Abstract

The dominant approaches to text representation in natural language rely on learning embeddings on massive corpora which have convenient properties such as compositionality and distance preservation. In this paper, we develop a novel method to learn a heavy-tailed embedding with desirable regularity properties regarding the distributional tails, which allows to analyze the points far away from the distribution bulk using the framework of multivariate extreme value theory. In particular, a classifier dedicated to the tails of the proposed embedding is obtained which exhibits a *scale invariance* property exploited in a novel text generation method for label preserving dataset augmentation. Experiments on synthetic and real text data show the relevance of the proposed framework and confirm that this method generates meaningful sentences with controllable attributes, e.g. positive or negative sentiments.

1 Introduction

Representing the meaning of natural language in a mathematically grounded way is a scientific challenge that has received increasing attention with the explosion of digital content and text data in the last decade. Relying on the richness of contents, several embeddings have been proposed \cite{44, 45, 19} with demonstrated efficiency for the considered tasks when learnt on massive datasets.

*Both authors contributed equally*
However, none of these embeddings take into account the fact that word frequency distributions are heavy tailed \cite{2,11,40}, so that extremes are naturally present in texts (see also Fig. \ref{fig:embeddings}a and \ref{fig:embeddings}b in the supplementary material). Similarly, \cite{3} shows that, contrary to image taxonomies, the underlying distributions for words and documents in large scale textual taxonomies are also heavy tailed. Exploiting this information, several studies, as \cite{13,38}, were able to improve text mining applications by accurately modeling the tails of textual elements.

In this work, we rely on the framework of multivariate extreme value analysis, based on extreme value theory (EVT) which focuses on the distributional tails. EVT is valid under a regularity assumption which amounts to a homogeneity property above large thresholds: the tail behavior of the considered variables must be well approximated by a power law, see Section \ref{sec:preliminaries} for a rigorous statement. The tail region (where samples are considered as extreme) of the input variable \(x \in \mathbb{R}^d\) is of the kind \(\{\|x\| \geq t\}\), for a large threshold \(t\). The latter is typically chosen such that a small but non negligible proportion of the data is considered as extreme, namely \(25\%\) in our experiments. A major advantage of this framework in the case of labeled data \cite{30} is that classification on the tail regions may be performed using the angle \(\Theta(x) = \|x\|^{-1}x\) only, see Figure \ref{fig:classifier}. The main idea behind the present paper is to take advantage of the scale invariance for two tasks regarding sentiment analysis of text data: (i) Improved classification of extreme inputs, (ii) Label preserving data augmentation, as the most probable label of an input \(x\) is unchanged by multiplying \(x\) by \(\lambda > 1\).

EVT in a machine learning framework has received increasing attention in the past few years. Learning tasks considered so far include anomaly detection \cite{48,49,12,23,53}, anomaly clustering \cite{9}, unsupervised learning \cite{22}, online learning \cite{6,1}, dimension reduction and support identification \cite{24,8,10,29}. The present paper builds upon the methodological framework proposed by Jalalzai et al. \cite{30} for classification in extreme regions. The goal of Jalalzai et al. \cite{30} is to improve the performance of classifiers \(\hat{g}(x)\) issued from Empirical Risk Minimization (ERM) on the tail regions \(\{\|x\| > t\}\). Indeed, they argue that for very large \(t\), there is no guarantee that \(\hat{g}\) would perform well conditionally to \(\{\|X\| > t\}\), precisely because of the scarcity of such examples in the training set. They thus propose to train a specific classifier dedicated to extremes leveraging the probabilistic structure of the tails. Jalalzai et al. \cite{30} demonstrate the usefulness of their framework with simulated and some real world datasets. However, there is no reason to assume that the previously mentioned text embeddings satisfy the required regularity assumptions. The aim of the present work is to extend \cite{30}'s methodology to datasets which do not satisfy their assumptions, in particular to text datasets embedded by state of the art techniques. This is achieved by the algorithm \textit{Learning a Heavy Tailed Representation} (in short \textbf{LHTR}) which learns a transformation mapping the input data \(X\) onto a random vector \(Z\) which does satisfy the aforementioned assumptions. The transformation is learnt by an adversarial strategy \cite{26}.

In Appendix \ref{sec:appen} we propose an interpretation of the extreme nature of an input in both \textbf{LHTR} and BERT representations. In a word, these sequences are longer and are more difficult to handle (for next token prediction and classification tasks) than non extreme ones.

Our second contribution is a novel data augmentation mechanism \textbf{GENELIEX} which takes advantage of the scale invariance properties of \(Z\) to generate synthetic sequences that keep invariant the attribute of the original sequence. Label preserving data augmentation is an effective solution to the data scarcity problem and is an efficient pre-processing step for moderate dimensional datasets \cite{55,56}. Adapting these methods to NLP problems remains a challenging issue. The problem consists in constructing a transformation \(h\) such that for any sample \(x\) with label \(y(x)\), the generated sample \(h(x)\) would remain label consistent: \(y(h(x)) = y(x)\) \cite{46}. The dominant approaches for text data augmentation rely on word level transformations such as synonym replacement, slot filling, swap deletion \cite{56} using external resources such as wordnet \cite{42}. Linguistic based approaches can also be
combined with vectorial representations provided by language models [32]. However, to the best of our knowledge, building a vectorial transformation without using any external linguistic resources remains an open problem. In this work, as the label \( y(h(x)) \) is unknown as soon as \( h(x) \) does not belong to the training set, we address this issue by learning both an embedding \( \varphi \) and a classifier \( g \) satisfying a relaxed version of the problem above mentioned, namely \( \forall \lambda \geq 1 \)

\[
g(h_\lambda(\varphi(x))) = g(\varphi(x)). \tag{1}
\]

For mathematical reasons which will appear clearly in Section 2.2, \( h_\lambda \) is chosen as the homothety with scale factor \( \lambda \), \( h_\lambda(x) = \lambda x \). In this paper, we work with output vectors issued by BERT [19]. BERT and its variants are currently the most widely used language model but we emphasize that the proposed methodology could equally be applied using any other representation as input. BERT embedding does not satisfy the regularity properties required by EVT (see the results from statistical tests performed in Appendix B.5). Besides, there is no reason why a classifier \( g \) trained on such embedding would be scale invariant, \( i.e., \) would satisfy for a given sequence \( u \), embedded as \( x \), \( g(h_\lambda(x)) = g(x) \forall \lambda \geq 1 \). On the classification task, we demonstrate on two datasets of sentiment analysis that the embedding learnt by LHTR on top of BERT is indeed following a heavy-tailed distribution. Besides, a classifier trained on the embedding learnt by LHTR outperforms the same classifier trained on BERT. On the dataset augmentation task, quantitative and qualitative experiments demonstrate the ability of GENELIEX to generate new sequences while preserving labels.

The rest of this paper is organized as follows. Section 2 introduces the necessary background in multivariate extremes. The methodology we propose is detailed at length in Section 3. Illustrative numerical experiments on both synthetic and real data are gathered in Sections 4 and 5. Further comments and experimental results are provided in the supplementary material.

## 2 Background

### 2.1 Extreme values, heavy tails and regular variation

Extreme value analysis is a branch of statistics whose main focus is on events characterized by an unusually high value of a monitored quantity. A convenient working assumption in EVT is regular variation. A real-valued random variable \( X \) is regularly varying with index \( \alpha > 0 \), a property denoted as \( RV(\alpha) \), if and only if there exists a function \( b(t) > 0 \), with \( b(t) \to \infty \) as \( t \to \infty \), such that for any fixed \( x > 0 \):

\[
tP\{X/b(t) > x\} \xrightarrow{t \to \infty} x^{-\alpha}. \tag{2}
\]

In the multivariate case \( X = (X_1, \ldots, X_d) \in \mathbb{R}^d \), it is usually assumed that a preliminary component-wise transformation has been applied so that each margin \( X_j \) is \( RV(1) \) with \( b(t) = t \) and takes only positive values. \( X \) is standard multivariate regularly varying if there exists a positive Radon measure \( \mu \) on \([0, \infty[^d \setminus \{0\}\]

\[
tP\{t^{-1}X \in A\} \xrightarrow{t \to \infty} \mu(A), \tag{2}
\]

for any Borelian set \( A \subset [0, \infty[^d \) which is bounded away from 0 and such that the limit measure \( \mu \) of the boundary \( \partial A \) is zero. For a complete introduction to the theory of Regular Variation, the reader may refer to [47]. The measure \( \mu \) may be understood as the limit distribution of tail events. In (2), \( \mu \) is homogeneous of order \(-1\), that is \( \mu(tA) = t^{-1}\mu(A), \ t > 0, A \subset [0, \infty[^d \setminus \{0\} \}. \) This scale invariance is key for our purposes, as detailed in Section 2.2. The main idea behind extreme value analysis is to learn relevant features of \( \mu \) using the largest available data.

