Differential Privacy, Linguistic Fairness, and Training Data Influence
Impossibility and Possibility Theorems for Multilingual Language Models
Rust, Phillip; Søgaard, Anders

Published in:
Proceedings of Machine Learning Research

Publication date:
2023

Document version
Publisher’s PDF, also known as Version of record

Document license:
Unspecified

Citation for published version (APA):
Differential Privacy, Linguistic Fairness, and Training Data Influence: Impossibility and Possibility Theorems for Multilingual Language Models

Phillip Rust 1, Anders Søgaard 1

Abstract

Language models such as mBERT, XLM-R, and BLOOM aim to achieve multilingual generalization or compression to facilitate transfer to a large number of (potentially unseen) languages. However, these models should ideally also be private, linguistically fair, and transparent, by relating their predictions to training data. Can these requirements be simultaneously satisfied? We show that multilingual compression and linguistic fairness are compatible with differential privacy, but that differential privacy is at odds with training data influence sparsity, an objective for transparency. We further present a series of experiments on two common NLP tasks and evaluate multilingual compression and training data influence sparsity under different privacy guarantees, exploring these trade-offs in more detail. Our results suggest that we need to develop ways to jointly optimize for these objectives in order to find practical trade-offs.

1. Introduction

One of the open challenges in AI is bridging the widening digital language divide by providing technologies that work well for all languages. Multilingual language models such as mBERT (Devlin et al., 2019), XLM-R (Conneau et al., 2020a), and BLOOM (Scao et al., 2022), facilitate transfer between closely related languages, enabling roll-out of technologies for low-resource languages, and are used for a wide range of real-world applications in many languages—e.g., from named entity recognition (Khalifa et al., 2021) to legal document classification (Wang & Banko, 2021). Generalization across languages is challenged by typological divides, language families, or scripts (Singh et al., 2019; Dufter & Schütze, 2020) and finding architectures that best facilitate such transfer, achieving optimal multilingual compression (Ravishankar & Søgaard, 2021) through parameter sharing (rather than compartmentalization), remains an open research problem.

With the widespread adaptation of multilingual language models also comes responsibility and requirements that models are trustworthy (Pruksachatkun et al., 2021). What does trustworthiness amount to for multilingual language models? A crucial requirement is that multilingual NLP models perform equally well across languages, not favoring any languages over others. (Choudhury & Deshpande, 2021) refer to this property as linguistic fairness. Linguistic fairness is defined as zero variance across language-specific losses, typically estimated on held-out data.1

Another crucial requirement is transparency, i.e., the ability to say why models make particular predictions. Methods to achieve transparency come in two flavors; Some methods—commonly referred to as feature attribution methods—present rationales behind predictions in terms of input token attributions, but such rationales are limited in that they cannot explain predictions motivated by the absence of input tokens or the presence of particular token combinations. Feature attribution methods have also been shown to be unreliable (Kindermans et al., 2019; Arun et al., 2020). Other methods highlight training data influence, i.e., provide influential data points as rationales for decisions. Often referred to as instance-based interpretability methods, they are argued to be more useful across different NLP tasks (Han et al., 2020; Han & Tsvetkov, 2021; Zhou et al., 2021b). We refer to the objective of achieving sparse training data influence, i.e., strong instance-interpretabiliy, as training data influence sparsity. Finally, for many NLP applications, we further need our models to be private, for which differential privacy (DP; Dwork, 2006) provides a theoretically rigorous framework.

1This definition of linguistic fairness is an instantiation of equal risk fairness or overall performance parity, i.e., equal model performance across groups (Berk et al., 2018; Verma & Rubin, 2018; Williamson & Menon, 2019), which balances precision-based and recall-based metrics and is considered more relevant than calibration-based metrics for standard NLP applications. Since the three are mutually exclusive (Miconi, 2017), we ignore calibration and balance precision and recall.
The trustworthiness objectives as defined above have primarily been considered in a monolingual context, and are often (falsely) assumed to be independent [Ruder et al., 2022]. Our paper investigates the extent to which these objectives align or are at odds. We do so in a multilingual setting and show how multilinguality presents options and challenges. Our theoretical contributions show that while privacy and linguistic fairness are compatible through multilingual compression, privacy and training data influence are not, and our empirical results indicate that these objectives interact in non-linear ways.

Contributions We begin (in §2) with a theoretical exploration of differential privacy, training data influence, and linguistic fairness in the context of multilingual language models. We show that differential privacy and training data influence sparsity are fundamentally at odds, a result which is not limited to the multilingual setting. While differential privacy and fairness are often said to be at odds, we also show that differential privacy and linguistic fairness over languages are compatible in the multilingual setting, as a result of compression.

Subsequently (in §§3–5), we present empirical results on the impact of differentially private fine-tuning on multilingual compression and training data influence: We analyze the effect of such fine-tuning on the multilingual compression of large LMs and find that it is possible to achieve (i) high compression with strong privacy at the cost of performance; (ii) high compression with high privacy at the cost of compression; or (iii) privacy and accuracy at the cost of compression. Since we show in §2 that performance, privacy and compression are theoretically compatible, this leaves us with an open problem: How do we practically optimize for both performance, privacy and compression? Furthermore, we compare four (proxy) metrics for quantifying multilingual compression—sentence retrieval, centered kernel alignment (CKA; Kornblith et al., 2019), IsoScore (Rudman et al., 2022), representational similarity analysis (RSA; Kriegeskorte et al., 2008; Edelman, 1998)—and discuss their usefulness for balancing these trade-offs.

Finally, we show that LMs exhibiting high multilingual compression are less instance-interpretable in that they make highlighting training data influence more difficult.

In sum, our work shows that linguistically fair and private high-performance multilingual models are possible, even if learning them is challenging. However, training data influence methods will fail for such models.

2. Theoretical Exploration

We consider language model learning and fine-tuning in a multilingual setting, in which our training data \( D = D_1 \cup \ldots \cup D_{|L|} \) is the union of disjoint training data from \(|L|\) different languages. We consider the interaction of differential privacy, training data influence and linguistic fairness with performance and compression in this setting.

Preliminaries We briefly introduce our formal definitions here: A randomized algorithm, here model, \( M: D \to Y \) is \( \varepsilon_p \)-differentially private (Dwork, 2006) iff for all adjacent datasets \( D, D' \in \mathcal{D} \) and all \( Y \subseteq Y \), \( \Pr[M(D) \in Y] \leq \exp(\varepsilon_p) \cdot \Pr[M(D') \in Y] \). Adjacent means that the datasets differ by exactly one example \( x_{\text{adj}} \).

A model \( M \) is said to be \( \varepsilon_l \)-instance-interpretable, i.e., having sparse training data influence, iff for any \( D, D', D'' \in \mathcal{D} \) with \( D' = D \setminus \{x_{\text{adj}}\}, D'' = D \setminus \{x'\} \), and \( x_{\text{adj}} \neq x' \), where \( x_{\text{adj}} \) is the most influential training data point under leave-one-out influence [Dwork 2006] it holds that \( \Pr[M(D) \in Y] - \Pr[M(D') \in Y] > \exp(\varepsilon_l) \cdot (\Pr[M(D) \in Y] - \Pr[M(D'') \in Y]) \). In other words, \( x_{\text{adj}} \) had more influence on \( M \) than any other data point \( x' \) by some margin \( \exp(\varepsilon_l) \) (Koh & Liang, 2017).

A model \( M \) is said to be fair if for a group partitioning \( g(D) \to D_{g_1}, \ldots, D_{g_n} \) into smaller samples and for some loss function \( \ell \), e.g., 0-1 loss, \( \ell(M(D_{g_l})) = \ell(M(D_{g_q})) \) (Williamson & Menon, 2019). A model that is fair for a group partitioning by languages is said to be linguistically fair (Choudhury & Deshpande, 2021).

Finally, a model \( M \) exhibits perfect multilingual compression when it outputs identical representations for semantically equivalent inputs irrespective of the input language. Formally, for a pair of translation equivalent sentences, \((i_j, i_q)\), the representations of \( i_j \) and \( i_q \) are identical at any layer \( l \) of the model, i.e \( M'(i_j) = M'(i_q) \).

Note how standard empirical risk minimization is not private, since it is a linear combination of training samples near the decision boundary, and if \( D \) and \( D' \) differ in one of those, the classifier changes significantly.

Leave-one-out here means \( D' = D \setminus \{x_{\text{adj}}\} \) and is the gold standard for instance-based methods, which explains the close connection to DP where we also deal with adjacent datasets.
In the following paragraphs, we discuss under what conditions DP, training data influence, linguistic fairness, and multilingual compression are at odds or are compatible, and how these conditions align with common scenarios in multilingual NLP.

