. When using an LSTM, the outcome seems to depend on the dataset. While the values for the privacy-utility trade-off are negative for the majority of experiments on the Trustpilot dataset, all experiments on the AG News corpus yield positive trade-offs. Therefore, the dimensionality approach seems to work well in this exact setting. If BERT is used instead of the LSTM, the decreases in utility are smaller. The same applies to the privacy gains on the AG News corpus. On the Trustpilot dataset, we see privacy losses for most of the experiments. This pattern exhibits how BERT can deal better with the distortions in the embedding space incurred by the dimensionality reduction. The model's performance with respect to the F1 score is overall very high and similar to the baseline score. Still, since this also benefits the simulated adversary, the privacy-utility trade-off mostly turns out negative.

The comparison to the experiments without the dimensionality reduction (Section 5.1) reveals that incorporating the approach only helps in certain settings. In our experiments, including the approach only significantly improves the privacy-utility trade-off for the Truncated Gumbel mechanism and an LSTM while other settings hurt the trade-off. Thus, the dataset, DP mechanism, and model greatly influence the privacy-utility trade-off, which we can reach when incorporating the approach. However, we also need to keep in mind that it likely hurts the performance of the models with respect to the F1 score due to distortions in the embedding space. Even though the JL lemma ensures limited distortion of the distances between pairs of embedding vectors, these still seem too large not to affect performance in downstream NLP tasks. It can be assumed that some of these distortions also originate from projecting embedding vectors to the unit ball before the dimensionality reduction. But, it is difficult to separate the corresponding effects in the results of our experiments.

6 Vector Mapping

This chapter examines the experiment results for different vector mapping approaches and combinations of approaches for bounding sensitivity and vector mapping approaches. We use the Multivariate Laplace mechanism to add noise. The Truncated Gumbel mechanism is not considered in combination with the vector mapping approaches because a vector mapping step is already inherent to the mechanism. Thus, its outputs are word embedding vectors corresponding to real words. In the subsequent discussion, we focus on the empirical results from the experiments to derive the effects of vector mapping approaches on privacy and utility in downstream NLP tasks. There is no need to discuss the theoretical perspective here since the vector mapping approaches are a post-processing step of the DP mechanism; thus, they do neither influence the amount of noise added nor the theoretical privacy guarantees. We will analyze the performance of the approaches for different tasks on the test datasets with respect to the F1 score.

6.1 Mapping to Nearest Neighbor

We start our discussion by considering the results for the Multivariate Laplace mechanism with the additional post-processing step of mapping perturbed word embedding vectors to their nearest neighbors. Table 6.1 provides a corresponding overview. We can see that if we use the nearest neighbor mapping with an LSTM, we always get a decrease in utility across all ϵ values for both datasets. Overall, utility decreases less for larger values of ϵ. This fits the theoretical assumption of introducing less noise for larger ϵ's. The less noise we introduce, the more accurate the classification becomes. With respect to empirical privacy, we notice differences between the two datasets. For the Trustpilot dataset, mapping to nearest neighbor embeddings yields privacy gains for $\epsilon \in \{0.1, 1, 10\}$ but privacy losses for larger ϵ values. For the AG News corpus, privacy increases for ϵ greater or equal to 10. We do not get reasonable results for smaller ϵ values as the F1 score on this task is zero. This is likely due to the model's inability to learn any reasonable relationship in the data for the smaller ϵ values which correspond to larger amounts of noise being added. Thus, the results resemble those of a random guesser. We can assume that the seemingly large increases in privacy for smaller ϵ's on the Trustpilot dataset are due to the same reason. Therefore, even though the privacy change might look promising for some of the smaller ϵ's, it might not be advisable to take this approach. For the larger ϵ values, the model's performance becomes more similar to the baseline model's. With respect to the privacy-utility trade-off, this means that while we get positive values for the smaller ϵ's, these are to be considered with caution as they are most likely due to too strong perturbations of the word embedding vector which lead to the model just randomly guessing its predictions. For the larger ϵ values, there is either a decrease in both utility and privacy (Trustpilot dataset) or utility decreases more strongly than privacy (AG News corpus). Both cases do not represent a favorable trade-off. For BERT, we also end up with negative privacy-utility trade-offs. The reasons behind these are different for the two datasets. For the Trustpilot dataset, there is an increase in utility but a loss in privacy. Since the latter is larger, the trade-off turns out negative. This shows that BERT in combination with vector mapping seems to help the performance of both tasks with respect to the F1 score, which helps utility but hurts privacy. For the AG News corpus, utility decreases to a greater extent than the privacy gain, resulting in a negative privacy-utility trade-off. If we compare the running times of model training including the nearest neighbor mapping to the ones using the same setup but without the mapping, we notice a growth of about three minutes for the LSTM. For BERT, the perturbation of words adds about 10 minutes to the training of a model for one experiment setup.

Now, we compare the results of combining the Multivariate Laplace mechanism and the mapping to the nearest neighbor to the results of using only the DP mechanism without the vector mapping. Note that

Vector Mapping	DP Mechanism	Dataset	Model	ϵ	Change in Utility	Change in Privacy	Trade-off
Nearest Neighbor	Multivariate Laplace	Trustpilot	LSTM	0.1	-0.284	+0.285	**+0.002**
				1	-0.284	+0.288	**+0.003**
				10	-0.285	+0.267	-0.018
				50	-0.088	-0.014	-0.101
				100	-0.059	-0.012	-0.071
				150	-0.061	-0.008	-0.068
			BERT	150	+0.005	-0.036	-0.031
		AG News	LSTM	0.1	-*	-*	-*
				1	-*	-*	-*
				10	-0.225	+0.534	**+0.309**
				50	-0.041	+0.003	-0.038
				100	-0.019	+0.004	-0.015
				150	-0.014	+0.004	-0.010
			BERT	150	-0.008	+0.005	-0.002

* We have left out results where the model behaves like a random guesser.

Table 6.1 Performance of experiments on combining the Multivariate Laplace mechanism with mapping to the nearest neighbor (without bounding sensitivity) with respect to privacy and utility. Favorable trade-offs are indicated in bold.

results for using only the Multivariate Laplace mechanism have already been discussed separately in Section 5.1 and are visualized in Table 5.1. This comparison reveals that the size of the utility decreases is approximately the same in both situations, so one can assume that the vector mapping approach does not substantially influence utility. On the other hand, empirical privacy seems to be negatively affected by the vector mapping as the results show smaller increases in privacy or even privacy losses for some ϵ's which do not occur without the vector mapping. This is also reflected in the values for the privacy-utility trade-off, which are primarily negative and further from zero than for the experiments without vector mapping. Overall, the vector mapping seems to especially harm privacy. A possible reason is that perturbed words are being mapped back to the original word by the mapping and thereby help the simulated adversary. Overall, incorporating the mapping to the nearest neighbor to the usage of the Multivariate Laplace mechanism does not yield an improved trade-off between privacy and utility as compared to not including it.
We continue to investigate if the situation changes after including different approaches for bounding sensitivity. These have already been discussed separately in the previous chapters. Table 6.2 gives an overview of the results of including these approaches on top of the nearest neighbor mapping. In the table, we have again left out the results for the AG News corpus, where the model fails to learn because too much noise is added. This is the case for the smallest two to four ϵ values. Additionally, we only performed experiments for $\epsilon = 150$ on BERT to save some computational effort. We selected this value of ϵ because previous experiments have shown that the Multivariate Laplace mechanism mostly achieves its most favorable privacy-utility trade-offs for this value of ϵ. Therefore, we believe that it is a good selection to gain an impression of how BERT would perform for a particular combination of approaches and allow for future research to perform more extensive experiments where reasonable. To learn how the vector mapping affects the different experiment settings, we will also compare the results in Table 6.2 to the corresponding experiment settings without any vector mapping, which have been described in Chapter 5.
For normalization to unit length, the privacy-utility trade-off is negative across all ϵ values and for both datasets. Even though the absolute values of these numbers are small, they are still comparably more negative than without the inclusion of the vector mapping. Also, without the vector mapping, we saw a