### 2.2 Classification in extreme regions

We now recall the classification setup for extremes as introduced in [30]. Let \( (X, Y) \in \mathbb{R}^d_+ \times \{-1, 1\} \) be a random pair. Authors of [30] assume standard regular variation for both classes, that is \( tP\{X \in tA \mid Y = \pm 1\} = t^{-\alpha} \mu_\pm(A) \), where \( A \) is as in (2). Let \( \|\cdot\| \) be any norm on \( \mathbb{R}^d \) and consider the risk of a classifier \( g : \mathbb{R}^d_+ \to \{\pm 1\} \) above a radial threshold \( t \),

\[
L_t(g) = P\{Y \neq g(X) \mid \|X\| > t\}. \tag{3}
\]

The goal is to minimize the asymptotic risk in the extremes \( L_\infty(g) = \limsup_{t \to \infty} L_t(g) \). Using the scale invariance property of \( \mu \), under additional mild regularity assumptions concerning the regression function, namely uniform convergence to the limit at infinity, one can prove the following result
We now introduce a novel algorithm \textbf{LHTR} (with corresponding sorted label \(Y\) (see [30], Theorem 1): there exists a classifier \(\text{Equation (4) in Appendix B.3}\), for \(\delta\) varying one. A workable example of such a target is provided in our experiments (Section 4). As a consequence, the optimal classifiers on extreme regions are based on indicator functions of truncated cones on the kind \(\{\|x\| > t, \Theta(x) \in B\}\), where \(B \subset S\), see \textbf{Figure 1}. We emphasize that the labels provided by such a classifier remain unchanged when rescaling the samples by a factor \(\lambda \geq 1\) (i.e. \(g(x) = g(\Theta(x)) = g(\Theta(Lx)), \forall x \in \{x, \|x\| \geq t\}\)). The angular structure of the optimal classifier \(g^*\) is the basis for the following ERM strategy using the most extreme points of a dataset. Let \(G_S\) be a class of angular classifiers defined on the sphere \(S\) with finite VC dimension \(V_{G_S} < \infty\). By extension, for any \(x \in \mathbb{R}^d_+\) and \(g \in G_S\), \(g(x) = g(\Theta(x)) \in \{-1, 1\}\). Given \(n\) training data \(\{(X_i, Y_i)\}_{i=1}^n\) made of \(i.i.d\) copies of \((X, Y)\), sorting the training observations by decreasing order of magnitude, let \(X_{(i)}\) (with corresponding sorted label \(Y_{(i)}\)) denote the \(i\)-th order statistic, i.e. \(\|X_{(1)}\| \geq \ldots \geq \|X_{(n)}\|\). The empirical risk for the \(k\) largest observations \(\hat{L}_k(g) = \frac{1}{k} \sum_{i=1}^{k} 1\{Y_{(i)} \neq g(\Theta(X_{(i)}))\}\) is an empirical version of the risk \(L_{\delta(k)}(g)\) as defined in (3) where \(t(k)\) is a \((1 - k/n)\)-quantile of the norm, \(p \{\|X\| > t(k)\} = k/n\). Selection of \(k\) is a bias-variance compromise, see Appendix B for further discussion. The strategy promoted by [30] is to use \(\hat{g}_k = \arg\min_{g \in G_S} \hat{L}_k(g)\), for classification in the extreme region \(\{x \in \mathbb{R}^d_+ : \|x\| > t(k)\}\). The following result provides guarantees concerning the excess risk of \(\hat{g}_k\) compared with the Bayes risk above level \(t = t(k)\), \(L_\delta = \inf_{g \in G_S} L_{\delta(k)}(g)\).

**Theorem 1** ([30], Theorem 2) If each class satisfies the regular variation assumption \(\Theta\), under an additional regularity assumption concerning the regression function \(\eta(x) = \mathbb{P}\{Y = +1 \mid x\}\) (see \textbf{Equation 4} in Appendix B.3), for \(\delta \in (0, 1), \forall n \geq 1\), it holds with probability larger than \(1 - \delta\) that

\[
L_{\delta(k)}(g) - L_\delta \leq \frac{1}{\sqrt{k}} \left( \frac{\sqrt{2(1 - k/n)} \log(2/\delta)}{1 - \delta} + C \sqrt{V_{G_S}} \log(1/\delta) \right) + \frac{1}{\sqrt{k}} \left( 5 + 2 \log(1/\delta) + \sqrt{\log(1/\delta)} \left( C \sqrt{V_{G_S}} + \sqrt{2} \right) \right) + \left\{ \inf_{g \in G_S} L_{\delta(k)}(g) - L_\delta \right\},
\]

where \(C\) is a universal constant.

In the present work we do not assume that the baseline representation \(X\) for text data satisfies the assumptions of Theorem 1. Instead, our goal is to render the latter theoretical framework applicable by learning a representation which satisfies the regular variation condition given in (2), hereafter referred as Condition (2) which is the main assumption for Theorem 1 to hold. Our experiments demonstrate empirically that enforcing Condition (2) is enough for our purposes, namely improved classification and label preserving data augmentation, see Appendix B.3 for further discussion.

### 3 Heavy-tailed Text Embeddings

#### 3.1 Learning a heavy-tailed representation

We now introduce a novel algorithm \textit{Learning a heavy-tailed representation (LHTR)} for text data from high dimensional vectors as issued by pre-trained embeddings such as BERT. The idea behind is to modify the output \(X\) of BERT so that classification in the tail regions enjoys the statistical guarantees presented in Section 2 while classification in the bulk (where many training points are available) can still be performed using standard models. Stated otherwise, LHTR increases the information carried by the resulting vector \(Z = \varphi(X) \in \mathbb{R}^d\) regarding the label \(Y\) in the tail regions of \(Z\) in order to improve the performance of a downstream classifier. In addition LHTR is a building block of the data augmentation algorithm \textbf{GENELIX} detailed in Section 3.2. LHTR proceeds by training an encoding function \(\varphi\) in such a way that (i) the marginal distribution \(q(z)\) of the code \(Z\) be close to a user-specified heavy tailed distribution \(p\) satisfying the regularity condition (2); and (ii) the classification loss of a multilayer perceptron trained on the code \(Z\) be small.

A major difference distinguishing LHTR from existing auto-encoding schemes is that the target distribution on the latent space is not chosen as a Gaussian distribution but as a heavy-tailed, regularly varying one. A workable example of such a target is provided in our experiments (Section 4). As the Bayes classifier (i.e. the optimal one among all possible classifiers) in the extreme region has a
potentially different structure from the Bayes classifier on the bulk (recall from Section 2) that the optimal classifier at infinity depends on the angle $\Theta(x)$ only. LHTR trains two different classifiers, $g^{\text{ext}}$ on the extreme region of the latent space on the one hand, and $g^{\text{bulk}}$ on its complementary set on the other hand. Given a high threshold $t$, the extreme region of the latent space is defined as the set $\{z : \|z\| > t\}$. In practice, the threshold $t$ is chosen as an empirical quantile of order $(1 - \kappa)$ (for some small, fixed $\kappa$) of the norm of encoded data $\|Z_i\| = \|\varphi(X_i)\|$. The classifier trained by LHTR is thus of the kind $g(z) = g^{\text{ext}}(z) \mathbb{1}(\|z\| > t) + g^{\text{bulk}}(z) \mathbb{1}(\|z\| \leq t)$. If the downstream task is classification on the whole input space, in the end the bulk classifier $g^{\text{bulk}}$ may be replaced with any other classifier $g'$ trained on the original input data $X$ restricted to the non-extreme samples ($i.e.$ $\{X_i, \|\varphi(X_i)\| \leq t\}$). Indeed training $g^{\text{bulk}}$ only serves as an intermediate step to learn an adequate representation $\varphi$.