**Differential Privacy and Training Data Influence Sparsity** We first show that differential privacy and training data influence sparsity are fundamentally at odds:

**Theorem 1.** A model $M$ becomes less $\varepsilon_i$-instance-interpretable as it becomes more $\varepsilon_p$-differentially private, and vice-versa.

*Proof.* Let $\mathbb{P}(M(D) \in Y)$ be abbreviated as $p$, $\mathbb{P}(M(D') \in Y) = \mathbb{P}(M(D \setminus \{x_{\text{test}}\}) \in Y)$ be abbreviated as $p_d$, and let $\mathbb{P}(M(D'') \in Y) = \mathbb{P}(M(D \setminus \{x'\}) \in Y)$ be abbreviated as $p_2$. Assume that $M$ is $\varepsilon_i$-instance-interpretable and $\varepsilon_p$-differentially private.

If $M$ is $\varepsilon_p$-differentially private, it holds that

$$p \leq \exp(\varepsilon_p) \cdot p_d \tag{1}$$

If $M$ is also $\varepsilon_i$-instance-interpretable, it also holds that

$$p \leq p_d \tag{2}$$

Step (iv) follows from Equation (1). We can now see from Equation (iv) that $p$ increases with increasing $\varepsilon_i$, i.e., the model becomes less differentially private as it becomes more instance-interpretable, and vice-versa. \(\square\)

This result is not limited to the multilingual setting.

**Differential Privacy and Linguistic Fairness** Fairness and differential privacy are occasionally at odds, as shown in [Bagdasaryan et al. 2019, Cummings et al. 2019, Chang & Shokri 2021, Hansen et al. 2022] and in the multilingual setting, fairness and privacy can be compatible (for the common definitions above). We first note that there is a trivial solution to obtaining differential privacy and linguistic fairness (a joint optimum), namely randomness. This simply shows that the two objectives can be simultaneously satisfied. Next, imagine a perfectly compressed multilingual language model trained on a multi-parallel dataset.

**Theorem 2.** If a model $M_D$ trained on parallel data from $|L| \geq 2$ languages, $D = \{\ldots, i_1, \ldots, i_{|L|}, \ldots\}$, with $i_j$ and $i_q$ being translation equivalents, is perfectly multilingually compressed, then it is $\varepsilon_q$-differentially private.

*Proof.* Since $M_D$ is perfectly compressed, the representation of $i_j$ is identical to $i_q$ at any layer $l$, i.e., $M_D^l(i_j) = M_D^l(i_q)$. This gives us strong $k$-anonymity [Li et al. 2012] in the representation space of $M_D$, with $k = |L|$ and all dimensions as quasi-identifiers. Since $k$-anonymity is not obtained through a deterministic (reversible) procedure, but a randomly initialized learning procedure with random sampling, and since our attributes are randomly initialized, $k$-anonymization entails differential privacy in our setting. $M_D$, given perfect compression and convergence, is $0$-differentially private, i.e., the probability distribution of $M_D$ is unaffected by the removal of any single row. \(\square\)

It follows directly from perfect compression that $M_D$ is also linguistically fair because identical representations imply identical performance across languages. It is therefore an immediate corollary of the above result that a linguistically fair model can be differentially private.

While the assumptions of a perfectly compressed model and clean multi-parallel dataset rarely hold up in practice and there is no obvious way to satisfy them while maintaining utility, the practical significance of this result is a reminder that multilingual training converges towards $k$-anonymization, and that safe $k$-anonymization of the representation space, if obtained, would provide us differential privacy. In the absence of strong guarantees, increasing the number of training languages (larger $k$) would strengthen privacy [Li et al. 2012]. Our empirical results below suggest that we can often obtain strong privacy and strong compression, but at the cost of performance.

### 3. Experimental Setup

In our experiments, we investigate the relation between the performance and multilingual compression of fine-tuned multilingual language models, and their privacy and training data influence. We rely on a commonly used multilingual NLP.

---

7 Differential privacy meaningfully protects any individual training example. However, sensitive information may be repeated across many training examples, so $\varepsilon$-DP does not necessarily prevent leakage of such information at the granularity of individual people, real-world events, etc. For example, in our multilingual setting, an attacker may still gain access to a social security number learned by the model, but they will be unable to identify whether the number was leaked in a particular language.

8 Several authors have considered practical trade-offs between privacy and fairness, including [Jagielski et al. 2019, Lyu et al. 2020], [Pannekoek & Spigler 2021], and [Liu et al. 2021b].
Differential Privacy, Linguistic Fairness, and Training Data Influence

gual pretrained language model, which we fine-tune with different levels of \((\epsilon, \delta)\)-differential privacy on two common NLP tasks and evaluate using metrics of compression and training data influence\footnote{For completeness, we explain the difference between \(\epsilon\)-DP and \((\epsilon, \delta)\)-DP in Appendix B.}. This section presents the pretrained language model, the tasks, the training protocol, the metrics of compression and training data influence, and the evaluation procedure.

**Model** We use a pretrained XLM-R Base (Conneau et al., 2020\footnote{One limitation of this selection is that we only consider classifiers with \(\epsilon\)-DP}), which is a 12-layer encoder-only transformer with \(\sim 277\)M parameters and 250K vocabulary size trained on CC-100 (100 languages) via masked language modeling.

**Tasks and Data** We fine-tune in a zero-shot cross-lingual transfer setting for part-of-speech (POS) tagging and natural language inference (NLI). Why these tasks? First, while POS tagging is driven by lower-level syntactic features, NLI requires a higher-level understanding (Lascher et al., 2020\footnote{See Table 2 for language details.}). Second, we can leverage multi-parallel corpora for multilingual fine-tuning and zero-shot cross-lingual transfer in both tasks, which helps eliminate confounders\footnote{We exclude English to keep the number of languages balanced and because the combined corpus is already biased towards Indo-European with Latin scripts (see Table 2).}. For POS tagging, we use the Parallel Universal Dependencies (UD) v2.8 (Nivre et al., 2020; Zeman et al., 2021), which contains 1000 sentences parallel across 15 languages. We train in 7 of these languages (FR, IT, JA, PT, TH, TR, ZH)\footnote{Regardless of test split size, each language contributes equally to the mean accuracy reported in Figure 1} and use the remaining 7 languages (AR, DE, ES, HI, ID, KO, RU) for validation. This split ensures that (1) we both train and evaluate on typologically diverse language samples, (2) there exist additional UD v2.8 treebanks in our validation set languages that we can harness for testing, and (3) there exist parallel sentences in our training set languages that we can harness to evaluate multilingual compression. We use the test splits of the following treebanks for testing: Arabic-PADT, German-GSD, Spanish-GSD, Hindi-HDTB, Indonesian-GSD, Korean-Kaist, and Russian-SynTagRus. Appendix Table 4 lists the treebanks’ sizes\footnote{For all languages \((\epsilon, \delta)\)-DP (Canonne et al., 2020)}. For NLI, we rely on the XNLI dataset (Conneau et al., 2018), which contains \((\text{premise}, \text{hypothesis}, \text{label})\)-triplets multi-parallel across 15 languages. We, again, train in 7 of these languages (BG, ES, FR, HI, TR, VI, ZH), exclude the original English data, and validate in the remaining 7 languages (AR, DE, EL, RU, SW, TH, UR). We train and validate our models on the original XNLI validation data (7500 examples per language), and we test the models on the original test data (15000 examples per language) in the validation set languages.

The idea to train and validate on the same sentences (in different languages) while testing on sentences from different treebanks (as we do for POS) or a different dataset split (as for XNLI) is to induce a slight distributional shift between validation and test data for the same language sample. This shift lets us evaluate the regularization strength of the gradient noise added by the DP-optimizer.

**Training** We employ the standard fine-tuning procedures for token classification (POS) and sequence classification (XNLI) proposed by (Devlin et al. 2019). Similar to (Li et al. 2022), we use DP-AdamW (i.e., the DP-SGD algorithm (Abadi et al. 2016) applied to the AdamW optimizer with default hyperparameters (Loshchilov & Hutter, 2019)), which is a 12-layer encoder-only transformer with \(\sim 277\)M parameters and 250K vocabulary size trained on \(\sim 17,180\)K examples. For POS tagging, where \(|D_{\text{train}}| = 7000\) is the length of the training dataset, and \(\delta = 1e^{-6}\) for XNLI because it turned out to be much harder to achieve convergence than for POS. Even with the looser DP bounds from \(\delta = 1e^{-6}\), we were unable to find a hyper-parameter setting for \(\epsilon = 1\) where the fine-tuned model was substantially better than random guessing.