positive trade-off for $\epsilon = 150$ on the Trustpilot dataset and $\epsilon = 100$ on the AG News corpus. This comparison strengthens our assumption that mapping to the nearest neighbor introduces additional distortions to the embedding space, which reduces performance on downstream tasks. However, if we consider the changes in privacy and utility, we see that with the added vector mapping, there is a smaller decrease in utility but also a smaller enhancement of privacy. Therefore, it is more likely that the mapping helps both, the performance of the main task and the simulated adversary on the privacy tasks. Thereby, the adversary seems to have a larger gain in performance, which then leads to a worse privacy-utility trade-off.
The picture looks different for the adapted version of normalization to unit length. Across different ϵ values, the decrease in utility is smaller if the nearest neighbor mapping is included. There is even an increase in utility for the two largest ϵ's on the Trustpilot dataset. However, vector mapping also helps the simulated adversary as it leads to smaller increases in privacy. Overall, the privacy-utility trade-off differs for the different datasets and ϵ values. On the Trustpilot dataset, we receive a favorable trade-off for $\epsilon = 100$ when including vector mapping while without it all trade-off values were negative. We do not have a favorable privacy-utility trade-off on the AG News corpus for any value of ϵ when including the vector mapping. Without the vector mapping, the trade-off is positive for $\epsilon = 100$. For the BERT model, there is no favorable privacy-utility trade-off for either of the two datasets.
Including the vector mapping in addition to the clipping to observed range approach has a similar effect. For most of the experiments, we notice a larger utility so it can be assumed that vector mapping helps performance on the main tasks. At the same time, it helps the adversary with the privacy tasks, which results in less private models. The only experiment, where the privacy-utility trade-off is increased using the mapping is for $\epsilon = 10$ on the AG News dataset. Here, privacy and utility are increased, leading to a trade-off, which is about four times as good as compared to not including the mapping. In the absence of any other positive trade-off, one can consider this an outlier. Again, for BERT, we do not get a favorable trade-off for any of the datasets if we include the mapping to the nearest neighbor.
If we include normalization to the observed range the effects differ depending on the dataset. On the Trustpilot dataset, the nearest neighbor mapping provides favorable privacy-utility trade-offs, which are also clearly improved compared to not including the mapping. This is mostly due to a smaller decrease in utility and slightly worse privacy. However, since the effect on utility is larger, it does not show in the trade-off. For the AG News corpus, there is a degradation of the privacy-utility trade-off visible. This can be explained by a larger decrease in utility while privacy grows stronger under vector mapping. For the two experiments on BERT, the privacy-utility trade-off is again negative for both two datasets.
A similar pattern shows for normalization to the interval $[-1, 1]^d$ when combined with the nearest neighbor vector mapping. Here, we also get a favorable privacy-utility trade-off for the Trustpilot while the trade-off is negative for the AG News corpus. However, the trade-off for the Trustpilot dataset would be even better if we do not use the vector mapping. The vector mapping has the effect of a smaller decrease in utility but it also shrinks the increase in privacy. The same trends apply to the AG News corpus. Since we already have a negative privacy-utility trade-off without the vector mapping, including it impairs the trade-off even more. Again, BERT fails to achieve a favorable trade-off in our two experiments on this model type.
For dimensionality reduction using the JL lemma, there is not a clear pattern visible. It is difficult to judge how the privacy-utility trade-off changes when the nearest neighbor vector mapping is included as for some ϵ values the trade-off is improved while for others it is reduced. This also applies to both datasets. However, the trade-off is mostly negative for the dimensionality reduction approach anyway, even without the vector mapping, so this situation does not change when the mapping is included. The BERT model yields negative privacy-utility trade-offs in this experiment setting as well.

Vector Mapping	Sensitivity Approach	Dataset	Model	ϵ	Change in Utility	Change in Privacy	Trade-off
Nearest Neighbor	Normalization to unit length	Trustpilot	LSTM	0.1	-0.292	+0.290	-0.002
				1	-0.292	+0.290	-0.002
				10	-0.292	+0.290	-0.002
				50	-0.283	+0.282	-0.001
				100	-0.130	+0.118	-0.012
				150	-0.129	+0.009	-0.119
			BERT	150	-0.007	-0.035	-0.041
		AG News	LSTM	0.1	-*	-*	-*
				1	-*	-*	-*
				10	-*	-*	-*
				50	-*	-*	-*
				100	-0.162	+0.126	-0.037
				150	-0.106	+0.053	-0.053
			BERT	150	-0.015	+0.010	-0.005
Nearest Neighbor	Normalization to unit length (adapted)	Trustpilot	LSTM	0.1	-0.292	+0.290	-0.002
				1	-0.292	+0.290	-0.002
				10	-0.292	+0.290	-0.002
				50	-0.289	+0.286	-0.002
				100	+0.120	+0.118	**+0.074**
				150	+0.066	+0.009	-0.014
			BERT	150	-0.002	-0.029	-0.030
		AG News	LSTM	0.1	-*	-*	-*
				1	-*	-*	-*
				10	-*	-*	-*
				50	-*	-*	-*
				100	-0.179	+0.130	-0.049
				150	-0.149	+0.021	-0.128
			BERT	150	-0.019	+0.002	-0.017
Nearest Neighbor	Normalization to the interval $[-1, 1]^d$	Trustpilot	LSTM	0.1	-0.288	+0.287	-0.001
				1	-0.286	+0.284	-0.002
				10	-0.279	+0.239	-0.040
				50	-0.034	+0.029	-0.005
				100	-0.009	+0.018	**+0.008**
				150	-0.012	+0.017	**+0.005**
			BERT	150	+0.003	-0.031	-0.028
		AG News	LSTM	0.1	-*	-*	-*
				1	-*	-*	-*
				10	-*	-*	-*
				50	-0.131	+0.027	-0.103
				100	-0.131	+0.020	-0.111
				150	-0.129	+0.020	-0.109
			BERT	150	-0.014	+0.002	-0.011
	Normalization to	Trustpilot	LSTM	0.1	-0.278	+0.273	-0.005
				1	-0.282	+0.275	-0.008
				10	-0.003	+0.048	**+0.045**
				50	-0.004	+0.023	**+0.020**
				100	-0.006	+0.024	**+0.018**
				150	-0.003	+0.025	**+0.022**