**Remark 1** Recall from Section 2.2 that the optimal classifier in the extreme region as $t \to \infty$ depends on the angular component $\vartheta(x)$ only, or in other words, is scale invariant. One can thus reasonably expect the trained classifier $g^{\text{ext}}(z)$ to enjoy the same property. This scale invariance is indeed verified in our experiments (see Sections 4 and 5) and is the starting point for our data augmentation algorithm in Section 3.2. An alternative strategy would be to train an angular classifier, i.e. to impose scale invariance. However in preliminary experiments (not shown here), the resulting classifier was less efficient and we decided against this option in view of the scale invariance and better performance of the unconstrained classifier.

The goal of LHTR is to minimize the weighted risk

$$R(\varphi, g^{\text{ext}}, g^{\text{bulk}}) = p_1 \mathbb{P}\{Y \neq g^{\text{ext}}(Z), \|Z\| \geq t\} + p_2 \mathbb{P}\{Y \neq g^{\text{bulk}}(Z), \|Z\| < t\} + p_3 \Theta(q(z), p(z)),$$

where $Z = \varphi(X)$, $\mathcal{D}$ is the Jensen-Shannon distance between the heavy tailed target distribution $p$ and the code distribution $q$, and $p_1, p_2, p_3$ are positive weights. Following common practice in the adversarial literature, the Jensen-Shannon distance is approached (up to a constant term) by the empirical proxy $\hat{L}(q, p) = \sup_{D \in \mathcal{D}} \bar{L}(q, p, D)$, with $\bar{L}(q, p, D) = \frac{1}{n} \sum_{i=1}^{n} \log D(Z_i) + \log (1 - D(\bar{Z}_i))$, where $\Gamma$ is a wide class of discriminant functions valued in $[0, 1]$, and where independent samples $Z_i, \bar{Z}_i$ are respectively sampled from the target distribution and the code distribution $q$.

Further details on adversarial learning are provided in Appendix A.1. The classifiers $g^{\text{ext}}, g^{\text{bulk}}$ are of the form $g^{\text{ext}}(z) = 2 \mathbb{1}(C^{\text{ext}}(z) > 1/2) - 1$, $g^{\text{bulk}}(z) = 2 \mathbb{1}(C^{\text{bulk}}(z) > 1/2) - 1$ where $C^{\text{ext}}, C^{\text{bulk}}$ are also discriminant functions valued in $[0, 1]$. Following common practice, we shall refer to $C^{\text{ext}}, C^{\text{bulk}}$ as classifiers as well. In the end, LHTR solves the following min-max problem

$$\inf_{C^{\text{ext}}, C^{\text{bulk}}, \varphi} \sup_{D} \bar{R}(\varphi, C^{\text{ext}}, C^{\text{bulk}}, D)$$

with

$$\bar{R}(\varphi, C^{\text{ext}}, C^{\text{bulk}}, D) = \frac{p_1}{k} \sum_{i=1}^{k} \ell(Y(i), C^{\text{ext}}(Z(i))) + \frac{p_2}{n-k} \sum_{i=k+1}^{n-k} \ell(Y(i), C^{\text{bulk}}(Z(i))) + p_3 \hat{L}(q, p, D),$$

where $\{Z(i) = \varphi(X(i)), i = 1, \ldots, n\}$ are the encoded observations with associated labels $Y(i)$ sorted by decreasing magnitude of $\|Z\|$ ($i.e.$ $\|Z(1)\| \geq \cdots \geq \|Z(n)\|$), $k = \lfloor \kappa n \rfloor$ is the number of extreme samples among the $n$ encoded observations and $\ell(y, C(x)) = -(y \log C(x) + (1-y) \log(1-C(x)))$ is the negative log-likelihood of the discriminant function $C(x) \in (0, 1)$. A summary of LHTR and an illustration of its workflow are provided in Appendices A.2 and A.3.

### 3.2 A heavy-tailed representation for dataset augmentation

We now introduce GENELIEX (Generating Label Invariant sequences from Extremes), a data augmentation algorithm, which relies on the label invariance property under rescaling of the classifier for the extremes learnt by LHTR. GENELIEX considers input sentences as sequences and follows the seq2seq approach [52][16][7]. It trains a Transformer Decoder [54] $G^{\text{ext}}$ on the extreme regions.

For an input sequence $U = (u_1, \ldots, u_T)$ of length $T$, represented as $X_U$ by BERT with latent code $Z = \varphi(X_U)$ lying in the extreme regions, GENELIEX produces, through its decoder $G^{\text{ext}}$ $M$ sequences $U'_j$ where $j \in \{1, \ldots, M\}$. The $M$ decoded sequences correspond to the codes $\{\lambda_j Z, j \in \{1, \ldots, M\}\}$ where $\lambda_j > 1$. To generate sequences, the decoder iteratively takes as input the previously generated word (the first word being a start symbol), updates its internal state, and returns the next word with the highest probability. This process is repeated until either the
decoder generates a stop symbol or the length of the generated sequence reaches the maximum length $T_{\text{max}}$. To train the decoder $G^{\text{ext}} : \mathbb{R}^d \to [1, \ldots, |\mathcal{V}|]^T_{\text{max}}$ where $\mathcal{V}$ is the vocabulary on the extreme regions, GENELIEX requires an additional dataset $\mathcal{D}_{\text{gen}} = (U_1, \ldots, U_n)$ (not necessarily labeled) with associated representation via BERT $(X_{U,1}, \ldots, X_{U,n})$. Learning is carried out by optimising the classical negative log-likelihood of individual tokens $\ell_{\text{gen}}$. The latter is defined as $\ell_{\text{gen}}(U; G^{\text{ext}}(X)) \overset{\text{def}}{=} \sum_{t=1}^{T_{\text{max}}} \sum_{v \in \mathcal{V}} \chi \{ u_t = v \} \log(p_{v,t})$, where $p_{v,t}$ is the probability predicted by $G^{\text{ext}}$ that the $t^{th}$ word is equal to $v$. A detailed description of the training step of GENELIEX is provided in Algorithm 2 in Appendix A.3 see also Appendix A.2 for an illustrative diagram.

Remark 2 Note that the proposed method only augments data on the extreme regions. A general data augmentation algorithm can be obtained by combining this approach with any other algorithm on the original input data $X$ whose latent code $Z = \varphi(X_U)$ does not lie in the extreme regions.

4 Experiments : Classification

In our experiments we work with the infinity norm. The proportion of extreme samples in the training step of LHTR is chosen as $\kappa = 1/4$. The threshold $t$ defining the extreme region $\{||x|| > t\}$ in the test set is $t = \langle Z_{(\kappa \text{NN})} \rangle$ as returned by LHTR. We denote by $T_{\text{test}}$ and $T_{\text{train}}$ respectively the extreme test and train sets thus defined. Classifiers $C^{\text{bulk}}, C^{\text{ext}}$ involved in LHTR are Multi Layer Perceptrons (MLP), see Appendix B.6 for a full description of the architectures.

Heavy-tailed distribution. The regularly varying target distribution is chosen as a multivariate logistic distribution with parameter $\delta = 0.9$, refer to Appendix B.4 for details and an illustration with various values of $\delta$. This distribution is widely used in the context of extreme values analysis [10, 53, 23] and differ from the classical logistic distribution.