To select suitable hyper-parameters for XNLI, we use slightly higher learning rates. For XNLI, we select hyper-parameters in a way that matches the sampling rate (\(\epsilon, \delta\))-DP (Canonne et al., 2020). Hyper-parameter tuning on private data—which the POS and XNLI data in our study simulate—has been shown to incur additional privacy leakage (Liu & Talwar 2019). Hyper-parameters are tuned according to a particular \((\epsilon, \delta)\)-DP (Canonne et al., 2020). For XNLI data, we use the Rényi differential privacy (RDP; Mironov 2017), which is a recently proposed privacy accountant to convert non-private training (\(\epsilon\)-DP) into \((\epsilon, \delta)\)-DP (Canonne et al., 2020). A hyper-parameter tuning on private data—which the POS and XNLI data in our study simulate—has been shown to incur additional privacy leakage (Liu & Talwar 2019, Papernot & Steinke 2022). Therefore, we try to keep hyper-parameter tuning to a minimum and rely on sensible priors to select a suitable range of hyper-parameters. For POS, we find that the range of good hyper-parameters for non-private settings transfers well to private settings if we just use slightly higher learning rates. For XNLI, we select hyper-parameters in a way that matches the sampling rate (Li et al. 2022) found to suit the NLI tasks in the GLUE benchmark (Wang et al. 2018) well\footnote{The sampling rate \(q = \frac{D_{\text{train}}}{|D_{\text{train}}|} B\) denoting the batch size.}. Accordingly, we train with a maximum sequence length of 128 for 10 epochs.

\footnote{The sampling rate \(q = \frac{D_{\text{train}}}{|D_{\text{train}}|} B\) denoting the batch size.}
with a total batch size of 96 for POS and 30 epochs with batch size 512 for XNLI. At each privacy budget, we train models (3 random initializations each) with 6 learning rates for POS ($1e-4, 3e-4, 5e-4, 7e-4, 1e-5, 3e-5, 1e-6$) and 3 learning rates for XNLI ($3e-4, 4e-4, 5e-4$ for private models and $9e-5, 1e-4, 2e-4$ for non-private models). Based on the validation accuracy we then select the 5 best settings for each privacy level and task, listed in Appendix C. The learning rate is linearly decayed after 50 warm-up steps for POS and without warm-up for XNLI. We perform gradient clipping (per-sample in private settings) with a threshold of 0.1. Weight decay is set to 0.01.

### Quantifying Multilingual Compression

We present four metrics of multilingual compression: A common proxy task to measure the quality of cross-lingual representations is sentence retrieval (Artetxe & Schwenk, 2019). Dufter & Schütze (2020) propose to use linear centered kernel alignment (CKA) as a similarity index for neural network representations. It is defined as

$$P = \frac{1}{2m} \sum_{i=1}^{m} I_{\arg\max_k R_{ik} = i} + I_{\arg\max_k R_{ki} = i}. \quad (3)$$

Here, $R \in \mathbb{R}^{m \times m}$ denotes the matrix of cosine similarities $R_{ij} = \cos(e_i^q, e_j^r)$ between the $m$ sub-word representations $e_i^q$ and $e_j^r$ from a LM at indices $i$ and $j$ for a set of parallel sentences in the languages $q$ and $r$.  

(Kornblith et al., 2019) propose to use linear centered kernel alignment (CKA) as a similarity index for neural network representations. It is defined as

$$\text{CKA}(X, Y) = \frac{\|Y^T X\|_F^2}{\|X^T X\|_F \|Y^T Y\|_F}. \quad (4)$$

For LMs, the matrices $X$ and $Y$ are obtained by mean-pooling $n$ sub-word representations at model layer $l$ (Conneau et al., 2020b; Glavaš & Vulić, 2021). Typically, $X$ and $Y$ correspond to the representations from two different models for identical examples (Kornblith et al., 2019).

### Quantifying Training Data Influence

Training data influence metrics can help us gain an understanding of the inner workings of a model (Koh & Liang, 2017; Yeh et al., 2018; Charpiat et al., 2019; Koh et al., 2019; Pruthi et al., 2020; Basu et al., 2020; K & Søgaard, 2021; Zhang et al., 2021; Kong & Chaudhuri, 2021, inter alia). Such metrics are approximations of leave-one-out-influence. (Pruthi et al., 2020) proposed a both effective and practical method, called TracInCP, to compute the influence of a training example $z$ on the model’s prediction for another example $z'$, which could be a test example or $z$ itself (called the self-influence). The influence is computed as follows:

$$\text{TracInCP}(z, z') = \sum_{i=1}^{k} \eta_i \nabla \ell(\theta_i, z) \cdot \nabla \ell(\theta_i, z'), \quad (5)$$

(Phang et al., 2021). We instead use the representations from a single model for a parallel sentence pair $(s_q, s_r)$ in languages $q$ and $r$ as $X$ and $Y$, respectively, to study the similarity of representations across languages, similar to (Muller et al., 2021) and (Conneau et al., 2020b). (Yang et al., 2022) also use CKA as a metric of compression. IsoScore (Rudman et al., 2022) is an isotropy metric, computed as outlined in Appendix D that quantifies the degree to which a point cloud uniformly utilizes the vector space. In our context, this point cloud corresponds to the $n$ sub-word representations of all examples in a corpus at layer $l$. Prior work has shown that anisotropic representation spaces, such as the embedding spaces of large LMs (Elhayarajh, 2019), suffer from so-called representation degeneration (Gao et al., 2019), and that the isotropy of a model’s representation space correlates with its task performance (Zhou et al., 2019; Wang et al., 2020; Zhou et al., 2021a; Rajaei & Pilehvar, 2021, inter alia). High isotropy also means languages are not compartmentalized and should therefore correlate with high compression.

Representational similarity analysis (RSA; Kriegeskorte et al., 2008; Edelman, 1998) was originally introduced in the field of cognitive neuroscience to analyze the similarity of fMRI activity patterns, but it is also applicable to neural network representations (Bouchacourt & Baroni, 2018; Chrupała 2019; Chrupała & Alshahi, 2019; Leport & McCoy, 2020; He et al., 2021, inter alia), e.g., to analyze their similarity across languages. RSA measures the similarity between the representational geometries (i.e., the arrangement in the vector space) of two sets of representations. The representational geometry is determined through pairwise (dis)similarity metrics, and similarity is typically measured using a rank-based correlation metric such as Spearman’s $\rho$ (Diedrichsen & Kriegeskorte, 2017).

(Conneau et al., 2020b) proposed a both effective and practical method, called TracInCP, to compute the influence of a training example $z$ on the model’s prediction for another example $z'$, which could be a test example or $z$ itself (called the self-influence). The influence is computed as follows:

$$\text{TracInCP}(z, z') = \sum_{i=1}^{k} \eta_i \nabla \ell(\theta_i, z) \cdot \nabla \ell(\theta_i, z'), \quad (5)$$

$\text{TracInIdeal}$, which is impractical to compute, through model checkpoints taken during training (Pruthi et al., 2020).
where $\eta_i$ is the learning rate and $\nabla \ell(\theta_i, z)$ is the gradient of the loss w.r.t. the model parameters $\theta_i$ and inputs $z$ for the $i$-th model checkpoint. We will use TraICP as an approximation of training data influence in our experiments.

Evaluation. We evaluate our models both during and after fine-tuning. For POS, we evaluate every 100 steps, and for XNLI, every 200 steps. We measure zero-shot cross-lingual transfer performance on the validation and test data by accuracy (token-level for POS and sequence-level for XNLI). To account for randomness, we take the mean of the best 5 seeds for each privacy budget.

The measures of multilingual compression (sentence retrieval precision, CKA, IsoScore, RSA) are computed using distinct evaluation corpora comprising parallel sentences for all languages pairs in the respective training set language sample. For models trained on XNLI, we use 3000 sentence pairs per language pair from the TED 2020 corpus (Reimers & Gurevych, 2020) and 3500 pairs from the WikiMatrix dataset (Schwenk et al., 2021). For models trained for POS, we use 3500 pairs from TED 2020, 3500 pairs from WikiMatrix, and 900 pairs from Tatoeba.

Following (Duffter & Schütze, 2020), we evaluate the models at layers 0 and 8, which complement each other well with regard to the properties they capture, e.g., multilinguality and task-specificity (Choenni & Shutova, 2020; de Vries et al., 2020; Muller et al., 2021). We compute the sentence retrieval precision between language pairs and take the mean. The IsoScore is computed for the contextualized representations of all examples in the respective corpus at once. In contrast, CKA and RSA scores are also computed per language pair, and then averaged across those. For RSA, we use $D = 1 - \text{Spearman's } \rho$ and $S = \text{Spearman's } \rho$ as the dissimilarity and similarity metrics, respectively. Finally, we average results for all four metrics across TED 2020, WikiMatrix, and Tatoeba, the two layers, and the 5 best seeds for each privacy budget.

For comparison, we also compute all metrics for the original pretrained and a randomly initialized XLM-R model.