Nearest Neighbor	observed range	AG News	BERT	150	-0.023	-0.037	-0.060
			LSTM	0.1	-*	-*	-*
				1	-*	-*	-*
				10	-0.102	+0.044	-0.058
				50	-0.099	+0.030	-0.069
				100	-0.107	+0.027	-0.080
				150	-0.107	+0.028	-0.078
			BERT	150	-0.016	+0.003	-0.013
Nearest Neighbor	Clipping to observed range	Trustpilot	LSTM	0.1	-0.287	+0.289	**+0.002**
				1	-0.279	+0.289	**+0.010**
				10	-0.273	+0.260	-0.013
				50	-0.078	-0.005	-0.083
				100	-0.074	-0.006	-0.080
				150	-0.047	-0.005	-0.051
			BERT	150	+0.020	-0.028	-0.008
		AG News	LSTM	0.1	-*	-*	-*
				1	-*	-*	-*
				10	-0.198	+0.601	**+0.403**
				50	-0.036	+0.001	-0.034
				100	-0.016	+0.003	-0.013
				150	-0.019	+0.004	-0.017
			BERT	150	-0.012	+0.004	-0.008
Nearest Neighbor	Dimensionality Reduction (JL lemma)	Trustpilot	LSTM	0.1	-0.292	+0.290	-0.002
				1	-0.292	+0.290	-0.002
				10	-0.292	+0.290	-0.002
				50	-0.276	+0.264	-0.011
				100	-0.206	+0.056	-0.150
				150	-0.209	+0.088	-0.120
			BERT	150	-0.041	-0.031	-0.072
		AG News	LSTM	0.1	-*	-*	-*
				1	-*	-*	-*
				10	-*	-*	-*
				50	-0.443	+0.777	**+0.334**
				100	-0.324	+0.237	-0.087
				150	-0.259	+0.226	-0.033
			BERT	150	-0.016	+0.012	-0.004

Table 6.2 Performance of experiments on combining the Multivariate Laplace mechanism with mapping to the nearest neighbor and different approaches for bounding sensitivity with respect to privacy and utility. Favorable trade-offs are indicated in bold.
* We have left out results where the model behaves like a random guesser.

6.2 Random Choice Between First and Second Nearest Neighbor

To examine this vector mapping approach, we again begin by comparing the results for using only the Multivariate Laplace mechanism to those where we additionally add the mapping to a random choice between the first and second nearest neighbor. The results for the Multivariate Laplace mechanism without the vector mapping approach have been talked about in Section 5.1 and are detailed in Table 5.1. We will now check if adding the random vector mapping step improves the privacy-utility trade-off. A detailed overview of the corresponding results can be found in Table 6.3.
We see that the mapping approach always leads to a decrease in utility across all ϵ values, datasets, and models. For some of the ϵ values used with the LSTM, for example, $\epsilon = 100$ on the Trustpilot dataset, this

Vector Mapping	Dataset	Model	ϵ	Change in Utility	Change in Privacy	Trade-off
Randomly choose 1st or 2nd nearest neighbor	Trustpilot	LSTM	0.1	-0.287	+0.289	**+0.003**
			1	-0.288	+0.289	**+0.001**
			10	-0.288	+0.246	-0.042
			50	-0.036	-0.002	-0.038
			100	-0.037	-0.002	-0.038
			150	-0.063	-0.005	-0.068
		BERT	150	-0.001	+0.000	-0.001
	AG News	LSTM	0.1	-*	-*	-*
			1	-*	-*	-*
			10	-0.229	+0.657	**+0.428**
			50	-0.031	+0.002	-0.029
			100	-0.031	+0.002	-0.029
			150	-0.029	+0.002	-0.027
		BERT	150	-0.018	+0.020	**+0.003**

* We have left out results where the model behaves like a random guesser.

Table 6.3 Performance of experiments on combining the Multivariate Laplace mechanism with mapping to a random choice between the first and second nearest neighbor (without bounding sensitivity) with respect to privacy and utility. Favorable trade-offs are indicated in bold.

decrease is smaller than if there was no vector mapping. The changes in empirical privacy are typically larger for smaller ϵ's. In many cases, this is due to the fact that the noise added by the mechanism is too large and the model, therefore, resembles a random guesser. Whenever this was very obvious from a model's performance metrics, we excluded the corresponding results from the table. While we also gain privacy for the larger ϵ values, we notice that one can achieve larger privacy values if the vector mapping is not included. Together with the results on utility, we assume that the vector mapping approach helps the performance of all models in the same way. Thus, they perform better on the utility as well as on the privacy tasks, while the latter is less desirable because this would be helping a potential adversary. Depending on which task it helps more, the privacy-utility trade-off is affected. For the majority of the experiments, the trade-off is unfavorable. The effect of the vector mapping on the trade-off varies. For some experiments, the trade-off is pushed towards a less negative, better value, while for others, it becomes worse. The BERT model achieves a positive trade-off for the AG News corpus and a slightly negative one for the Trustpilot dataset. Judging from these results, BERT might profit more from this vector mapping approach. Overall, it is unclear from the results if mapping perturbed word embedding vectors to a random choice between their first or second nearest neighbor improves the privacy-utility trade-off over not using any vector mapping.
With respect to the running times, we again see an increase of about three minutes for LSTM training in comparison to the experiment setup, which is the same apart from the vector mapping. This is approximately the same growth in running time as we have seen for the first vector mapping approach, mapping to the nearest neighbor embedding.

Also for this vector mapping approach, we next combine it with the different approaches for bounding sensitivity to see if the vector mapping does have a positive effect on the trade-off between privacy and utility for them. The detailed results are presented in Table 6.4 and also compared against those from experiments on the same sensitivity approach but without any vector mapping as described in Chapter 5. This will help to assess how large the effect of the vector mapping approach is. Note that in Table 6.4, results are excluded if the model obviously fails to learn and rather behaves like a random guesser. This was the case for some of the smaller ϵ values, which most likely led to too much noise being added and the model not being able to cope with it. As before, experiments using BERT were restricted to $\epsilon = 150$ to

limit some computational effort.
Normalization to unit length yields a decrease in utility and a gain in privacy across all values of ϵ and for both datasets. For most experiments, we lose less of our utility if we include the mapping to a random choice of the first and second nearest neighbor. For the empirical privacy, there is no clear pattern visible. Some experiments profit from the vector mappings while others do not. The same applies to the privacy-utility trade-off. Nevertheless, there is a greater number of experiments, where the trade-off improves if the vector mapping is included. For the BERT model, we get a negative trade-off for the Trustpilot dataset but a positive one for the AG News corpus. With the adapted version of this normalization to unit length, we achieve very similar results. There are also more experiments, where the privacy-utility trade-off benefits from including this vector mapping approach. Furthermore, using BERT results in similar trade-offs on both datasets as with the non-adapted version of this sensitivity approach.
The approach that clips embedding vectors to the observed range also seems to profit from adding the vector mapping. We do experience a loss in utility for this combination. However, incorporating the mapping makes this loss smaller for almost all experiments on the two datasets. The change in privacy is mostly negative for the Trustpilot dataset and mostly positive for the AG News corpus. However, we cannot make any statement about the vector mapping's effect on privacy since the results vary for different values of ϵ. As a consequence, the privacy-utility trade-off is negative for most of the settings where the model performs better than random guessing. Only for the LSTM on the AG News corpus with an ϵ of 10, do we notice a positive trade-off since the gain in privacy is larger than the loss in utility. Also for the other ϵ values on AG News, the trade-offs improve slightly over the experiments without vector mapping but they still come up negative. BERT yields a negative privacy-utility trade-off for both datasets but the value achieved on the AG News corpus is not very far from zero.
When combining normalization to the observed range with the random vector mapping approach, we see a different picture. Including the vector mapping seems to lead to a stronger decrease in utility than leaving it out. At the same time, the privacy gain becomes smaller for many of our experiments. Overall, this leads to negative values for the privacy-utility trade-off in the majority of experiments. Without vector mapping, we received a higher number of positive trade-off values for normalization to the observed range.
If we include normalization to the interval $[-1, 1]^d$ with the vector mapping approach, we get utility decreases for $\epsilon \leq 50$ on the Trustpilot dataset and increases for all larger values. On the AG News dataset, utility is reduced for all values of ϵ. The values for the change in utility signal a slight improvement in utility when using the random mapping procedure on the Trustpilot dataset, while there are decreases on AG News. Thus, with respect to utility, the vector mapping seems to have a positive influence only in the case of the Trustpilot dataset. With respect to privacy, we see an inverse effect. While privacy is enhanced for all ϵ values, the AG News corpus seems to profit more. On this corpus, we get a stronger enhancement in privacy by including the vector mapping. On the Trustpilot dataset, using the mapping seems to shrink the privacy gains. Still, we get generally favorable trade-offs for the majority of the ϵ values. Additionally, for all ϵ, the trade-off is improved when the vector mapping is included. On the AG News corpus, there are only negative trade-offs and no clear improvement over not using the vector mapping is perceptible. BERT yields negative trade-offs in the experiments on both datasets.
For dimensionality reduction using the JL lemma, there is no clear picture noticeable across different ϵ values, which would tell us if utility and privacy profit or are harmed by the vector mapping approach. This is partly due to the approach leading to large performance decreases for all tasks, which further distort the changes in privacy and utility. In general, we see utility decreases and privacy enhancements for all experiments. With respect to the trade-off between the two, there are positive values for $\epsilon = 100$ on Trustpilot and for $\epsilon \in \{50, 100\}$ on AG News. Thus, using the vector mapping approach gives us one more instance of a positive trade-off. Whenever the trade-off is positive for the dimensionality reduction approach with a vector mapping, it increases in absolute value when the random mapping is incorporated. BERT yields a negative trade-off on Trustpilot and a positive one on AG News.