4.1 Toy example: about LHTR

We start with a simple bivariate illustration of the heavy tailed representation learnt by LHTR. Our goal is to provide insight on how the learnt mapping $\varphi$ acts on the input space and how the transformation affects the definition of extremes (recall that extreme samples are defined as those samples which norm exceeds an empirical quantile). Labeled samples are simulated from a Gaussian mixture

![Figure 2](image_url)

Figure 2: Figure 2a Bivariate samples $X_i$ in the input space. Figure 2b $X_i$’s in the input space with extremes from each class selected in the input space. Figure 2c Latent space representation $Z_i = \varphi(X_i)$. Extremes of each class are selected in the latent space. Figure 2d $X_i$’s in the input space with extremes from each class selected in the latent space.

distribution with two components of identical weight. The label indicates the component from which
Figure 2c) In Figure 2b, the extreme raw observations are selected according to their norm after a component-wise standardisation of $X_t$, refer to Appendix B for details. The extreme threshold $t$ is chosen as the 75% empirical quantile of the norm on the training set in the input space. Notice in the latter figure the class imbalance among extremes. In Figure 2c, extremes are selected as the 25% samples with the largest norm in the latent space. Figure 2d is similar to Figure 2c, except for the selection of extremes which is performed in the latent space as in Figure 2c. On this toy example, the adversarial strategy appears to succeed in learning a code which distribution is close to the logistic target, as illustrated by the similarity between Figure 2c and Figure 5a in the supplementary. In addition, the heavy tailed representation allows a more balanced selection of extremes than the input representation.

4.2 Application to positive vs. negative classification of sequences

In this section, we dissect LHTR to better understand the relative importance of: (i) working with a heavy-tailed representation, (ii) training two independent classifiers: one dedicated to the bulk and the second one dedicated to the extremes. In addition, we verify experimentally that the latter classifier is scale invariant, which is neither the case for the former, nor for a classifier trained on BERT input.

Experimental settings. We compare the performance of three models. The baseline NN model is a MLP trained on BERT. The second model LHTR$_2$ is a variant of LHTR where a single MLP ($C$) is trained on the output of the encoder $\varphi$, using all the available data, both extreme and non extreme ones. The third model (LHTR) trains two separate MLP classifiers $C^{ext}$ and $C^{bulk}$ respectively dedicated to the extreme and bulk regions of the learnt representation $\varphi$. All models take the same training inputs, use BERT embedding and their classifiers have identical structure, see Appendix A.2 and B.6 for a summary of model workflows and additional details concerning the network architectures.

Comparing LHTR$_2$ with NN model assesses the relevance of working with heavy-tailed embeddings. Since LHTR$_2$ is obtained by using LHTR with $C^{ext} = C^{bulk}$, comparing LHTR$_2$ with LHTR validates the use of two separate classifiers so that extremes are handled in a specific manner. As we make no claim concerning the usefulness of LHTR in the bulk, at the prediction step we suggest working with a combination of two models: LHTR with $C^{ext}$ for extreme samples and any other off-the-shelf ML tool for the remaining samples (e.g. NN model).

Datasets. In our experiments we rely on two large datasets from Amazon (231k reviews) [41] and from Yelp (1,450k reviews) [58, 36]. Reviews, (made of multiple sentences) with a rating greater than or equal to 4/s are labeled as +1, while those with a rating smaller or equal to 2/s are labeled as −1. The gap in reviews’ ratings is designed to avoid any overlap between labels of different contents.

Results. Figure 3 gathers the results obtained by the three considered classifiers on the tail regions of the two datasets mentioned above. To illustrate the generalization ability of the proposed classifier in the extreme regions we consider nested subsets of the extreme test set $T_{ext}$: $T_\lambda = \{(x,t) \in T_{ext}, \|x\| \geq \lambda t\}$, $\lambda \geq 1$. For all factor $\lambda$ greater than or equal to 1, $T_\lambda \subseteq T_{ext}$. The greater $\lambda$, the fewer the samples retained for evaluation and the greater their norms. On both datasets, LHTR$_2$ outperforms the baseline NN model. This shows the improvement offered by the heavy-tailed embedding on the extreme region. In addition, LHTR$_1$ is in turn largely outperformed by the classifier LHTR, which proves the importance of working with two separate classifiers. The performance of the proposed model respectively on the bulk region, tail region and overall, is reported in Table 1 which shows that using a specific classifier dedicated to extremes improves the overall performance.
Table 1: Classification losses on Amazon and Yelp. ‘Proposed Model’ results from using NN model for the bulk and LHTR for the extreme test sets. The extreme region contains 6.9k samples for Amazon and 6.1k samples for Yelp, both corresponding roughly to 25% of the whole test set size.

Scale invariance. On all datasets, the extreme classifier $g_{\text{ext}}$ verifies Equation (1) for each sample of the test set, $g_{\text{ext}}(\lambda Z) = g_{\text{ext}}(Z)$ with $\lambda$ ranging from 1 to 20, demonstrating scale invariance of $g_{\text{ext}}$ on the extreme region. The same experiments conducted both with NN model and a MLP classifier trained on BERT and LHTR 1 show label changes for varying values of $\lambda$; none of them are scale invariant. Appendix B.5 gathers additional experimental details. The scale invariance property will be exploited in the next section to perform label invariant generation.

5 Experiments: Label Invariant Generation

5.1 Experimental Setting

Comparison with existing work. We compare GENELIEX with two state of the art methods for dataset augmentation, Wei and Zou [56] and Kobayashi [32]. Contrarily to these works which use heuristics and a synonym dictionary, GENELIEX does not require any linguistic resource. To ensure that the improvement brought by GENELIEX is not only due to BERT, we have updated the method in [32] with a BERT language model (see Appendix B.7 for details and Table 7 for hyperparameters).

Evaluation Metrics. Automatic evaluation of generative models for text is still an open research problem. We rely both on perceptive evaluation and automatic measures to evaluate our model through four criteria ($C_1$, $C_2$, $C_3$, $C_4$). $C_1$ measures Cohesion [17] (Are the generated sequences grammatically and semantically consistent?). $C_2$ (named Sent. in Table 3) evaluates label conservation (Does the expressed sentiment in the generated sequence match the sentiment of the input sequence?). $C_3$ measures the diversity [35] (corresponding to dist1 or dist2 in Table 3) of the sequences (Does the augmented dataset contain diverse sequences?). Augmenting the training set with very diverse sequences can lead to better classification performance. $C_4$ measures the improvement in terms of F1 score when training a classifier (fastText [31]) on the augmented training set (Does the augmented dataset improve classification performance?).

Datasets. GENELIEX is evaluated on two datasets, a medium and a large one (see [50]) which respectively contains 1k and 10k labeled samples. In both cases, we have access to $D_{\text{gn}}$, a dataset of 80k unlabeled samples. Datasets are randomly sampled from Amazon and Yelp.

Experiment description. We augment extreme regions of each dataset according to three algorithms: GENELIEX (with scaling factor $\lambda$ ranging from 1 to 1.5), Kobayashi [32], and Wei and Zou [56]. For each train set’s sequence considered as extreme, 10 new sequences are generated using each algorithm. Appendix B.7 gathers further details. For experiment $C_4$ the test set contains $10^4$ sequences.

5.2 Results

Automatic measures. The results of $C_3$ and $C_4$ evaluation are reported in Table 2. Augmented data with GENELIEX are more diverse than the one augmented with Kobayashi [32] and Wei and Zou [56]. The F1-score with dataset augmentation performed by GENELIEX outperforms the aforementioned methods on Amazon in medium and large dataset and on Yelp for the medium dataset. It equals state of the art performances on Yelp for the large dataset. As expected, for all three algorithms, the benefits of data augmentation decrease as the original training dataset size increases. Interestingly, we observe a strong correlation between more diverse sequences in the extreme regions and higher F1 score: the more diverse the augmented dataset, the higher the F1 score. More diverse sequences

\[^2\text{dist} n \text{ is obtained by calculating the number of distinct } n\text{-grams divided by the total number of generated tokens to avoid favoring long sequences.}\]
Table 2: Quantitative Evaluation. Algorithms are compared according to C3 and C4. dist1 and dist2 respectively stand for distinct 1 and 2, it measures the diversity of new sequences in terms of unigrams and bigrams. F1 is the F1-score for FastText classifier trained on an augmented labelled training set.