4. Results

Privacy, Compression, Performance. We now empirically investigate the relationship between differential privacy, multilingual compression, and cross-lingual transfer performance. We present aggregated results in Figure 1 and non-aggregated results in Appendix G. We observe that the zero-shot accuracy decreases as we fine-tune with stronger privacy guarantees (Figures 1a and 1h), which is expected due to the privacy–utility tradeoff (Geng et al., 2020). In particular, the relatively small sizes of our training datasets make private LM fine-tuning more challenging (Kerrigan et al., 2020; Habernal, 2021; Senge et al., 2022). Yu et al. (2022) because, for a fixed number of update steps, the gradient noise added per update step grows as the size of the training dataset decreases (Tramer & Boneh, 2021; Maharjan et al., 2018). Note that although the private models tend to underperform the non-private models by a large margin on the validation set (>30% for XNLI, as shown in Appendix Table 6), the performance gap on the test set is noticeably smaller, showing that training with differential privacy, like other noise injection schemes (Bishop, 1995), is also a form of regularization.

Figures 1b and 1g display sentence retrieval precision when fine-tuning with different privacy budgets. The highest compression is achieved by the non-private models. The second-highest compression is achieved for $\varepsilon = 1$, our most private models. Both suggest non-linear privacy–compression interactions, with POS showing lowest compression for $\varepsilon = 30$ (or higher) and XNLI showing lowest compression for $\varepsilon = 8$. The results are very similar for IsoScore (Figures 1d and 1i) and also similar, albeit less pronounced for CKA (Figures 1c, 1f). RSA, in contrast, exhibits very low scores for highly private models; see Appendix F.

These results show that we can achieve strong compression and strong performance at the cost of privacy ($\varepsilon = \infty$), strong compression and strong privacy at the cost of performance ($\varepsilon = 1$), or trade-off performance and privacy at the cost of compression (e.g., $\varepsilon = 8$). It may seem counter-intuitive that multilingual compression and cross-lingual transfer performance are not strictly correlated. However, in the fine-tuning setting, we can sacrifice task-specific knowledge in favor of multilingual compression, which leads to poor performance. Vice-versa, a model may exhibit

24 We extract sentence pairs from Tatoeba using the tatoebatools library (https://github.com/LBeaudoux/tatoebatools).
25 We exclude TH from the WikiMatrix and Tatoeba evaluation sets for POS as there are insufficiently many sentence pairs available between TH and the remaining languages.
26 Sentence retrieval is bidirectional (see Eq. 3). Given $|L|$ languages, we therefore average over the full $|L| \times |L|$ language pair matrix, only excluding the main diagonal.
27 CKA and RSA are symmetrical. Given $|L|$ languages, we thus only use the upper triangle of the $|L| \times |L|$ language pair matrix, still excluding the main diagonal.
28 This is consistent with Zhelezniak et al., 2019 and Lepor & McCoy, 2020 who show that Spearman’s $\rho$ is more suitable for RSA with embeddings than conventional similarity metrics such as cosine similarity.
29 The randomly initialized XLM-R model shows high CKA scores. This is explained by the high dimensionality ($d = 768$) of the contextualized representations, considering that CKA saturates with increasing network width (Kornblith et al., 2019), and the high centroid similarity of random activations.
We also find that in some fine-tuning settings, e.g., lingual transfer performance in the downstream tasks. An example for the former case is the pretrained (but not fine-tuned) XLM-R, which scores highly in multilingual predictions without actually relying on cross-lingual signal. As shown in Figure 1, this finding clearly establishes that sentence retrieval performance typically drops (i.e., compression worsens) over the course of fine-tuning.

Lastly, retrieval and CKA scores are always highest between typologically similar languages and languages over-represented in pretraining (see Table 2 for a comparison across languages) across all levels of privacy, as shown by the non-aggregated results in the Appendix Figures 6-13. This finding thus extends conclusions from prior work (Pires et al., 2019; Wu & Dredze, 2019; K et al., 2020; Lauscher et al., 2020) to private models.

5. More multilingual, less interpretable?

**Metric** To answer this question, we introduce InfU (Influence Uniformity), a measure of uniformity based on TracInCP influence scores for each training example in the multiparallel dataset $D = \{l_1, \ldots, l_{|L|}\}$, with $l_j$ and $l_k$ translation equivalents. We compute InfU for $\mathcal{M}$ and the translation equivalents $i = \{i_1, \ldots, i_{|L|}\}$ as follows:

$$\text{InfU}(i) = \frac{1}{|L|} \sum_{k=1}^{|L|} H(\sigma(\text{TracInCP}(i_k, i)))$$

where $H$ is the entropy with $\log_{|L|}$ and $\sigma$ is a softmax used to obtain a probability distribution over influence scores. InfU is maximized ($\text{InfU} = 1$) for uniform influence scores, fulfilling $\text{TracInCP}(i_j, i_k) = \text{TracInCP}(i_q, i_r)$, $\forall j, k, q, r \in L$. This means a perfectly multilingual model that yields equivalent representations for translation equivalent examples obtains $\text{InfU} = 1$. In this scenario of maximum uniformity our model is also the least instance-interpretable because training data influence is minimally sparse, so we cannot easily identify influential examples for a prediction. We use InfU to study to what extent influence sparsity aligns with the metrics privacy and cross-lingual performance.

Figure 1: Task performance, sentence retrieval, CKA, IsoScore, and RSA results when fine-tuning with different privacy guarantees ($\infty$=non-private). We add the original pretrained XLM-R and XLM-R with randomly initialized weights for comparison. The results show how non-private fine-tuning balances multilingual compression and task performance. Strongly private fine-tuning ($\varepsilon = 1$) is compatible with high compression (retrieval, CKA, IsoScore), but not with task performance. For medium levels of privacy (e.g., $\varepsilon = 8$), we see the result of balancing privacy and task performance at the expense of multilingual compression.

---

Differential Privacy, Linguistic Fairness, and Training Data Influence

---

(a) POS Performance  
(b) POS Retrieval  
(c) POS CKA  
(d) POS IsoScore  
(e) POS RSA

(f) XNLI Performance  
(g) XNLI Retrieval  
(h) XNLI CKA  
(i) XNLI IsoScore  
(j) XNLI RSA
While privacy, fairness, and interpretability individually have enjoyed ample attention from the research community in recent years [Liu et al., 2021a; Mehrabi et al., 2021; Søgaard, 2021], the interactions between these objectives have not been explored much [Ruder et al., 2022]. Some prior work has focused on the interactions between group fairness and differential privacy, suggesting that the two objectives are at odds, although this relationship also depends on the selected notion of fairness [Bagdasaryan et al., 2019; Cummings et al., 2019; Chang & Shokri, 2021; Hansen et al., 2022]. Somewhat in contrast to this work, we show that linguistic fairness (group fairness over linguistic communities) and differential privacy may align for multilingual language models. Furthermore, [Naidu et al., 2021] and [Shokri et al., 2021] have studied the interaction between privacy and feature attribution methods for model explainability. While the former show that privacy and feature attribution methods can align, the latter find that model explanations are at risk of membership inference attacks. Closest to our work is contemporaneous work by [Strobel & Shokri, 2022] who discuss the interactions of data privacy with fairness, explainability, and robustness. Our work differs from theirs in that we are particularly concerned with multilingual language models and we consider instance-based interpretability methods while they consider feature attribution methods. [Strobel & Shokri, 2022] also call for more research at the intersection of different objectives rather than working on one at a time.

7. Conclusion

We presented a preliminary investigation of how multilingual compression, differential privacy, training data influence, and linguistic fairness interact in multilingual models. We found that privacy and influence are incompatible, while privacy and linguistic fairness, often said to be at odds, are theoretically compatible through multilingual compression. We also explored these interactions empirically. Our results support the idea that high multilingual compression can be achieved either while optimizing for performance or while optimizing for privacy, but that by trading off privacy and performance, we compromise compression. Finding practical trade-offs between all these di-
Differential Privacy, Linguistic Fairness, and Training Data Influence

dimensions remains an open challenge. Finally, we introduced a new diagnostic metric, influence uniformity, which we used to validate that privacy and training data influence sparsity are incompatible, and that the interactions between privacy, training data influence sparsity, and multilingual compression are, therefore, also non-linear.

Ethical Aspects and Broader Impact

It is crucial that NLP goes beyond performance and studies the interaction of objectives such as privacy, interpretability, and fairness, also in multilingual NLP (Ruder et al. 2022). Our work aims to provide a starting point for further research in this area. Our empirical investigation, including the models we train, fully relies on publicly available models and data. Moreover, we do not create any new datasets. Therefore, we foresee no misuse of the results of our work.

Acknowledgements

We thank the anonymous reviewers and members of the CoAStaL group for their helpful feedback and suggestions. Phillip Rust is funded by the Novo Nordisk Foundation (grant NNF 20SA0066568).