Vector Mapping	Sensitivity Approach	Dataset	Model	ϵ	Change in Utility	Change in Privacy	Trade-off
Randomly choose 1st or 2nd nearest neighbor	Normalization to unit length	Trustpilot	LSTM	0.1	-0.292	+0.290	-0.002
				1	-0.292	+0.290	-0.002
				10	-0.292	+0.290	-0.002
				50	-0.275	+0.281	**+0.005**
				100	-0.126	+0.125	-0.001
				150	-0.099	+0.125	**+0.026**
			BERT	150	-0.101	+0.018	-0.083
		AG News	LSTM	0.1	-*	-*	-*
				1	-*	-*	-*
				10	-*	-*	-*
				50	-*	-*	-*
				100	-0.208	+0.306	**+0.099**
				150	-0.162	+0.157	-0.005
			BERT	150	-0.053	+0.097	**+0.044**
Randomly choose 1st or 2nd nearest neighbor	Normalization to unit length (adapted)	Trustpilot	LSTM	0.1	-0.292	+0.290	-0.002
				1	-0.292	+0.290	-0.002
				10	-0.292	+0.290	-0.002
				50	-0.284	+0.286	**+0.002**
				100	-0.097	+0.083	-0.014
				150	-0.171	+0.098	-0.073
			BERT	150	-0.078	+0.016	-0.063
		AG News	LSTM	0.1	-*	-*	-*
				1	-*	-*	-*
				10	-*	-*	-*
				50	-*	-*	-*
				100	-0.204	+0.357	**+0.153**
				150	-0.207	+0.156	-0.052
			BERT	150	-0.033	+0.097	**+0.064**
Randomly choose 1st or 2nd nearest neighbor	Normalization to the interval $[-1, 1]^d$	Trustpilot	LSTM	0.1	-0.287	+0.288	**+0.001**
				1	-0.289	+0.287	-0.002
				10	-0.260	+0.251	-0.010
				50	-0.023	+0.048	**+0.025**
				100	+0.011	+0.051	**+0.062**
				150	+0.010	+0.042	**+0.052**
			BERT	150	+0.010	-0.029	-0.019
		AG News	LSTM	0.1	-*	-*	-*
				1	-*	-*	-*
				10	-*	-*	-*
				50	-0.208	+0.053	-0.155
				100	-0.208	+0.049	-0.159
				150	-0.106	+0.050	-0.056
			BERT	150	-0.019	+0.005	-0.014
Randomly choose	Normalization to	Trustpilot	LSTM	0.1	-0.279	+0.267	-0.013
				1	-0.280	+0.267	-0.013
				10	-0.044	+0.015	-0.028
				50	-0.027	+0.016	-0.011
				100	-0.027	+0.019	-0.008
				150	-0.007	+0.026	**+0.019**

1st or 2nd nearest neighbor	observed range	AG News	BERT	150	-0.001	-0.017	-0.018
			LSTM	0.1	-*	-*	-*
				1	-*	-*	-*
				10	-0.128	+0.046	-0.081
				50	-0.079	+0.018	-0.060
				100	-0.081	+0.019	-0.062
				150	-0.081	+0.019	-0.062
			BERT	150	-0.020	+0.004	-0.016
Randomly choose 1st or 2nd nearest neighbor	Clipping to observed range	Trustpilot	LSTM	0.1	-0.289	+0.286	-0.003
				1	-0.287	+0.289	**+0.001**
				10	-0.264	+0.245	-0.018
				50	-0.018	-0.010	-0.027
				100	-0.017	-0.010	-0.027
				150	-0.013	-0.003	-0.016
			BERT	150	+0.004	-0.036	-0.031
		AG News	LSTM	0.1	-*	-*	-*
				1	-*	-*	-*
				10	-0.204	+0.564	**+0.360**
				50	-0.011	+0.004	-0.007
				100	-0.013	+0.005	-0.009
				150	-0.011	+0.007	-0.004
			BERT	150	-0.006	+0.001	-0.005
Randomly choose 1st or 2nd nearest neighbor	Dimensionality Reduction (JL lemma)	Trustpilot	LSTM	0.1	-0.292	+0.290	-0.002
				1	-0.292	+0.290	-0.002
				10	-0.292	+0.290	-0.002
				50	-0.286	+0.248	-0.038
				100	-0.178	+0.246	**+0.068**
				150	-0.160	+0.159	-0.001
			BERT	150	-0.123	+0.003	-0.120
		AG News	LSTM	0.1	-*	-*	-*
				1	-*	-*	-*
				10	-*	-*	-*
				50	-0.428	+0.933	**+0.505**
				100	-0.393	+0.475	**+0.082**
				150	-0.262	+0.209	-0.053
			BERT	150	-0.053	+0.120	**+0.067**

Table 6.4 Performance of experiments on combining the Multivariate Laplace mechanism with mapping to a random choice between the first or second nearest neighbor and different approaches for bounding sensitivity with respect to privacy and utility. Favorable trade-offs are indicated in bold.
* We have left out results where the model behaves like a random guesser.

7 Discussion

The following chapter summarizes our main findings, discusses limitations, and outlines potential objectives for future research.

7.1 Main Findings

Our experiments as well as the theoretical considerations have led to some interesting findings regarding different factors influencing the privacy-utility trade-off.

Differential Privacy Mechanisms

We learned that the Multivariate Laplace as well as the Truncated Gumbel mechanism can be used to guarantee DP for individual word embeddings. Our experiments showed that they can both yield positive privacy-utility trade-offs, where the gain in privacy exceeds the loss in utility. For the Multivariate Laplace mechanism, there is a strong dependence on the privacy budget ϵ such that an increasing ϵ leads to smaller utility losses as well as smaller privacy gains. Thus, positive trade-offs between privacy and utility can often only be achieved for larger privacy budgets. For smaller privacy budgets, the large amount of noise added by this mechanism might hinder a downstream model from learning and cause it to behave like a random guesser. To spot this, one should not only look at the privacy-utility trade-off but also at the performance of tasks with respect to metrics like the F1 score. The Truncated Gumbel mechanism exhibits better chances of reaching a reasonable privacy-utility trade-off also for smaller values of ϵ since empirical utility and privacy values are more consistent across different ϵ values. However, if one has a large privacy budget at hand, the Multivariate Laplace mechanism might be a reasonable choice because it has the potential to outperform the Truncated Gumbel mechanism. Our results suggest that for smaller privacy budgets, the Truncated Gumbel mechanism is the better option.