<table>
<thead>
<tr>
<th>Model</th>
<th>Amazon: F1 dist1/dist2</th>
<th>Yelp: F1 dist1/dist2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Raw Data</td>
<td>84.0</td>
<td>X</td>
</tr>
<tr>
<td>Kobayashi [32]</td>
<td>85.0 0.10/0.47</td>
<td>92.9 0.14/0.53</td>
</tr>
<tr>
<td>Wei and Zou [56]</td>
<td>85.2 0.11/0.50</td>
<td>93.2 0.14/0.54</td>
</tr>
<tr>
<td>GENELIEX</td>
<td>86.3 0.14/0.52</td>
<td>94.0 0.18/0.58</td>
</tr>
</tbody>
</table>

Table 3: Qualitative evaluation with three turkers. Sent. stands for sentiment label preservation. The Krippendorff Alpha for Amazon is $\alpha = 0.28$ on the sentiment classification and $\alpha = 0.20$ for cohesion. The Krippendorff Alpha for Yelp is $\alpha = 0.57$ on the sentiment classification and $\alpha = 0.48$ for cohesion.

<table>
<thead>
<tr>
<th>Model</th>
<th>Amazon: Sent. Cohesion</th>
<th>Yelp: Sent. Cohesion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Raw Data</td>
<td>83.6 78.3</td>
<td>80.6 0.71</td>
</tr>
<tr>
<td>Kobayashi [32]</td>
<td>80.0 84.2</td>
<td>82.9 0.72</td>
</tr>
<tr>
<td>Wei and Zou [56]</td>
<td>69.0 67.4</td>
<td>80.0 0.60</td>
</tr>
<tr>
<td>GENELIEX</td>
<td>78.4 73.2</td>
<td>85.7 0.77</td>
</tr>
</tbody>
</table>

are thus more likely to lead to better improvement on downstream tasks (e.g. classification).

**Perceptive Measures.** To evaluate C1, C2, three turkers were asked to annotate the cohesion and the sentiment of 100 generated sequences for each algorithm and for the raw data. F1 scores of this evaluation are reported in Table 3. Grammar evaluation confirms the findings of [56] showing that random swaps and deletions do not always maintain the cohesion of the sequence. In contrast, GENELIEX and Kobayashi [32], using vectorial representations, produce more coherent sequences. Concerning sentiment label preservation, on Yelp, GENELIEX achieves the highest score which confirms the observed improvement reported in Table 2. On Amazon, turker annotations with data from GENELIEX obtain a lower F1-score than from Kobayashi [32]. This does not correlate with results in Table 2 and may be explained by a lower Krippendorff Alpha on Amazon ($\alpha = 0.20$) than on Yelp ($\alpha = 0.57$).

### 6 Broader Impact

In this work, we propose a method resulting in heavy-tailed text embeddings. As we make no assumption on the nature of the input data, the suggested method is not limited to textual data and can be extended to any type of modality (e.g. audio, video, images). A classifier, trained on aforementioned embedding is dilation invariant (see Equation 1) on the extreme region. A dilation invariant classifier enables better generalization for new samples falling out of the training envelop. For critical application ranging from web content filtering (e.g. spam [27], hate speech detection [18], fake news [43]) to medical case reports to court decisions it is crucial to build classifiers with lower generalization error. The scale invariance property can also be exploited to automatically augment a small dataset on its extreme region. For application where data collection requires a huge effort both in time and cost (e.g. industrial factory design, classification for rare language [4]), beyond industrial aspect, active learning problems involving heavy-tailed data may highly benefit from our data augmentation approach.

### 7 Acknowledgement

Anne Sabourin was partly supported by the Chaire Stress testing from Ecole Polytechnique and BNP Paribas. Concerning Eric Gaussier, this project partly fits within the MIAI project (ANR-19-P3IA-0003).

---

$^3$ measure of inter-rater reliability in $[0, 1]$: 0 is perfect disagreement and 1 is perfect agreement.
References


[58] Xiao Yu, Xiang Ren, Yizhou Sun, Quanquan Gu, Bradley Sturt, Urvashi Khandelwal, Brandon Norick, and Jiawei Han. Personalized entity recommendation: A heterogeneous information network approach. In Proceedings of the 7th ACM international conference on Web search and data mining, pages 283–292. ACM, 2014.
A Models

A.1 Background on Adversarial Learning

Adversarial networks, introduced in [25], form a system where two neural networks are competing. A first model $G$, called the generator, generates samples as close as possible to the input dataset. A second model $D$, called the discriminator, aims at distinguishing samples produced by the generator from the input dataset. The goal of the generator is to maximize the probability of the discriminator making a mistake. Hence, if $P_{\text{input}}$ is the distribution of the input dataset then the adversarial network intends to minimize the distance (as measured by the Jensen-Shannon divergence) between the distribution of the generated data $P_G$ and $P_{\text{input}}$. In short, the problem is a minmax game with value function $V(D,G)$

$$
\min_G \max_D V(D,G) = E_{x \sim P_{\text{input}}} \left[ \log D(x) \right] + E_{z \sim P_G} \left[ \log (1 - D(G(z))) \right].
$$

Auto-encoders and derivatives [26, 34, 21] form a subclass of neural networks whose purpose is to build a suitable representation by learning encoding and decoding functions which capture the core properties of the input data. An adversarial auto-encoder (see [39]) is a specific kind of auto-encoders where the encoder plays the role of the generator of an adversarial network. Thus the latent code is forced to follow a given distribution while containing information relevant to reconstructing the input. In the remaining of this paper, a similar adversarial encoder constrains the encoded representation to be heavy-tailed.

A.2 Models Overview

Figure 4 provides an overview of the different algorithms proposed in the paper. Figure 4a describes the pipeline for LHTR detailed in Algorithm 1. Figure 4b describes the pipeline for the comparative baseline LHTR where $C_{\text{ext}} = C_{\text{bulk}}$. Figure 4c illustrates the pipeline for the baseline classifier trained on BERT. Figure 4d describes GENELIEX described in Algorithm 2, note that the hatched components are inherited from LHTR and are not used in the workflow.

![Figure 4: Illustrative pipelines.](image)

A.3 LHTR and GENELIEX algorithm

This subsection provides detailed algorithm for both models LHTR and GENELIEX.
Algorithm 1 LHTR

**INPUT:** Weighting coef. $\rho_1, \rho_2, \rho_3 > 0$, Training dataset $D_n = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$, batch size $m$, proportion of extremes $\kappa$, heavy tailed prior $P_Z$.

**Initialization:** parameters $(\tau, \theta, \theta', \gamma)$ of the encoder $\varphi_\tau$, classifiers $C^\text{ext}_\theta$, $C^\text{bulk}_\theta$ and discriminator $D_\gamma$.

**Optimization:**

while $(\tau, \theta, \theta', \gamma)$ not converged do
    Sample $\{(X_1, Y_1), \ldots, (X_m, Y_m)\}$ from $D_n$ and define $\tilde{Z}_i = \varphi(X_i)$, $i \leq m$.
    Sample $\{Z_1, \ldots, Z_m\}$ from the prior $P_Z$.
    Update $\gamma$ by ascending:
    \[
    \frac{\rho_1}{m} \sum_{i=1}^{m} \log D_\gamma(Z_i) + \log(1 - D_\gamma(\tilde{Z}_i)).
    \]
    Sort $\{\tilde{Z}_i\}_{i \in \{1, \ldots, m\}}$ by decreasing order of magnitude $||\tilde{Z}_{(1)}|| \geq \ldots \geq ||\tilde{Z}_{(m)}||$.
    Update $\theta$ by descending:
    \[
    \mathcal{L}^\text{ext}(\theta, \tau) \overset{\text{def}}{=} \frac{\rho_1}{|\kappa m|} \sum_{i=1}^{\lfloor \kappa m \rfloor} \ell(Y_i, C^\text{ext}_\theta(\tilde{Z}_i)).
    \]
    Update $\theta'$ by descending:
    \[
    \mathcal{L}^\text{bulk}(\theta', \tau) \overset{\text{def}}{=} \frac{\rho_2}{m - |\kappa m|} \sum_{i=|\kappa m|+1}^{m} \ell(Y_i, C^\text{bulk}_\theta(\tilde{Z}_i)).
    \]
    Update $\tau$ by descending:
    \[
    \frac{1}{m} \sum_{i=1}^{m} -\rho_3 \log D_\gamma(\tilde{Z}_i) + \mathcal{L}^\text{ext}(\theta, \tau) + \mathcal{L}^\text{bulk}(\theta', \tau).
    \]
end while
Compute $\{\tilde{Z}_i\}_{i \in \{1, \ldots, n\}} = \varphi(X_i)_{i \in \{1, \ldots, n\}}$
Sort $\{\tilde{Z}_i\}_{i \in \{1, \ldots, n\}}$ by decreasing order of magnitude $||\tilde{Z}_{(1)}|| \geq \ldots ||\tilde{Z}_{(\lfloor \kappa n \rfloor)}|| \geq \ldots \geq ||\tilde{Z}_{(n)}||$.