References


Kerrigan, G., Slack, D., and Tulya, J. Differentially private language models benefit from public pre-training.
Differential Privacy, Linguistic Fairness, and Training Data Influence


Linguistics. URL https://aclanthology.org/2021.emnlp-demo.21


Muller, B., Elazar, Y., Sagot, B., and Seddah, D. First align, then predict: Understanding the cross-lingual ability of multilingual BERT. In Proceedings of the 16th Conference of the European Chapter of the Association
Nivre, J., de Marneffe, M.-C., Ginter, F., Hajiˇc, J., Man-
Naidu, R., Priyanshu, A., Kumar, A., Kotti, S., Wang, H.,
and Miresghallah, F. When differential privacy meets
interpretability: A case study. In CVPR 2021 Workshop
for Responsible Computer Vision (RCV), 2021. URL
https://arxiv.org/abs/2106.13203

Nivre, J., de Marneffe, M.-C., Ginter, F., Hajiˇc, J., Man-
ning, C. D., Pyysalo, S., Schuster, S., Tyers, F., and
Zeman, D. Universal Dependencies v2: An evergrow-
ing multilingual treebank collection. In Proceedings of
the 12th Language Resources and Evaluation Conference,
pp. 4034–4043, Marseille, France, May 2020. European
Language Resources Association. ISBN 979-10-
95546-34-4. URL
https://aclanthology.org/L20-1048

Pannekoek, M. and Spigler, G. Investigating trade-offs
in utility, fairness and differential privacy in neural
https://arxiv.org/abs/2102.05975

Papernot, N. and Steinke, T. Hyperparameter tuning
with renyi differential privacy. In Proceedings of the
10th International Conference on Learning Representa-
tions (ICLR), Online, 2022. OpenReview.net. URL
https://openreview.net/forum?id=-
70L8lpp9DF

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury,
J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N.,
Antiga, L., Desmaison, A., Köpf, A., Yang, E. Z.,
DeVito, Z., Raison, M., Tejani, A., Chilamkurthy,
S., Steiner, B., Fang, L., Bai, J., and Chintala, S.
Pytorch: An imperative style, high-performance deep
learning library. In Wallach, H. M., Larochelle, H.,
Beygelzimer, A., d’Alché-Buc, F., Fox, E. B., and
Garnett, R. (eds.), Advances in Neural Information
Processing Systems 32: Annual Conference on Neu-
8024–8035, Vancouver, BC, Canada, 2019. Curran
Associates, Inc. URL
https://proceedings.neurips.
cc/paper/2019/hash/e6385d39ec9394f2f3a354d9d2b88eecc-
Abstract.html

Phang, J., Liu, H., and Bowman, S. R. Fine-tuned trans-
fomers show clusters of similar representations across
layers. In Proceedings of the Fourth BlackboxNLP Work-
shop on Analyzing and Interpreting Neural Networks
for NLP, pp. 529–538, Punta Cana, Dominican Re-
public, November 2021. Association for Computational
Linguistics. URL
https://aclanthology.org/
2021.blackboxnlp-1.42

Pires, T., Schlinger, E., and Garrette, D. How multili-
gual is multilingual BERT? In Proceedings of the
57th Annual Meeting of the Association for Computa-
tional Linguistics, pp. 4996–5001, Florence, Italy, July
2019. Association for Computational Linguistics. URL
https://aclanthology.org/P19-1493

Pruksachatkun, Y., Ramakrishna, A., Chang, K.-W., Kr-
ishna, S., Dhamala, J., Guha, T., and Ren, X. (eds.),
Proceedings of the First Workshop on Trustworthy Nat-
ural Language Processing, Online, June 2021. Asso-
ciation for Computational Linguistics. URL
https://aclanthology.org/2021.trustnlp-1.0

Pruthi, G., Liu, F., Kale, S., and Sundararajan, M. Es-
timating training data influence by tracing gradient
descent. In Larochelle, H., Ranzato, M., Hadsell,
R., Balcan, M., and Lin, H. (eds.), Advances in Neu-
ral Information Processing Systems 33: An-
ual Conference on Neural Information Processing
Systems (NeurIPS), Online, 2020. Curran Asso-
ciates, Inc. URL
https://proceedings.
neurips.cc/paper/2020/hash/
e6385d39ec9394f2f3a354d9d2b88eecc-
Abstract.html

Raju, S. and Pilehvar, M. T. A cluster-based approach
for improving isotropy in contextual embedding space.
In Proceedings of the 59th Annual Meeting of the Asso-
ciation for Computational Linguistics and the 11th In-
nernational Joint Conference on Natural Language
Processing (Volume 2: Short Papers), pp. 575–584,
Online, August 2021. Association for Computational
Linguistics. URL
acl-short.73

Ravishankar, V. and Søgaard, A. The impact of positional
encodings on multilingual compression. In Proceed-
ses of the 2021 Conference on Empirical Methods in
Natural Language Processing, pp. 763–777, Online and Punta
Cana, Dominican Republic, November 2021. Associa-
tion for Computational Linguistics. URL
acl-short.59

Reimers, N. and Gurevych, I. Making monolingual sen-
tence embeddings multilingual using knowledge distilla-
tion. In Proceedings of the 2020 Conference on Empiri-
ical Methods in Natural Language Processing (EMNLP),
pp. 4512–4525, Online, November 2020. Associa-
tion for Computational Linguistics. URL
acl-short.365

Ruder, S., Vučić, I., and Søgaard, A. Square one bias in
NLP: Towards a multi-dimensional exploration of the
Differential Privacy, Linguistic Fairness, and Training Data Influence


Tramèr, F. and Boneh, D. Differentially private learning needs better features (or much more data). In *Proceedings of the 9th International Conference on Learning Representations (ICLR)*, Online, 2021. OpenReview.net. URL [https://openreview.net/forum?id=YTWGvpFQqD](https://openreview.net/forum?id=YTWGvpFQqD)


Wu, S. and Dredze, M. Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT. In *Proceedings


Differential Privacy, Linguistic Fairness, and Training Data Influence


A. Reproducibility
We make our code available at https://github.com/xplip/multilingual-lm-objectives.

Implementation  Our implementation is written in PyTorch v1.10.0 (Paszke et al., 2019) for Python 3.9.5 and builds on code from the following repositories:

- https://github.com/huggingface/transformers v4.9.2 (Wolf et al., 2020) for model training and evaluation
- https://github.com/lxuechen/private-transformers v0.1.0 (Li et al., 2022) for DP-training
- https://github.com/pdufter/minimult (Dufter & Schütze, 2020) for computing sentence retrieval precision
- https://github.com/jayroxis/CKA-similarity for computing CKA scores
- https://github.com/lmlepori/Picking_BERTs_Brain (Lepori & McCoy, 2020) for computing RSA scores
- https://github.com/bcbi-edu/p_eickhoff_isoscore (Rudman et al., 2022) for computing IsoScores
- https://github.com/FengNiMa/VAE-TracIn-pytorch (Kong & Chaudhuri, 2021) for computing TracInCP scores.

Models  We primarily use the pretrained XLM-RoBERTa (XLM-R; Conneau et al., 2020a) base model and tokenizer from https://huggingface.co/xlm-roberta-base. XLM-R (base) is a 12-layer encoder-only transformer with a vocabulary size of 250k and ~277M total parameters pretrained via masked language modeling on the 100-language CC-100 dataset.

In Appendix [7] we further conduct experiments with multilingual BERT (mBERT; Devlin et al., 2019), using the base model and tokenizer from https://huggingface.co/bert-base-multilingual-cased. mBERT is a 12-layer encoder-only transformer with a vocabulary size of 120k and ~177M total parameters pretrained via masked language modeling on Wikipedia data in 104 languages.

Data  We provide download links and references for the various datasets we used in Table 3.

Hardware  We train on single Nvidia Titan RTX, A100 (both with CUDA version 11.0), and RTX 3090 (with CUDA version 11.5) GPUs. All machines have at least 64GB of RAM, which is required to compute the IsoScore for our larger evaluation sets (e.g., TED 2020 for POS).

Runtime  Fine-tuning with evaluation during training on the Titan RTX, which is the slowest of the GPUs used, takes 2–3 hours for POS and 5–6 hours for XNLI. Computing TracInCP influence scores for one fine-tuned model takes about 30–45 minutes.

Carbon Footprint  Our fine-tuning runs accumulated ~36 compute days on the hardware mentioned above (most experiments were conducted on the less powerful Titan RTX GPUs) according to Weights & Biases (https://wandb.ai/), where we logged our experiments. Although we do not have precise numbers, a highly conservative estimate of the total compute spent including prototyping, hyper-parameter search, and all our evaluations is ~75 compute days.