Dependency on Evaluation Datasets and Tasks

We further noticed that the same mechanism can yield different results for the privacy-utility trade-off on different datasets in our experiments. We assume that this is not only due to characteristics of the dataset but can also be attributed to the tasks, which have been chosen to evaluate empirical utility and privacy on the respective dataset. A possible explanation for the differences in the results is that certain approaches are better suited for providing privacy for the respective dataset and tasks. Alternatively, it might be that some tasks are easier to privatize through DP than others. Providing indistinguishability for individual words might have a greater impact on the performance of a simulated adversary because of the task's design.

Theoretical and Empirical Privacy Guarantees

The influence of the dataset and tasks on the experiments' results could be one reason why we cannot establish a clear connection between theoretical privacy guarantees and empirical results for privacy and utility. In our experiments, stronger privacy guarantees do not seem to be related to larger gains in privacy or a more favorable privacy-utility trade-off.

Model Selection

Moreover, the type of model used in the downstream tasks affects empirical privacy and utility as well as the trade-off between the two. Already for the baseline models, BERT consistently outperforms the LSTM. During the experiments, BERT seems to cope better with the noise introduced by the DP mechanisms and attains F1 scores similar to the baselines on the perturbed data. Thereby, it makes part of the noise additions ineffective and leads to smaller utility losses and smaller privacy gains than an LSTM. While smaller utility losses are generally desirable, smaller privacy gains are not. In some cases, BERT even led to reductions in empirical privacy, which corresponds to giving the adversary an advantage and harms privacy. Thus, when using DP for BERT, one should keep in mind that better performance with respect to a metric like the F1 score does not necessarily go hand in hand with an enhanced privacy-utility trade-off.

Approaches to Bounding Sensitivity

Since the privacy-utility trade-off is influenced by the dataset and task it is evaluated on as well as by the DP mechanism and model used, we cannot make a generally valid statement about how an approach to bounding sensitivity enhances the privacy-utility trade-off. Still, all approaches achieved a favorable trade-off in at least one of the settings. In many of our experiment setups, combining the Truncated Gumbel mechanism and an LSTM achieved the most favorable privacy-utility trade-off across different values of ϵ. Thus, this combination can be a good starting point when deploying DP mechanisms for privatizing word embedding vectors.
When using dimensionality reduction with the JL lemma, we noticed that this approach seems to cause strong distortions to the embedding space, which generally hurts the performance on downstream tasks. While in some settings this may come with an improvement in the privacy-utility trade-off, other approaches might achieve similar improvements without hurting model performance as much. Some approaches, like clipping to the observed range, generally lead to an improved performance of downstream models with respect to the F1 score. Thereby, they do not only help utility but also a potential adversary as they enhance performance on the privacy task, which results in a decrease in privacy. This makes it a gamble whether the positive effect on the utility task is larger than the negative effect on the privacy task. Its outcome will determine if there is an improvement in the privacy-utility trade-off.

Approaches to Vector Mapping

The results for our experiments where perturbed embedding vectors are mapped to their nearest neighbor suggest that such a mapping generally increases model performance. Thereby, the utility losses compared to the baselines are smaller but privacy gains also become smaller simultaneously. Overall, this often negatively affects the privacy-utility trade-off. We hypothesize that this is due to perturbed words being mapped back to the original word during the nearest neighbor mapping. Since there is a high risk that this harms privacy, we do not see any benefit from incorporating the mapping to the nearest neighbor. We observe the same effects when we choose a random mapping to either the first or the second nearest neighbor as our vector mapping approach. It also seems to result in general increases in performance on downstream tasks, which improves utility but simultaneously supports a potential adversary. The noisy vector mapping, inherent to the Truncated Gumbel mechanism, seems to work better than a vector mapping approach added to the Multivariate Laplace mechanism since it can improve the privacy-utility trade-off more reliably.

Running Times

With respect to the running times, we notice that the DP mechanisms are the primary source of additional running time. For a single model's training, the running time increases by three minutes for both, the Multivariate Laplace mechanism and the Truncated Gumbel mechanism compared to the baseline scenario with unperturbed word embeddings. However, if the perturbations need to be pre-computed, as is the case for BERT, we need to budget more time for these pre-computations. This is also due to the vector

mapping step that needs to be added for the computation of the perturbations for BERT. In general, the computation of nearest neighbors for vector mapping further adds to the running time. Using the optimized and approximating computations provided by the Python package *faiss* [Joh+19], this increase in running time can be kept at a negligible level. Adding approaches to bounding sensitivity only increases the running times in the order of seconds. Thus, it can be advisable to try out different approaches for a concrete use case to see if one of the approaches can support the privacy-utility trade-off for the respective setting.

7.2 Limitations

It should be noted that the present work and the results of the experiments are subject to several limitations, which confine the generality of our observations.

Evaluation Datasets and Tasks

A strong limitation of this work follows from the dependency of empirical privacy and utility on the dataset and tasks that are used to determine the former. Our experiment results show that the privacy-utility trade-off can vary largely for different datasets and tasks. It raises the question of the extent to which the different datasets' results are comparable since they have been determined using different dataset-specific tasks. While the gender identification task aims to identify information implicitly contained in the texts, the named entities, which are the target of the other privacy task, are explicitly present. DP provides indistinguishability between individual words so it can be assumed that it is better suited to mask words than implicit features of the texts. Consequently, it can be assumed that the suitability of tasks to evaluate privacy varies, naturally leading to differences in the results.
Another difficulty linked to the selection of tasks arises especially if utility and privacy tasks for one dataset have very different performance levels to begin with. For example on the AG News corpus using an LSTM, our baseline for empirical utility is about 0.608 while our baseline for empirical privacy is about 0.041. Thus, an absolute deviation of, for instance, 0.01 would have a different scope and impact in the two cases. Considering relative deviations instead would also not solve the issue. Then, we would see deviations larger than 1000% if, for example, empirical privacy increases to about 0.5. Consequently, the changes in privacy and utility are not directly comparable, especially in situations where the baseline performances of the privacy and utility tasks are very different. In the case of this work, we furthermore observed strong skewness with class distributions of the different tasks. It is also unclear if and how this might affect privacy and utility.

Privacy-Utility Trade-off Metric

In addition to this, there is a lack of a generally suitable way to quantify the privacy-utility trade-off. We use a simple sum of empirical privacy and utility as a heuristic for the trade-off. Thereby, privacy and utility are weighted equally. By definition, this measure turns out positive whenever our privacy gain outweighs the utility loss. This characterizes a favorable privacy-utility trade-off. However, the sum is also positive if there is a utility gain larger than the privacy loss. Since the main purpose of applying DP is to provide privacy guarantees, this outcome does not align with the underlying goal. We identify such cases during our evaluation by considering the changes in empirical privacy and utility in addition to the heuristic for the trade-off. This way we can determine if a positive value actually signals a favorable privacy-utility trade-off. However, it would be desirable to have a more suitable metric for quantifying the trade-off, which emphasizes the favorable scenario. Using a weighted sum where privacy is weighted more heavily might be a straightforward approach. However, this would also require a careful choice of the weighting which might further depend on the concrete use case.
Alternatively, a more targeted metric for empirical privacy could make the changes in privacy and utility more independent. While it does logically make sense that an increase in privacy is connected to a decrease in utility, the way they are defined in this work results in them behaving very similarly. Per our definition,

both metrics are based on the F1 score and one is the inverse of the other. Even though they are evaluated on separate tasks, it is not surprising that their values exhibit reverse but similar trends. This, in turn, makes it difficult to derive meaningful insights about the privacy-utility trade-off because the reverse effects of empirical privacy and utility make it likely that they cancel each other out. The approaches to bounding sensitivity and the vector mapping approaches mainly target utility improvement since they do not provide any privacy guarantees by themselves. At the same time, they should not cause decreases in privacy. However, the direct connection between the two metrics makes it difficult to isolate these effects.