**OUTPUT:** encoder $\varphi$, classifiers $C^\text{ext}$ for $\{x : ||\varphi(x)|| \geq t := ||\tilde{Z}_{(\lfloor \kappa n \rfloor)}||\}$ and $C^\text{bulk}$ on the complementary set.

Algorithm 2 GENELIE\textsubscript{X}: training step

**INPUT:** input of LHTR, $D_n = \{U_1, \ldots, U_n\}$

**Initialization:** parameters of $\varphi_\tau$, $C^\text{ext}_\theta$, $C^\text{bulk}_\theta$, $D_\gamma$ and decoder $G^\text{ext}_\psi$

**Optimization:**

$\varphi, C^\text{ext}, C^\text{bulk} = \text{LHTR}(\rho_1, \rho_2, \rho_3, D_n, \kappa, m)$
while $\psi$ not converged do
    Sample $\{U_1, \ldots, U_m\}$ from the training set $D_n$ and define $\tilde{Z}_i = \varphi(X_{U,i})$ for $i \in \{1, \ldots, m\}$.
    Sort $\{\tilde{Z}_i\}_{i \in \{1, \ldots, m\}}$ by decreasing order of magnitude $||\tilde{Z}_{(1)}|| \geq \ldots \geq ||\tilde{Z}_{(m)}||$.
    Update $\psi$ by descending:
    \[
    \mathcal{L}^\text{ext}_g(\psi) \overset{\text{def}}{=} \frac{\rho_1}{|\kappa m|} \sum_{i=1}^{\lfloor \kappa m \rfloor} \ell_{\text{gen}}(U_i, G^\text{ext}_\psi(\tilde{Z}_i)).
    \]
end while
Compute $\{\tilde{Z}_i\}_{i \in \{1, \ldots, n\}} = \varphi(X_i)_{i \in \{1, \ldots, n\}}$
Sort $\{\tilde{Z}_i\}_{i \in \{1, \ldots, n\}}$ by decreasing order of magnitude $||\tilde{Z}_{(1)}|| \geq \ldots ||\tilde{Z}_{(\lfloor \kappa n \rfloor)}|| \geq \ldots \geq ||\tilde{Z}_{(n)}||$.

**OUTPUT:** encoder $\varphi$, decoder $G^\text{ext}$ applicable on the region $\{x : ||\varphi(x)|| \geq t := ||\tilde{Z}_{(\lfloor \kappa n \rfloor)}||\}$
B Extreme Value Analysis: additional material

B.1 Choice of \( k \)

To the best of our knowledge, selection of \( k \) in extreme value analysis (in particular in Algorithm 1 and Algorithm 2) is still a vivid problem in EVT for which no absolute answer exists. As \( k \) gets large the number of extreme points increases including samples which are not large enough and deviates from the asymptotic distribution of extremes. Smaller values of \( k \) increase the variance of the classifier/generator. This bias-variance trade-off is beyond the scope of this paper.

B.2 Preliminary standardization for selecting extreme samples

In Figure 2 selecting the extreme samples on the input space is not a straightforward step as the two components of the vector are not on the same scale, componentwise standardisation is a natural and necessary preliminary step. Following common practice in multivariate extreme value analysis it was decided to standardise the input data \((X_i)_{i \in \{1, \ldots, n\}}\) by applying the rank-transformation:

\[
\hat{T}(x) = \left(1 / \left(1 - \hat{F}_j(x)\right)\right)_{j=1, \ldots, d}
\]

for all \( x = (x^1, \ldots, x^d) \in \mathbb{R}^d \) where \( \hat{F}_j(x) \triangleq \frac{1}{n^{\frac{1}{d}}} \sum_{i=1}^{n} \mathbb{I}\{X^j_i \leq x\} \) is the \( j \)th empirical marginal distribution. Denoting by \( V_i \) the standardized variables, \( \forall i \in \{1, \ldots, n\}, V_i = \hat{T}(X_i) \). The marginal distributions of \( V_i \) are well approximated by standard Pareto distribution, the approximation error comes from the fact that the empirical c.d.f’s are used in \( \hat{T} \) instead of the genuine marginal c.d.f’s \( F_j \). After this standardization step, the selected extreme samples are \( \{V_i, \|V_i\| \geq \hat{V}_{\text{ini}}\} \).

B.3 Enforcing regularity assumptions in Theorem 1

The methodology in the present paper consists in learning a representation \( Z \) for text data via \textsc{LHTR} satisfying the regular variation condition (2). This condition is weaker than the assumptions from Theorem 1 for two reasons: first, it does not imply that each class (conditionally to the label \( Y \)) is regularly varying, only that the distribution of \( Z \) (unconditionally to the label) is. Second, in Jalalzai et al. [30], it is additionally required that the regression function \( \eta(z) = \mathbb{P}\{Y = +1 | Z = z\} \) converges uniformly as \( \|z\| \to \infty \). Getting into details, one needs to introduce a limit random pair \((Z_{\infty}, Y_{\infty})\) which distribution is the limit of \( \mathbb{P}\left\{Y = \cdot, t^{-1}Z \in \cdot | \|Z\| > t\right\} \) as \( t \to \infty \). Denote by \( \eta_{\infty} \) the limiting regression function, \( \eta_{\infty}(z) = \mathbb{P}\{Y_{\infty} = +1 | Z_{\infty} = z\} \). The required assumption is that

\[
\sup_{\{z \in \mathbb{R}^d : \|z\| > t\}} |\eta(z) - \eta_{\infty}(z)| \xrightarrow{t \to \infty} 0.
\]

Uniform convergence (4) is not enforced in \textsc{LHTR} and the question of how to enforce it together with regular variation of each class separately remains open. However, our experiments in sections 4 and 5 demonstrate that enforcing Condition 2 is enough for our purposes, namely improved classification and label preserving data augmentation.

B.4 Logistic distribution

The logistic distribution with dependence parameter \( \delta \in (0, 1) \) is defined in \( \mathbb{R}^d \) by its c.d.f. \( F(x) = \exp\left\{-\sum_{j=1}^{d} x^{(j)} \hat{F_j}(x)\right\} \). Samples from the logistic distribution can be simulated according to the algorithm proposed in Stephenson [51]. Figure 3 illustrates this distribution with various values of \( \delta \). Values of \( \delta \) close to 1 yield non concomitant extremes, i.e. the probability of a simultaneous excess of a high threshold by more than one vector component is negligible. Conversely, for small values of \( \delta \), extreme values tend to occur simultaneously. These two distinct tail dependence structures are respectively called ‘asymptotic independence’ and ‘asymptotic dependence’ in the EVT terminology.