B. (ε, δ)-Differential Privacy
In §2 we provide the definition of ε-differential privacy (DP), also called pure DP, as the basis for our theoretical exploration. In our experiments, we rely on (ε, δ)-DP (Dwork & Roth, 2014), also called approximate-DP, which is typically used in practice and relaxes the privacy guarantees by a (small) δ as follows:

A randomized algorithm \( M : \mathcal{D} \rightarrow \mathcal{Y} \) is \((\varepsilon, \delta)\)-differentially private (Dwork, 2006) iff for all adjacent datasets \( D, D' \in \mathcal{D} \) and all \( Y \subset \mathcal{Y} \), \( \mathbb{P}(M(D) \in Y) \leq \exp(\varepsilon) \cdot \mathbb{P}(M(D') \in Y) + \delta \).
C. Best Fine-Tuning Settings

As mentioned in §3, we pre-selected a set of suitable learning rates (LRs) for each task and ran 3 random initializations each. Based on the validation performance, we then selected the following 5 best settings for each privacy budget and task:

Table 1: Best 5 settings for each task and privacy budget. Includes LR and the corresponding number of random initializations (# seeds).

<table>
<thead>
<tr>
<th>ε</th>
<th>POS LR (# Seeds)</th>
<th>XNLI LR (# Seeds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5e−4 (2); 7e−4 (3)</td>
<td>3e−4 (1); 4e−4 (2); 5e−4 (2)</td>
</tr>
<tr>
<td>3</td>
<td>5e−4 (2); 7e−4 (3)</td>
<td>3e−4 (1); 4e−4 (2); 5e−4 (2)</td>
</tr>
<tr>
<td>8</td>
<td>5e−4 (3); 7e−4 (2)</td>
<td>4e−4 (2); 5e−4 (3)</td>
</tr>
<tr>
<td>15</td>
<td>3e−4 (1); 5e−4 (2); 7e−4 (2)</td>
<td>3e−4 (1); 4e−4 (2); 5e−4 (2)</td>
</tr>
<tr>
<td>30</td>
<td>3e−4 (1); 5e−4 (2); 7e−4 (2)</td>
<td>3e−4 (1); 4e−4 (2); 5e−4 (2)</td>
</tr>
<tr>
<td>∞</td>
<td>5e−5 (2); 7e−5 (2); 1e−4 (1)</td>
<td>9e−5 (2); 1e−4 (3)</td>
</tr>
</tbody>
</table>

D. IsoScore Algorithm

Algorithm describes the IsoScore algorithm (Rudman et al., 2022).

Algorithm 1 IsoScore

1: begin
2: Let $X \subset \mathbb{R}^n$ be a finite collection of points.
3: Let $X^{PCA}$ denote the points in $X$ transformed by the first $n$ principal components.
4: Define $\Sigma_D \in \mathbb{R}^n$ as the diagonal of the covariance matrix of $X^{PCA}$.
5: Normalize diagonal to $\hat{\Sigma} := \sqrt{n} \cdot \Sigma_D / \| \Sigma_D \|$, where $\| \cdot \|$ is the standard Euclidean norm.
6: The isotropy defect is $\delta(X) := \| \hat{\Sigma} - I \| / \sqrt{2(n - \sqrt{n})}$, where $I = (1, \ldots, 1)^T \in \mathbb{R}^n$.
7: $X$ uniformly occupies $\phi(X) := (n - \delta(X))^2(n - \sqrt{n})^2 / n^2$ percent of ambient dimensions.
8: Transform $\phi(X)$ so it can take values in $[0, 1]$, via $\iota(X) := (n \cdot \phi(X) - 1) / (n - 1)$.
9: return $\iota(X)$
10: end

E. Further Analysis of RSA Results

As we see in §3, RSA aligns with sentence retrieval precision, CKA, and IsoScore in producing higher scores for non-private models. However, there is a mismatch between RSA and the other metrics in highly private regimes, where our most private models (ε = 1) do not exhibit high RSA scores. Instead, the aggregated RSA scores peak at medium levels of privacy (ε ∈ {8, 15}) and for the non-private (ε = ∞) models. Unlike for the other metrics, there is also no clear trend among our two tasks in terms of whether the pretrained or a randomly initialized XLM-R model scores higher in RSA.

A closer look at the non-aggregated results (Appendix Figures [10], [11] and [14]) shows how the similarity patterns obtained from RSA are often unexpected. For instance, the similarities between the typologically distant languages FR and ZH are consistently high for the TED 2020 corpus whereas scores for typologically closer languages are lower (Fig. [10]). Based on prior work by, for example, (Pires et al., 2019), (Wu & Dredze, 2019), and (Lauscher et al., 2020), we would expect the model to first compress similar languages before achieving compression for distant ones. Sometimes, we also observe extreme jumps in similarity between layers 0 and 8, for instance, between FR and TR in the Tatoeba corpus (Fig. [11]). We do not find these jumps in CKA and sentence retrieval.

One reason why RSA scores may be more sensitive to stricter privacy guarantees (e.g., ε = 1) is that the correlation between sentence vector distances is very sensitive to outliers. Differential privacy reduces the number of such outliers, effectively regularizing the correlation coefficients.
F. Multilingual BERT Results

In Figures 3 and 4, we present results from re-running the experiments from §4 and §5 with mBERT. We make two changes to the experimental setup outlined above: We use representations extracted at layer 8, which showed to be more meaningful than layer 0 in the XLM-R experiments, to compute the multilinguality metrics. We also include two additional privacy settings, $\varepsilon = 0.5$ and $\varepsilon = 0.7$, as we found mBERT to be easier to finetune with strong privacy guarantees than XLM-R.

We see the same trends as for XLM-R: performance strictly increases with decreasing privacy while the multilinguality metrics tend to follow a U-shape, i.e., they are high for strong privacy settings (small $\varepsilon$) and low privacy settings (large $\varepsilon$) and decrease towards medium privacy. Likewise, we find a positive correlation between InfU and cross-lingual sentence retrieval precision. The correlation is strong for part-of-speech tagging (POS) but it is mild for XNLI. We believe this may be due to mBERT being less sensitive to the privacy parameter (Figure 3g is not symmetrical; considering even stronger privacy settings would likely even out the U-shape). Overall, these results further support our finding that there is a negative correlation between multilingual compression and training data influence sparsity.

We again refer to Appendix E for a discussion of the RSA results.

33We again refer to Appendix E for a discussion of the RSA results.
G. Detailed Results for Experiments in §4

Figure 5 shows the development of the mean sentence retrieval precision at layer 8 for POS and XNLI over the course of fine-tuning with different privacy budgets.

We further present non-aggregated results for

- POS performance in Table 5
- XNLI performance in Table 6
- Sentence retrieval for POS in Figures 6 and 7
- Sentence retrieval for XNLI in Figure 12
- CKA for POS in Figures 8 and 9
- CKA for XNLI in Figure 13
- IsoScore for POS in Table 7
- IsoScore for XNLI in Table 8
- RSA for POS in Figures 10 and 11
- RSA for XNLI in Figure 14.

Table 2: Overview of languages used in our experiments. Tokens (in millions) and size (in Gibibytes) refer to the respective monolingual corpora in XLM-R’s pretraining corpus. Numbers taken from (Conneau et al., 2020a). *: includes romanized variants also used in pretraining.