Selection of Mechanisms, Parameters, and Models

For our work, DP mechanisms and models were selected based on current research trends. We have selected more than one for our experiments to see if any patterns persist across different mechanisms and models. However, this limited selection cannot paint a complete picture of the approaches' influence on the privacy-utility trade-off.

Also, there are other potential influencing factors, which have not been examined in this work. The implementations of many of the methods used are based on assumptions about their parameters. For the Truncated Gumbel mechanism, the maximum and minimum inter-word distances are central parameters that determine the amount of noise the mechanism adds. In many of the experiment settings where we apply the mechanism, we use estimates for these parameters based on our training dataset. These approximations might not be valid for the development or test datasets, let alone for other data corpora than the ones we are working with. This limits the validity and generality of the theoretical privacy guarantees we provide. Additionally, it is unclear how the estimates' limited validity affect the mechanism's reliability and results.

The implementation of our dimensionality reduction approach using the JL lemma heavily relies on the target dimension, which we approximate via a bound on the Gaussian width (see Lemma 4.5.2). Using a different bound could considerably influence our experiment results for this approach. The same holds for the parameters β and δ, which we choose based on the research of [FK21]. For our vector mapping approach, which randomly selects either the first or the second nearest neighbor embedding, we had to define a tuning parameter to govern the probability of choosing the first or second nearest neighbor. Although we followed [Xu+21b] and set the tuning parameter to their best-performing value, it might not be the most suitable choice for each of our experiment settings.

Due to computational constraints, we had to cut out some of the originally planned experiments. Neither adding differentially private noise through the Multivariate Laplace or the Truncated Gumbel mechanism nor any of our approaches lead to considerable increases in running time. Still, due to the sheer number of experiments performed in this work, we had to carefully choose and restrict our experiment settings. Therefore, we had to cut out some other DP mechanisms that we had originally planned to examine next to the Multivariate Laplace and the Truncated Gumbel mechanism. It would have been too expensive to perform all experiments for varying values of ϵ for further DP mechanisms. Conditioned by the original plan to compare against further DP mechanisms, we chose not to calibrate the noise of the Multivariate Laplace mechanism to the sensitivity even for the experiment setting where sensitivity is bounded and such a calibration would have been possible. Not calibrating the noise would have facilitated the comparison with other DP mechanisms. In hindsight, now that we are only considering the Multivariate Laplace and the Truncated Gumbel mechanism, the comparison between those would have profited if the Laplace noise had been calibrated to sensitivity. This would allow us to also compare across the two mechanisms how the different approaches affect the amount of noise added. Without this calibration, for the Multivariate Laplace mechanism, only the theoretical privacy guarantees are affected by the approaches while, for the Truncated Gumbel mechanism, it is the noise. Thus, we are now limited to observing the changes in noise or theoretical privacy guarantees for each of the mechanisms individually.

The comparison between our two choices of models, LSTM and BERT, should be done with caution because of the differences in their perturbation processes. For the LSTM, we directly perturb the embeddings that the models work with and build the perturbation process around the model's embedding layer. For BERT, we prepend the perturbation process to the model. The perturbation process returns real words,

which are put back together to text sequences. The model then uses them to construct its own embeddings. We only compare BERT to LSTM with an additional vector mapping step such that the two settings are more similar. However, the additional embedding step might be part of the reason why we often observe differences in the privacy-utility trade-off for the two different models.

Computational Constraints

Another restriction of the results of our experiments, which results from computational constraints, is that we limited the experiments using BERT and a Multivariate Laplace mechanism to only using $\epsilon = 150$. Including experiments also for the other ϵ values, would have given a more complete picture.
Due to the computational constraints we also refrained from performing additional hyperparameter tuning during the experiments and instead used the same parameters that yielded superior results during hyperparameter tuning for the baseline models. As a consequence, it can be assumed that the overall model performance during the experiments could have been higher if the hyperparameters had been coordinated with the respective experiment setting. That would also better fit how it would be handled in a real-world scenario. Here, the entity who is training the model would only have access to the perturbed data and would, therefore, perform hyperparameter tuning on the perturbed data. During hyperparameter tuning, we should not just select the best-performing hyperparameter set with respect to the F1 score even if this corresponds to common practice. Keeping in mind the metrics we are using for empirical privacy and utility, purely looking at the F1 score gives utility an advantage. It essentially corresponds to selecting the hyperparameter set, which yields the best utility, and completely neglecting the privacy aspect.

7.3 Future Work

There are several possible directions for future work, which could help tackle the aforementioned limitations.

Evaluation Datasets and Tasks

To mitigate the strong dependency of the privacy-utility trade-off on the dataset and tasks used for the evaluation, it should be explored how different datasets and tasks impact privacy and utility. Furthermore, it needs to be examined which datasets and tasks are suitable for examining empirical privacy and utility after applying DP mechanisms. Thereby, one should not only consider the tasks for evaluating privacy and utility separately but also mind that the two tasks should go well together so that they both contribute equally to the privacy-utility trade-off. In the course of this research, it could also be explored how class distributions affect the adequacy of tasks for the evaluation. In the long run, it would be great to have a fixed set of carefully chosen datasets and tasks, which could be used to evaluate any kind of methods with respect to their impact on the privacy-utility trade-off.

Privacy-Utility Trade-off Metric

A second important research direction is that of a metric to quantify the privacy-utility trade-off. As described in the previous section, there are several shortcomings associated with the heuristic that was used in our experiments. Future research could look into possible alternatives that allow for a combined view of privacy and utility. Ideally, the effects of privacy and utility are decoupled so that they do not both depend on the same evaluation metric. A metric for the privacy-utility trade-off should prioritize the improvement of privacy since the primary goal of implementing DP is to safeguard privacy. Also, it should allow to adapt the weighting of privacy and utility in accordance with concrete use cases and the respective relevance of privacy and utility to them. A better-tailored metric for empirical privacy and utility could also help during hyperparameter tuning on perturbed data so that this does not need to happen only based on the F1 score and can better incorporate the privacy aspect.

Different Differential Privacy Levels

This work focuses on word-level DP, where each word is perturbed individually and we gain indistinguishability guarantees for those individual words. As it has been pointed out in existing research, there are some disadvantages of word-level DP [MWK22]. For example, the perturbed output of a text sequence using word-level DP will always be of the same length as the input sequence, which can also leak information about the input [MWK22]. Such risks can be overcome by considering sentence-, document-, or user-level DP. It might, thus, also be an interesting question to examine how the approaches to bounding sensitivity and vector mapping affect the privacy-utility trade-off if DP is provided on a different level instead. On the other hand, DP on a word-level provides the advantage that different words could be perturbed using different amounts of noise and, thereby, be masked more or less strongly than others. This can be based on the privacy risk associated with the individual words. While there is already current research that deals with this topic, it might also be interesting to consider the different approaches examined in this thesis in the context of such adapted privacy allocations.