B.5 Scale invariance comparison of BERT and \textsc{LHTR}

In this preliminary experiment we compare \textsc{LHTR} and BERT and show that the latter is not scale invariant. For this preliminary experiment we rely on labeled fractions of both Amazon and Yelp datasets respectively.
Figure 5: Illustration of the distribution of the angle $\Theta(X)$ obtained with bivariate samples $X$ generated from a logistic model with different coefficients of dependence ranging from near asymptotic independence (Figure 5a ($\delta = 0.9$)) to high asymptotic dependence (Figure 5c ($\delta = 0.1$)) including moderate dependence (Figure 5b ($\delta = 0.5$)). Non extreme samples are plotted in gray, extreme samples are plotted in black and the angles $\Theta(X)$ (extreme samples projected on the sup norm sphere) are plotted in red. Note that not all extremes are shown since the plot was truncated for a better visualization. However all projections on the sphere are shown.

denoted as Amazon small dataset and Yelp small dataset detailed in [33], each of them containing 1000 sequences from the large dataset. Both datasets are divided at random in a train set $T_{\text{train}}$ and $T_{\text{test}}$. The train set represents $3/4$ of the whole dataset while the remaining samples represent the test set. We use the hyperparameters reported in Table 4.

<table>
<thead>
<tr>
<th>NN model</th>
<th>LHTR$_1$</th>
<th>LHTR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sizes of the layers $\varphi$</td>
<td>[768,384,200,50,8,1]</td>
<td>[768,384,200,100]</td>
</tr>
<tr>
<td>Sizes of the layers $C_{\text{bulk}}^{\text{ext}}$</td>
<td>X</td>
<td>[100,50,8,1]</td>
</tr>
<tr>
<td>Sizes of the layers $C_{\text{ext}}^{\varphi}$</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>$\rho_3$</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

Table 4: Network architectures for Amazon small dataset and Yelp small dataset. The weight decay is set to $10^{-5}$, the learning rate is set to $5 \times 10^{-4}$, the number of epochs is set to 500 and the batch size is set to 64.

**BERT is not regularly varying.** In order to show that $X$ is not regularly varying, independence between $\|X\|$ and a margin of $\Theta(X)$ can be tested [14], which is easily done via correlation tests. Pearson correlation tests were run on the extreme samples of BERT and LHTR embeddings of Amazon small dataset and Yelp small dataset. The statistical tests were performed between all margins of $(\Theta(X_i))_{1 \leq i \leq n}$ and $(\|X_i\|)_{1 \leq i \leq n}$.
Figure 6: Histograms of the \( p \)-values for the non-correlation test between \((\Theta(X_i))_{1 \leq i \leq n}\) and \((\|X_i\|)_{1 \leq i \leq n}\) on embeddings provided by BERT (Figure 6a and Figure 6b) or LHTR (Figure 6c and Figure 6d).

Each histogram in Figure 6 displays the distribution of the \( p \)-values of the correlation tests between the margins \( X_j \) and the angle \( \Theta(X) \) for \( j \in \{1, \ldots, d\} \), in a given representation (BERT or LHTR) for a given dataset. For both Amazon small dataset and Yelp small dataset the distribution of the \( p \)-values is shifted towards larger values in the representation of LHTR than in BERT, which means that the correlations are weaker in the former representation than in the latter. This phenomenon is more pronounced with Yelp small dataset than with Amazon small dataset. Thus, in BERT representation, even the largest data points exhibit a non negligible correlation between the radius and the angle and the regular variation condition does not seem to be satisfied. As a consequence, in a classification setup such as binary sentiment analysis detailed in Section 4.2, classifiers trained on BERT embedding are not guaranteed to be scale invariant. In other words for a representation \( X \) of a sequence \( U \) with a given label \( Y \), the predicted label \( g(\lambda X) \) is not necessarily constant for varying values of \( \lambda \geq 1 \). Figure 7 illustrates this fact on a particular example taken from Yelp small dataset. The color (white or black respectively) indicates the predicted class (respectively \(-1\) and \(+1\)). For values of \( \lambda \) close to 1, the predicted class is \(-1\) but the prediction shifts to class \(+1\) for larger values of \( \lambda \).

Figure 7: Lack of scale invariance of the classifier trained on BERT: evolution of the predicted label \( g(\lambda X) \) from \(-1\) to \(+1\) for increasing values of \( \lambda \), for one particular example \( X \).

Scale invariance of LHTR. We provide here experimental evidence that LHTR’s classifier \( g^{ext} \) is scale invariant (as defined in Equation (1)). Figure 8 displays the predictions \( g^{ext}(\lambda Z_i) \) for increasing values of the scale factor \( \lambda \geq 1 \) and \( Z_i \) belonging to \( T_{test} \), the set of samples considered as extreme in
the learnt representation. For any such sample $Z$, the predicted label remains constant as $\lambda$ varies, i.e. it is scale invariant, $g^{\text{ext}}(\lambda Z) = g^{\text{ext}}(Z)$, for all $\lambda \geq 1$.

![Figure 8: Scale invariance of $g^{\text{ext}}$ trained on LHTR: evolution of the predicted label $g^{\text{ext}}(\lambda Z_i)$ (white or black for $-1/ + 1$) for increasing values of $\lambda$, for samples $Z_i$ from the extreme test set $T_{\text{test}}$ from Amazon small dataset (Figure 8a) and Yelp small dataset (Figure 8b).](image)

### B.6 Experimental settings (Classification): additional details

**Toy example.** For the toy example, we generate 3000 points distributed as a mixture of two normal distributions in dimension two. For training LHTR, the number of epochs is set to 100 with a dropout rate equal to 0.4, a batch size of 64 and a learning rate of $5 \times 10^{-4}$. The weight parameter $\rho_3$ in the loss function (Jensen-Shannon divergence from the target) is set to $10^{-3}$. Each component $\varphi$, $C_{\text{bulk}}$ and $C^{\text{ext}}$ is made of 3 fully connected layers, the sizes of which are reported in Table 5.

**Datasets.** For Amazon, we work with the video games subdataset from [http://jmcauley.ucsd.edu/data/amazon/](http://jmcauley.ucsd.edu/data/amazon/). For Yelp [58, 36], we work with 1,450,000 reviews after that can be found at [https://www.yelp.com/dataset](https://www.yelp.com/dataset).

<table>
<thead>
<tr>
<th>Layers’ sizes</th>
<th>LHTR</th>
<th>LHTR</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\varphi$</td>
<td>[2,4,2]</td>
<td>[2,4,2]</td>
</tr>
<tr>
<td>$C_{\text{bulk}}$</td>
<td>[2,8,1]</td>
<td>[2,8,1]</td>
</tr>
<tr>
<td>$C^{\text{ext}}$</td>
<td>[2,8,1]</td>
<td>[2,8,1]</td>
</tr>
</tbody>
</table>

Table 5: Sizes of the successive layers in each component of LHTR used in the toy example.

**BERT representation for text data.** We use BERT pretrained models and code from the library [Transformers](https://github.com/huggingface/transformers). All models were implemented using Pytorch and trained on a single Nvidia P100. The output of BERT is a $\mathbb{R}^{768}$ vector. All parameters of the models have been selected using the same grid search.

**Network architectures.** Tables 6 report the architectures (layers sizes) chosen for each component of the three algorithms considered for performance comparison (Section 4), respectively for the moderate and large datasets used in our experiments. We set $\rho_1 = (1 - \hat{P}(|Z| \geq |Z_{(\lfloor \kappa n \rfloor)}|))^{-1}$ and $\rho_2 = \hat{P}(|Z| \geq |Z_{(\lfloor \kappa n \rfloor)}|))^{-1}$.

<table>
<thead>
<tr>
<th>Sizes of the layers $\varphi$</th>
<th>NN model</th>
<th>LHTR $t$</th>
<th>LHTR</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathbb{R}^{768,384,200,50,8,1}$</td>
<td>[768,384,200,50,8,1]</td>
<td>[768,384,200,100]</td>
<td>[768,384,200,150]</td>
</tr>
<tr>
<td>Sizes of the layers of $C_{\text{bulk}}$</td>
<td>[150,75,8,1]</td>
<td>[100,50,8,1]</td>
<td>[150,75,8,1]</td>
</tr>
<tr>
<td>Sizes of the layers of $C^{\text{ext}}$</td>
<td>X</td>
<td>X</td>
<td>[150,75,8,1]</td>
</tr>
<tr>
<td>$\rho_3$</td>
<td>X</td>
<td>X</td>
<td>0.01</td>
</tr>
</tbody>
</table>

Table 6: Network architectures for Amazon dataset and Yelp dataset. The weight decay is set to $10^{-5}$, the learning rate is set to $1 \times 10^{-4}$, the number of epochs is set to 500 and the batch size is set to 256.