<table>
<thead>
<tr>
<th>Language</th>
<th>ISO</th>
<th>Family</th>
<th>Script</th>
<th>Tokens (M)</th>
<th>Size (GiB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arabic</td>
<td>AR</td>
<td>Afro-Asiatic</td>
<td>Arabic</td>
<td>2869</td>
<td>28.0</td>
</tr>
<tr>
<td>Bulgarian</td>
<td>BG</td>
<td>Indo-European</td>
<td>Cyrillic</td>
<td>5487</td>
<td>57.5</td>
</tr>
<tr>
<td>Chinese</td>
<td>ZH</td>
<td>Sino-Tibetan</td>
<td>Chinese</td>
<td>435</td>
<td>63.5</td>
</tr>
<tr>
<td>French</td>
<td>FR</td>
<td>Indo-European</td>
<td>Latin</td>
<td>9780</td>
<td>56.8</td>
</tr>
<tr>
<td>German</td>
<td>DE</td>
<td>Indo-European</td>
<td>Latin</td>
<td>10297</td>
<td>66.6</td>
</tr>
<tr>
<td>Greek</td>
<td>EL</td>
<td>Indo-European</td>
<td>Greek</td>
<td>4285</td>
<td>46.9</td>
</tr>
<tr>
<td>Hindi</td>
<td>HI</td>
<td>Indo-European</td>
<td>Devanagari</td>
<td>1803*</td>
<td>20.7*</td>
</tr>
<tr>
<td>Indonesian</td>
<td>ID</td>
<td>Austronesian</td>
<td>Latin</td>
<td>22704</td>
<td>148.3</td>
</tr>
<tr>
<td>Italian</td>
<td>IT</td>
<td>Indo-European</td>
<td>Latin</td>
<td>4983</td>
<td>30.2</td>
</tr>
<tr>
<td>Japanese</td>
<td>JA</td>
<td>Japonic</td>
<td>Japanese</td>
<td>530</td>
<td>69.3</td>
</tr>
<tr>
<td>Kiswahili</td>
<td>SW</td>
<td>Niger-Congo</td>
<td>Latin</td>
<td>275</td>
<td>1.6</td>
</tr>
<tr>
<td>Korean</td>
<td>KO</td>
<td>Koreanic</td>
<td>Korean</td>
<td>5644</td>
<td>54.2</td>
</tr>
<tr>
<td>Portuguese</td>
<td>PT</td>
<td>Indo-European</td>
<td>Latin</td>
<td>8405</td>
<td>49.1</td>
</tr>
<tr>
<td>Russian</td>
<td>RU</td>
<td>Indo-European</td>
<td>Cyrillic</td>
<td>23408</td>
<td>278.0</td>
</tr>
<tr>
<td>Thai</td>
<td>TH</td>
<td>Kra-Dai</td>
<td>Thai</td>
<td>1834</td>
<td>71.7</td>
</tr>
<tr>
<td>Turkish</td>
<td>TR</td>
<td>Turkic</td>
<td>Latin</td>
<td>2736</td>
<td>20.9</td>
</tr>
<tr>
<td>Urdu</td>
<td>UR</td>
<td>Indo-European</td>
<td>Arabic</td>
<td>815*</td>
<td>6.2*</td>
</tr>
<tr>
<td>Vietnamese</td>
<td>VI</td>
<td>Austro-Asiatic</td>
<td>Latin</td>
<td>24757</td>
<td>137.3</td>
</tr>
</tbody>
</table>

Table 3: Links and references to the datasets we used in our experiments. License information are also available via these links. We ensure that we comply with respective license conditions and only use the data within their intended use policy where applicable.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Download Link</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>UD v2.8 (POS)</td>
<td><a href="https://lindat.mff.cuni.cz/repository/xmlui/handle/11234/1-3683">https://lindat.mff.cuni.cz/repository/xmlui/handle/11234/1-3683</a></td>
<td>Nivre et al., 2020; Zeman et al., 2021</td>
</tr>
<tr>
<td>XNLI</td>
<td><a href="https://huggingface.co/datasets/xnli">https://huggingface.co/datasets/xnli</a></td>
<td>Conneau et al., 2018; Lhoest et al., 2021</td>
</tr>
<tr>
<td>Tatoeba</td>
<td><a href="https://github.com/LBeaudoux/tatoebatools">https://github.com/LBeaudoux/tatoebatools</a></td>
<td></td>
</tr>
</tbody>
</table>
Table 4: Overview of the UD v2.8 [Nivre et al., 2020; Zeman et al., 2021] treebanks (test splits only) that we use as test sets in our POS tagging experiments (§3,4) including their respective sizes (number of sentences).

<table>
<thead>
<tr>
<th>Language</th>
<th>Treebank</th>
<th># Sentences</th>
</tr>
</thead>
<tbody>
<tr>
<td>AR</td>
<td>Arabic-PADT</td>
<td>680</td>
</tr>
<tr>
<td>DE</td>
<td>German-GSD</td>
<td>977</td>
</tr>
<tr>
<td>ES</td>
<td>Spanish-GSD</td>
<td>426</td>
</tr>
<tr>
<td>HI</td>
<td>Hindi-HDTB</td>
<td>1684</td>
</tr>
<tr>
<td>ID</td>
<td>Indonesian-GSD</td>
<td>557</td>
</tr>
<tr>
<td>KO</td>
<td>Korean-Kaist</td>
<td>2287</td>
</tr>
<tr>
<td>RU</td>
<td>Russian-SynTagRus</td>
<td>6491</td>
</tr>
</tbody>
</table>

Figure 5: Mean sentence retrieval precision for our TED 2020 splits (different languages/data for POS and XNLI) at layer 8 over the course of fine-tuning with different privacy budgets ($\epsilon$). $\epsilon = \infty$ denotes non-private models. Error bands show variation around the mean over 5 random seeds. At Steps = 0, all models are equivalent to the pretrained XLM-R Base. We see that the non-private models can retain (and for XNLI even improve) their multilingual compression much better than the private models and have less variation.
Table 5: POS Performance (validation / test accuracy) when fine-tuning XLM-R Base with different privacy budgets ($\varepsilon$). We show results averaged over 5 random seeds each. $\varepsilon = \infty$ denotes non-private models. AVG is the average over the 7 languages. See §3 for our experimental setup. We see that performance increases with decreased privacy across all languages.

<table>
<thead>
<tr>
<th>$\varepsilon$</th>
<th>AR</th>
<th>DE</th>
<th>ES</th>
<th>HI</th>
<th>ID</th>
<th>KO</th>
<th>RU</th>
<th>AVG</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>68.3 / 64.6</td>
<td>75.5 / 75.1</td>
<td>79.8 / 79.0</td>
<td>65.0 / 63.3</td>
<td>73.8 / 71.9</td>
<td>66.1 / 54.2</td>
<td>74.8 / 74.0</td>
<td>71.9 / 68.9</td>
</tr>
<tr>
<td>3</td>
<td>79.1 / 76.6</td>
<td>86.6 / 86.8</td>
<td>90.3 / 89.3</td>
<td>74.4 / 70.9</td>
<td>82.6 / 79.4</td>
<td>71.1 / 59.4</td>
<td>86.1 / 86.3</td>
<td>81.4 / 78.4</td>
</tr>
<tr>
<td>8</td>
<td>81.0 / 77.6</td>
<td>88.4 / 88.3</td>
<td>91.6 / 90.2</td>
<td>78.2 / 75.6</td>
<td>84.2 / 81.2</td>
<td>70.8 / 60.9</td>
<td>87.1 / 87.4</td>
<td>83.0 / 80.2</td>
</tr>
<tr>
<td>15</td>
<td>81.3 / 78.4</td>
<td>88.8 / 89.0</td>
<td>92.4 / 90.9</td>
<td>77.0 / 73.2</td>
<td>83.9 / 80.7</td>
<td>71.9 / 61.8</td>
<td>87.7 / 87.8</td>
<td>83.3 / 80.3</td>
</tr>
<tr>
<td>30</td>
<td>81.8 / 78.7</td>
<td>89.4 / 89.6</td>
<td>92.9 / 91.5</td>
<td>77.6 / 74.0</td>
<td>84.3 / 81.1</td>
<td>72.3 / 62.2</td>
<td>88.2 / 88.4</td>
<td>83.8 / 80.8</td>
</tr>
<tr>
<td>$\infty$</td>
<td>83.8 / 79.7</td>
<td>91.5 / 91.2</td>
<td>95.0 / 93.2</td>
<td>82.8 / 80.2</td>
<td>86.2 / 81.3</td>
<td>74.2 / 62.9</td>
<td>89.9 / 90.2</td>
<td>86.2 / 82.7</td>
</tr>
</tbody>
</table>

Table 6: XNLI Performance (validation / test accuracy) when fine-tuning XLM-R Base with different privacy budgets ($\varepsilon$). We show results averaged over 5 random seeds each. $\varepsilon = \infty$ denotes non-private models. AVG is the average over the 7 languages. See §3 for our experimental setup. We see that performance increases with decreased privacy across all languages. Here, we also particularly observe that the gap between validation and test performance is substantially lower for private models, which shows the strong regularization effect of training with differential privacy.