Further Experiments

To continue the present work and get a more complete picture of different approaches to bounding sensitivity and vector mapping, future research should expand our experiments by considering variations of the parameters and estimates, which have been used in this thesis. This concerns, for example, the bound and parameters used for dimensionality reduction via the JL lemma or the tuning parameter for randomly mapping perturbed embedding vectors to either their first or second nearest neighbor. Additionally, one could generally look at further DP mechanisms, apart from the Multivariate Laplace and the Truncated Gumbel mechanism, and how they are influenced by the integration of the different approaches. In the course of this, one could also try out the Multivariate Laplace mechanism with noise calibrated to the sensitivity for the experiment settings with bounded sensitivity and compare to our results.

8 Conclusion

This thesis aimed to examine how using DP in combination with different approaches to bounding sensitivity and vector mapping influence the privacy-utility trade-off in downstream NLP tasks. We identified five approaches to bounding sensitivity and two approaches for vector mapping and evaluated them through their combination with the Multivariate Laplace and the Truncated Gumbel mechanism. For the evaluation, we first explored the implications of the different approaches from a theoretical perspective. We derived theoretical privacy guarantees and examined how the amount of noise added by the mechanisms is influenced if sensitivity is bounded or if perturbed word embedding vectors are mapped to the nearest neighbor. After that, we performed extensive practical experiments for different combinations of DP mechanisms and approaches to bounding sensitivity and vector mapping.

Our experiments showed that changes in empirical privacy and utility can be governed through the privacy budget ϵ for the Multivariate Laplace mechanism. In most cases, larger privacy budgets are related to smaller utility losses while, at the same time, they are also linked to smaller privacy gains. However, it needs to be noted that it is important for the privacy gains to be larger than the utility losses to end up with a favorable trade-off between the two. The Truncated Gumbel mechanism performs more consistently for different values of ϵ and can, therefore, provide smaller utility losses and larger privacy gains for smaller privacy budgets. Therefore, if the available privacy budget is small the Truncated Gumbel mechanism might be a more reasonable choice. If one has a large privacy budget at hand, it could also be worthwhile to give the Multivariate Laplace mechanism a try since it could potentially also yield better results in such cases. When incorporating different approaches to bounding sensitivity and vector mapping we noticed a general increase in performance of downstream tasks for many of them. While their integration can lead to smaller utility losses, we often observe an improved performance of the simulated adversary simultaneously. This corresponds to reduced privacy gains and, thus, rarely improves the privacy-utility trade-off. We assume that the observation of this effect can largely be attributed to interdependencies between the metrics for empirical privacy and utility which are both based on the F1 score and one is the inverse of the other. Therefore, it is hard to delineate the effects on privacy and utility for the different approaches in our practical experiments. Our results further suggest that the influence on the privacy-utility trade-off largely depends on the respective experiment setting. We frequently noticed differences in the results for the same approach and DP mechanism depending on the privacy budget ϵ, model, dataset, and tasks used. There was no clear pattern visible indicating that certain combinations might work better together than others. Therefore, it is advisable to test the approaches on a concrete use case to see if they can improve the privacy-utility trade-off. Since the approaches themselves can be implemented with relatively low additional computational effort, the effort of such tests mainly depends on the computational costs of the underlying model and experiment setup. A good starting point might be the combination of the Truncated Gumbel mechanism and an LSTM, which led to a favorable privacy-utility trade-off more often than other settings in our experiments.

We detected several limitations which generally restrict the empirical evaluation with respect to the privacy-utility trade-off. First, the interdependencies between metrics for empirical privacy and utility make it difficult to separate any effects on these variables from each other. Secondly, it is difficult to put the two variables in relation to each other due to the lack of a quantitative metric for the privacy-utility trade-off. Another limitation is linked to the strong dependency of experiment results on the dataset and tasks used. Therefore, future research should focus on improving the metrics for measuring empirical privacy and utility as well as the trade-off between the two. Additionally, datasets and tasks, on which these metrics are evaluated, should be carefully selected. This will greatly aid empirical evaluations in the context of DP and make it easier to perform and assess experiments similar to those performed in this thesis.

A Appendix

A.1 Hyperparameter Choices for Baseline Models

Dataset & Task	Model	Max. Length	Hidden Size	Learning Rate	Dropout
Trustpilot: Sentiment Analysis	LSTM	500	64	1e-3	0.1
Trustpilot: Sentiment Analysis	BERT	512	-	2e-5	0
Trustpilot: Gender Identification	LSTM	None	64	1e-3	0.1
Trustpilot: Gender Identification	BERT	512	-	2e-5	0
AG News: Topic Classification	LSTM	None	64	1e-5	0
AG News: Topic Classification	BERT	512	-	2e-5	0
AG News: NE Identification	LSTM	100	128	1e-3	0
AG News: NE Identification	BERT	512	-	2e-5	0

Table A.1 Hyperparameter Choices for Baseline Models

A.2 Performance of Baseline Models on Development Datasets

Dataset & Task	Model	Avg. Acc.	Std. Acc.	Avg. F1	Std. F1	Avg. Time (in sec.)	Std. Time (in sec.)
Trustpilot: Sentiment Analysis	LSTM	58.61%	0.0046	0.4582	0.0073	33.29	0.7819
	BERT	79.35%	0	0.6021	0	2265.07	0
Trustpilot: Gender Identification	LSTM	69.44%	0.0013	0.6633	0.0025	38.03	0.7992
	BERT	66.47%	0	0.6647	0	2276.36	0
AG News: Topic Classification	LSTM	69.87%	0.0150	0.5887	0.0194	43.72	0.5549
	BERT	83.38%	0	0.7865	0	3052.01	0
AG News: NE Identification	LSTM	99.89%	2.46e-05	0.9626	0.0004	50.77	0.4508
	BERT	99.90%	0	0.9663	0	3056.69	0

Table A.2 Performance of unperturbed baseline models on the development dataset (averages are over three training runs for LSTM and one training run for BERT

Acronyms

BERT Bidirectional Encoder Representations from Transformers

DP Differential Privacy

JL Johnson-Lindenstrauss

LSTM Long Short-Term Memory

NLP Natural Language Processing

RNN Recurrent Neural Network

List of Figures

List of Tables

Bibliography

[Alv+18] M. Alvim et al. "Local Differential Privacy on Metric Spaces: optimizing the Trade-off with Utility". In: *2018 IEEE 31st Computer Security Foundations Symposium (CSF)* (2018), pp. 262–267.

[BLK09] S. Bird, E. Loper, and E. Klein. *Natural Language Processing with Python.* O'Reilly Media Inc., 2009.

[Boj+17] P. Bojanowski et al. "Enriching Word Vectors with Subword Information". In: *Transactions of the association for computational linguistics 5* (2017), pp. 135–146.

[Bon+22] K. Bonawitz et al. "Federated Learning and Privacy". In: *Communications of the ACM 65, no. 4* (2022), pp. 90–97.

[Car+23] R. S. Carvalho et al. "TEM: High Utility Metric Differential Privacy on Text". In: *Proceedings of the 2023 SIAM International Conference on Data Mining (SDM)* (2023), pp. 883–890.

[Cha+13] K. Chatzikokolakis et al. "Broadening the Scope of Differential Privacy Using Metrics". In: *Privacy Enhancing Technologies: 13th International Symposium, PETS 2013, Bloomington, IN, USA, July 10-12, 2013* (2013), pp. 82–102.