### B.7 Experiments for data generation

#### B.7.1 Experimental setting

As mentioned in Section 5.1, hyperparameters for dataset augmentation are detailed in Table 7. For
Sizes of the layers $\varphi$ | LHTR  
---|---
Sizes of the layers of $C_{\rho}^{bulk}$ | [768,384,200,150]  
Sizes of the layers of $C_{\rho}^{ext}$ | [150,75,8,1]  
$\rho_3$ | 0.01

Table 7: For Amazon and Yelp, we follow [57] the weight decay is set to $10^{-5}$, the learning rate is set to $1 \times 10^{-4}$, the number of epochs is set to 100 and the batch size is set to 256.

the Transformer Decoder we use 2 layers with 8 heads, the dimension of the key and value is set to 64 [54] and the inner dimension is set to 512. The architectures for the models proposed by Wei and Zou [56] and Kobayashi [32] are chosen according to the original papers. For a fair comparison with Kobayashi [32], we update the language model with a BERT model, the labels are embedded in $\mathbb{R}^{10}$ and fed to a single MLP layer (this dimension is chosen using the same procedure as in [15, 20]). The new model is trained using AdamW [37].

### B.7.2 Influence of the scaling factor on the linguistic content

Table 8 gathers some extreme sequences generated by GENELIEX for $\lambda$ ranging from 1 to 1.5. No major linguistic change appears when $\lambda$ varies. The generated sequences are grammatically correct and share the same polarity (positive or negative sentiment) as the input sequence. Note that for greater values of $\lambda$, a repetition phenomenon appears. The resulting sequences keep the label and polarity of the input sequence but repeat some words [28].

## C Extremes in Text

### Aim of the experiments

The aim of this section is double: first, to provide some intuition on what characterizes sequences falling in the extreme region of LHTR. Second, to investigate the hypothesis that extremes from LHTR are input sequences which tend to be harder to model than non extreme ones.

Regarding the first aim (i) Are there interpretable text features correlated with the extreme nature of a text sample?, since we characterize extremes by their norm in LHTR representation, in practice the question boils down to finding text features which are positively correlated with the norm of the text samples in LHTR, which we denote by $\|\varphi(X)\|$ and referred to as the ‘LHTR norm’ in the sequel. Preliminary investigations did not reveal semantic features (related to the meaning or the sentiment expressed in the sequence) displaying such correlation. However we have identified two features which are positively correlated both together and with the norm in LHTR, namely the sequence length $|U|$ as measured by the number of tokens of the input (recall that in our case an input sequence $U$ is a review composed of multiple sequences) and the norm of the input in BERT representation (‘BERT norm’, denoted by $\|X\|$).

As for the second question (ii) Are LHTR’s extremes harder to model? we consider the next token prediction loss [5] (‘LM loss’ in the sequel) obtained by training a language model on top of BERT. The next token prediction loss can be seen as a measure of hardness to model the input sequence. The question is thus to determine whether this prediction loss is correlated with the norm in LHTR (or in BERT, or with the sequence length).

### Results

Figure 9 displays pairwise scatterplots for the four considered variables on Yelp dataset (left) and Amazon dataset (right). These scatterplot suggest strong dependence for all pairs of variables. For a more quantitative assessment, Figure 10 displays the correlation matrices between the four quantities $\|\varphi(X)\|$, $\|X\|$, $|U|$ and ‘LM Loss’ described above on Amazon and Yelp datasets. Pearson and Spearman two-sided correlation tests are performed on all pairs of variables, both tests having as null hypothesis that the correlation between two variables is zero. For all tests, $p$-values are smaller than $10^{-16}$, therefore null hypotheses are rejected for all pairs.

These results prove that the four considered variables are indeed significantly positively correlated, which answers questions (i) and (ii) above.
Figure 9: Scatterplots of the four variables ‘BERT norm’, ‘LHTR norm’, ‘LM loss’ and ‘sequence length’ on Yelp dataset (top) and Amazon dataset (bottom).
very sloppy and slow service. when we arrived they told us to sit anywhere but all the tables were still dirty and haven’t been cleaned. they didn’t bother to ask if we wanted refills on our drinks. we needed an extra plate and didn’t get one so my nephew decides to go up to the counter and ask for one because he’s hungry. they gave our check when we were still eating. the list can go on and on. i wouldn’t recommend this place. go somewhere else for faster and better service. very disappointed

λ = 1.1

very sloppy and sluggish service. when we got there, they told us to sit anywhere but all the tables were empty full of dishes and were not cleaned at all. they didn’t bother to ask if our drinks would be added. we needed an extra dish and didn’t get one, so my cousin decided to go to the counter and ask one because he’s hungry. they were going to watch while we were still eating. the list could go on and on. i would not recommend this place. go elsewhere for faster and better service. very disappointed

λ = 1.2

services and survivors. when he got there, he told us we were sitting everywhere but all the tables were full of dishes and we didn’t wash everything. he never bothered to ask if our drinks would be added. we needed extra food and didn’t get one, so my brother decided to go to the locker and ask because he was thirsty. they want to watch it while we eat. the list can be continuous and active. i would not recommend this place. go elsewhere for faster and better service. very disappointed

λ = 1.3

visited today with my husband. we were in the firearms section. there were 3 employees in attendance with one customer. my husband asked a question and was ignored. he waited around for another 10 minutes or so. if it had been busy i could understand not receiving help. we left and went somewhere else for our purchases.

λ = 1.1

visited today with my husband. we were in the firearms section. together with one customer there were 3 employees. my husband asked and was ignored. waited about another 10 minutes. if it was busy, i would understand that i wouldn’t get help. we left and went somewhere else because of our purchases.

λ = 1.2

today she visited with her husband. we were in the gun department. there were 3 employees together with one customer. my husband asked and was ignored. waited another 10 minutes. if he was busy, i would understand that i would not receive help. we went and went somewhere else because of our shopping.

λ = 1.3

today, she went with her husband. we are in the gun department. there are 3 employees and one customer. my husband rejected me and ignored him. wait another minute. if he has a job at hand, i will understand that i will not get help. we went somewhere else because of our business.

Table 8: Sequences generated by GENELIEX for extreme embeddings implying label (sentiment polarity) invariance for generated Sequence. λ is the scale factor. Two first reviews are negatives, two last reviews are positive.
Figure 10: Non diagonal entries of the correlation matrices of the four variables ‘BERT norm’, ‘LHTR norm’, ‘LM loss’ and ‘sequence length’ for Yelp dataset (left) and Amazon dataset (right).

Figure 11: Histograms of the samples’ sequence length for Yelp dataset (Figure 11a and Figure 11b) and Amazon (Figure 11c and Figure 11d). The number of sequences in the bulk is approximately 3 times the number of extreme sequences for each dataset 10000 sequences are considered and extreme region contains approximately 3000 sequences.

Experimental conclusions  We summarize the empirical findings of this section:

1. An ‘extreme’ text sequence in LHTR representation is more likely to have a greater length (number of tokens) than a non extreme one.

2. Positive correlation between the BERT norm and the LHTR norm indicates that a large sample in the BERT representation is likely to have a large norm in the LHTR representation as well: the learnt representation LHTR taking BERT as input keeps invariant (in probability) the ordering implied by the norm.
3. A consequence of the two above points is that long sequences tend to have a large norm in BERT.

4. Extreme text samples (regarding the BERT norm or the LHTR norm) tend to be harder to model than non-extreme ones.

5. Since extreme texts are harder to model and also somewhat harder to classify in view of the BERT classification scores reported in Table I there is room for improvement in their analysis and it is no wonder that a method dedicated to extremes i.e. relying on EVT such as LHTR outperforms the baseline.