<table>
<thead>
<tr>
<th>$\varepsilon$</th>
<th>AR</th>
<th>DE</th>
<th>EL</th>
<th>RU</th>
<th>SW</th>
<th>TH</th>
<th>UR</th>
<th>AVG</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>37.3 / 37.4</td>
<td>36.8 / 37.0</td>
<td>36.6 / 36.5</td>
<td>36.3 / 36.2</td>
<td>34.3 / 34.5</td>
<td>35.6 / 35.7</td>
<td>35.6 / 35.6</td>
<td>36.1 / 36.1</td>
</tr>
<tr>
<td>3</td>
<td>49.6 / 50.3</td>
<td>49.3 / 51.0</td>
<td>50.8 / 51.5</td>
<td>49.7 / 50.2</td>
<td>45.9 / 47.2</td>
<td>48.8 / 49.5</td>
<td>47.6 / 48.2</td>
<td>48.8 / 49.7</td>
</tr>
<tr>
<td>8</td>
<td>55.9 / 56.4</td>
<td>56.8 / 58.5</td>
<td>58.2 / 58.1</td>
<td>56.3 / 57.1</td>
<td>52.0 / 53.2</td>
<td>55.6 / 55.7</td>
<td>53.3 / 53.7</td>
<td>55.5 / 56.1</td>
</tr>
<tr>
<td>15</td>
<td>59.1 / 58.3</td>
<td>60.4 / 60.8</td>
<td>61.5 / 60.9</td>
<td>59.7 / 59.5</td>
<td>54.4 / 54.8</td>
<td>58.9 / 58.2</td>
<td>56.4 / 56.1</td>
<td>58.6 / 58.4</td>
</tr>
<tr>
<td>30</td>
<td>61.6 / 60.8</td>
<td>63.6 / 63.1</td>
<td>64.8 / 62.0</td>
<td>62.0 / 61.1</td>
<td>56.5 / 57.3</td>
<td>61.2 / 60.2</td>
<td>58.6 / 57.8</td>
<td>61.2 / 60.3</td>
</tr>
<tr>
<td>$\infty$</td>
<td>90.9 / 67.8</td>
<td>96.2 / 70.5</td>
<td>95.5 / 70.1</td>
<td>93.4 / 69.7</td>
<td>79.0 / 62.5</td>
<td>91.6 / 68.5</td>
<td>86.8 / 65.4</td>
<td>90.5 / 67.8</td>
</tr>
</tbody>
</table>
Figure 6: POS Sentence retrieval results for the TED 2020 (TED) and WikiMatrix (WM) datasets and different combinations of privacy budgets ($\varepsilon$) and layers ($l$). Each heatmap cell corresponds to the average over 5 random seeds. We observe that the overall patterns are highly similar across all levels of privacy, particularly at layer 0.
Figure 7: **POS** sentence retrieval results for the Tatoeba dataset and different combinations of privacy budgets ($\varepsilon$) and layers ($l$). Each heatmap cell corresponds to the average over 5 random seeds. We observe that the overall patterns are highly similar across all levels of privacy, particularly at layer 0.
Figure 8: POS CKA results for the TED 2020 (TED) and WikiMatrix (WM) datasets and different combinations of privacy budgets ($\epsilon$) and layers ($l$). Each heatmap cell corresponds to the average over 5 random seeds. We observe that the overall patterns are highly similar across all levels of privacy, particularly at layer 0.
Figure 9: **POS CKA results for the Tatoeba dataset and different combinations of privacy budgets ($\varepsilon$) and layers ($l$).** Each heatmap cell corresponds to the average over 5 random seeds. We observe that the overall patterns are highly similar across all levels of privacy, particularly at layer 0.
Figure 10: POS RSA results for the TED 2020 (TED) and WikiMatrix (WM) datasets and different combinations of privacy budgets ($\varepsilon$) and layers ($l$). Each heatmap cell corresponds to the average over 5 random seeds. We observe that the overall patterns are highly similar across all levels of privacy, particularly at layer 0.
Figure 11: POS RSA results for the Tatoeba dataset and different combinations of privacy budgets ($\varepsilon$) and layers ($l$). Each heatmap cell corresponds to the average over 5 random seeds. We observe that the overall patterns are highly similar across all levels of privacy, particularly at layer 0. Also note that, unlike in CKA (Figure 9), the similarity between IT and TR is high at layer 0 but low at layer 8.

30
Figure 12: XNLI Sentence retrieval results for the TED 2020 (TED) and WikiMatrix (WM) datasets and different combinations of privacy budgets ($\varepsilon$) and layers ($l$). Each heatmap cell corresponds to the average over 5 random seeds. We observe that the overall patterns are highly similar across all levels of privacy, particularly at layer 0.
Figure 13: XNLI CKA results for the TED 2020 (TED) and WikiMatrix (WM) datasets and different combinations of privacy budgets ($\epsilon$) and layers ($l$). Each heatmap cell corresponds to the average over 5 random seeds. We observe that the overall patterns are highly similar across all levels of privacy, particularly at layer 0.
Figure 14: XNLI RSA results for the TED 2020 (TED) and WikiMatrix (WM) datasets and different combinations of privacy budgets ($\varepsilon$) and layers ($l$). Each heatmap cell corresponds to the average over 5 random seeds. We observe that the overall patterns are highly similar across all levels of privacy, particularly at layer 0.
Table 7: **POS IsoScores** for different combinations of privacy budgets ($\varepsilon$) and layers ($l$). We show results averaged over 5 random seeds, except for RND and PRE. RND and PRE (added for comparison) denote XLM-R with randomly initialized weights and the original pretrained XLM-R, respectively. We see that the isotropy is fairly uniform across privacy budgets at layer 0 and generally higher at layer 0 than at layer 8. At layer 8, it peaks for non-private ($\varepsilon = \infty$) and our most private ($\varepsilon = 1$) models.

<table>
<thead>
<tr>
<th>$\varepsilon$</th>
<th>TED 2020 $l = 0$</th>
<th>TED 2020 $l = 8$</th>
<th>WikiMatrix $l = 0$</th>
<th>WikiMatrix $l = 8$</th>
<th>Tatoeba $l = 0$</th>
<th>Tatoeba $l = 8$</th>
</tr>
</thead>
<tbody>
<tr>
<td>RND</td>
<td>0.141</td>
<td>0.132</td>
<td>0.114</td>
<td>0.111</td>
<td>0.054</td>
<td>0.061</td>
</tr>
<tr>
<td>PRE</td>
<td>0.187</td>
<td>0.130</td>
<td>0.198</td>
<td>0.112</td>
<td>0.134</td>
<td>0.075</td>
</tr>
<tr>
<td>1</td>
<td>0.188</td>
<td>0.054</td>
<td>0.199</td>
<td>0.046</td>
<td>0.135</td>
<td>0.033</td>
</tr>
<tr>
<td>3</td>
<td>0.188</td>
<td>0.044</td>
<td>0.199</td>
<td>0.038</td>
<td>0.135</td>
<td>0.027</td>
</tr>
<tr>
<td>8</td>
<td>0.187</td>
<td>0.045</td>
<td>0.197</td>
<td>0.038</td>
<td>0.133</td>
<td>0.027</td>
</tr>
<tr>
<td>15</td>
<td>0.187</td>
<td>0.047</td>
<td>0.199</td>
<td>0.040</td>
<td>0.135</td>
<td>0.028</td>
</tr>
<tr>
<td>30</td>
<td>0.187</td>
<td>0.047</td>
<td>0.199</td>
<td>0.040</td>
<td>0.135</td>
<td>0.028</td>
</tr>
<tr>
<td>$\infty$</td>
<td>0.188</td>
<td>0.087</td>
<td>0.199</td>
<td>0.070</td>
<td>0.135</td>
<td>0.051</td>
</tr>
</tbody>
</table>

Table 8: **XNLI IsoScores** for different combinations of privacy budgets ($\varepsilon$) and layers ($l$). We show results averaged over 5 random seeds, except for RND and PRE. RND and PRE (added for comparison) denote XLM-R with randomly initialized weights and the original pretrained XLM-R, respectively. We see that the isotropy is fairly uniform across privacy budgets at layer 0 and generally higher at layer 0 than at layer 8. At layer 8, it peaks for non-private ($\varepsilon = \infty$) and our most private ($\varepsilon = 1$) models.

<table>
<thead>
<tr>
<th>$\varepsilon$</th>
<th>TED 2020 $l = 0$</th>
<th>TED 2020 $l = 8$</th>
<th>WikiMatrix $l = 0$</th>
<th>WikiMatrix $l = 8$</th>
</tr>
</thead>
<tbody>
<tr>
<td>RND</td>
<td>0.144</td>
<td>0.134</td>
<td>0.130</td>
<td>0.124</td>
</tr>
<tr>
<td>PRE</td>
<td>0.195</td>
<td>0.138</td>
<td>0.210</td>
<td>0.129</td>
</tr>
<tr>
<td>1</td>
<td>0.195</td>
<td>0.121</td>
<td>0.211</td>
<td>0.120</td>
</tr>
<tr>
<td>3</td>
<td>0.196</td>
<td>0.101</td>
<td>0.211</td>
<td>0.104</td>
</tr>
<tr>
<td>8</td>
<td>0.196</td>
<td>0.074</td>
<td>0.212</td>
<td>0.079</td>
</tr>
<tr>
<td>15</td>
<td>0.196</td>
<td>0.071</td>
<td>0.212</td>
<td>0.077</td>
</tr>
<tr>
<td>30</td>
<td>0.194</td>
<td>0.087</td>
<td>0.210</td>
<td>0.089</td>
</tr>
<tr>
<td>$\infty$</td>
<td>0.195</td>
<td>0.182</td>
<td>0.211</td>
<td>0.166</td>
</tr>
</tbody>
</table>