[Che+23] S. Chen et al. "A Customized Text Sanitization Mechanism with Differential Privacy". In: *Findings of the Association for Computational Linguistics: ACL 2023* (2023), pp. 5747–5758.

[CAP18] G. Ciaburro, V. K. Ayyadevara, and A. Perrier. *Hands-on Machine Learning on Google Cloud Platform: Implementing Smart and Efficient Analytics Using Cloud ML Engine.* Packt Publishing Ltd, 2018.

[CNC18] M. Coavoux, S. Narayan, and S. B. Cohen. "Privacy-Preserving Neural Representations of Text". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018* (2018), pp. 1–10.

[CGR05] G. M. D. Corso, A. Gulli, and F. Romani. "Ranking a Stream of News". In: *Proceedings of the 14th international conference on World Wide Web* (2005), pp. 97–106.

[Dev+19] J. Devlin et al. "Bert: Pre-Training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (2019), pp. 4171–4186.

[Du+23] M. Du et al. "Sanitizing Sentence Embeddings (and Labels) for Local Differential Privacy". In: *Proceedings of the ACM Web Conference 2023* (2023), pp. 2349–2359.

[Dwo+06] C. Dwork et al. "Calibrating Noise to Sensitivity in Private Data Analysis". In: *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006* (2006), pp. 265–284.

[FDM18] N. Fernandes, M. Dras, and A. McIver. "Author Obfuscation Using Generalised Differential Privacy". In: *arXiv preprint arXiv:1805.08866* (2018).

[FDD19] O. Feyisetan, T. Diethe, and T. Drake. "Leveraging Hierarchical Representations for Preserving Privacy and Utility in Text". In: *2019 IEEE International Conference on Data Mining (ICDM)* (2019), pp. 210–219.

[FK21] O. Feyisetan and S. Kasiviswanathan. "Private Release of Text Embedding Vectors". In: *Proceedings of the First Workshop on Trustworthy Natural Language Processing* (2021), pp. 15–27.

[Fey+20] O. Feyisetan et al. “Privacy-and Utility-Preserving Textual Analysis via Calibrated Multivariate Perturbations”. In: *Proceedings of the 13th international conference on web search and data mining* (2020), pp. 178–186.

[Hab21] I. Habernal. “When Differential Privacy Meets NLP: The Devil Is in the Detail”. In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Langugage Processing, EMNLP 2021, Virtual Event/Punta Cana, Dominican Republic, 7-11 November, 2021* (2021), pp. 1522–1528.

[HS97] S. Hochreiter and J. Schmidhuber. “Long Short-Term Memory”. In: *Neural computation 9, no. 8* (1997), pp. 1735–1780.

[Hoo+21] S. Hoory et al. “Learning and Evaluating a Differentially Private Pre-trained Language Model”. In: *Findings of the Association for Computational Linguistics: EMNLP 2021* (2021), pp. 1178–1189.

[HS15] D. Hovy and A. Søgaard. “Tagging Performance Correlates with Author Age”. In: *Proceedings of the 53rd annual meeting of the Association for Computational Linguistics and the 7th international joint conference on natural language processing (volume 2: Short papers)* (2015), pp. 483–488.

[Hu+23] L. Hu et al. “Differentially Private Natural Language Models: Recent Advances and Future Directions”. In: *arXiv preprint arXiv:2301.09112* (2023).

[Joh+19] J. Johnson et al. “Billion-Scale Similarity Search with GPUs”. In: *IEEE Transactions on Big Data* 7.3 (2019), pp. 535–547.

[KMM22] O. Klymenko, S. Meisenbacher, and F. Matthes. “Differential Privacy in Natural Language Processing: The Story So Far”. In: *Proceedings of the Fourth Workshop on Privacy in Natural Language Processing* (2022), pp. 1–11.

[KGD22] S. Krishna, R. Gupta, and C. Dupuy. “ADePT: Auto-Encoder based Differentially Private Text Transformation”. In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume* (2022), pp. 2435–2439.

[LC21] T. Li and C. Clifton. “Differentially Private Imaging via Latent Space Manipulation”. In: (2021), pp. 210–219.

[LB12] M. Lui and T. Baldwin. “langid.py: An Off-the-shelf Language Identification Tool”. In: *Proceedings of the ACL 2012 System Demonstrations* (2012), pp. 25–30.

[LHL20] L. Lyu, X. He, and Y. Li. “Differentially Private Representation for NLP: Formal Guarantee and an Empirical Study on Privacy and Fairnes”. In: *Findings of the Association for Computational Linguistics: EMNLP 2020* (2020), pp. 2355–2365.

[Mah+22] G. Maheshwari et al. “Fair NLP Models with Differentially Private Text Encoders”. In: *arXiv preprint arXiv:2205.06135* (2022).

[MWK22] J. Mattern, B. Weggenmann, and F. Kerschbaum. “The Limits of Word Level Differential Privacy”. In: *arXiv preprint arXiv:2205.02130* (2022).

[MT07] F. McSherry and K. Talwar. “Mechanism Design via Differential Privacy”. In: *48th Annual IEEE Symposium on Foundations of Computer Science* (2007), pp. 94–103.

[MMC22] C. Meehan, K. Mrini, and K. Chaudhuri. “Sentence-Level Privacy for Document Embeddings”. In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (2022), pp. 3367–3380.

[Pan+20] X. Pan et al. “Privacy Risks of General-Purpose Language Models”. In: *2020 IEEE Symposium on Security and Privacy (SP)* (2020), pp. 1314–1331.

[PSM14] J. Pennington, R. Socher, and C. D. Manning. “Glove: Global Vectors for Word Representation”. In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (2014), pp. 1532–1543.

[PGG21] R. Plant, D. Gkatzia, and V. Giuffrida. "CAPE: Context-Aware Private Embeddings for Private Language Learning". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing* (2021), pp. 7970–7978.

[Pon+23] N. Ponomareva et al. "How to DP-fy ML: A Practical Guide to Machine Learning with Differential Privacy". In: *Journal of Artificial Intelligence Research 77* (2023), pp. 1113–1201.

[Qu+21] C. Qu et al. "Natural Language Understanding with Privacy-Preserving BERT". In: *Proceedings of the 30th ACM International Conference on Information and Knowledge Management* (2021), pp. 1488–1497.

[SMR08] H. Schütze, C. D. Manning, and P. Raghavan. *Introduction to Information Retrieval*. Cambridge: Cambridge University Press, 2008.

[SR20] C. Song and A. Raghunathan. "Information Leakage in Embedding Models". In: *Proceedings of the 2020 ACM SIGSAC conference on computer and communications security* (2020), pp. 377–390.

[Wu+17] X. Wu et al. "Bolt-on Differential Privacy for Scalable Stochastic Gradient Descent-Based Analytics". In: *Proceedings of the 2017 ACM International Conference on Management of Data* (2017), pp. 1307–1322.

[Xu+21a] N. Xu et al. "Density-Aware Differentially Private Textual Perturbations Using Truncated Gumbel Noise". In: *The International FLAIRS Conference Proceedings* 34 (2021).

[Xu+20] Z. Xu et al. "A Differentially Private Text Perturbation Method Using a Regularized Mahalanobis Metric". In: *arXiv preprint arXiv:2010.11947* (2020).

[Xu+21b] Z. Xu et al. "On a Utilitarian Approach to Privacy Preserving Text Generation". In: *PrivateNLP 2021* 11 (2021).

[ZC22] Y. Zhao and J. Chen. "A Survey on Differential Privacy for Unstructured Data Content". In: *ACM Computing Surveys (CSUR) 54* (2022), pp. 1–28.