arXiv:2003.11593v1 [stat.ML] 25 Mar 2020

HEAVY-TAILED REPRESENTATIONS, TEXT POLARITY CLASSIFICATION & DATA AUGMENTATION

A PREPRINT

Hamid Jalalzai*
LTCI, Télécom Paris
Institut Polytechnique de Paris
hamid.jalalzai@telecom-paris.fr

Pierre Colombo*
IBM France
LTCI, Télécom Paris
Institut Polytechnique de Paris
pierre.colombo@telecom-paris.fr

Chloé Clavel
LTCI, Télécom Paris
Institut Polytechnique de Paris
chloe.clavel@telecom-paris.fr

Eric Gaussier
Univ. Grenoble Alpes, CNRS, Grenoble INP, LIG
eric.gaussier@imag.fr

Giovanna Varni
LTCI, Télécom Paris
Institut Polytechnique de Paris
giovanna.varni@telecom-paris.fr

Emmanuel Vignon
IBM France
emmanuel.vignon@fr.ibm.com

Anne Sabourin
LTCI, Télécom Paris
Institut Polytechnique de Paris
anne.sabourin@telecom-paris.fr

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 performance outperforms the baseline. This classifier exhibits a *scale invariance* property which we leverage by introducing a novel text generation method for label preserving dataset augmentation. Numerical experiments on synthetic and real text data demonstrate the relevance of the proposed framework and confirm that this method generates meaningful sentences with controllable attribute, *e.g.* positive or negative sentiment.

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 Peters et al. (2018); Radford et al. (2018); Devlin et al. (2018) with demonstrated efficiency for the considered tasks when learnt on massive datasets. In particular, these embeddings are commonly used to perform downstream tasks such as text classification or generation. However, how

*Both authors contributed equally

to use these representations as an input of learning algorithms which are well understood mathematically but rely on specific distributional assumptions remains a largely unexplored topic. One example of such a framework is that of multivariate extreme value analysis, based on extreme value theory (EVT), which focus is on the tails of the variables of interest. EVT is valid under a regularity assumption which amounts to a homogeneity property above large thresholds: The tail behaviour of the considered variables must be well approximated by a power law, see Section 2 for a rigorous statement. A major advantage of this framework in the case of labeled data Jalalzai et al. (2018) is that classification on the tail regions of the explanatory variable $x \in \mathbb{R}^d$ of the kind $\{\|x \geq t\|\}$, for a large threshold t, may be performed using the angle $\Theta(x) = \|x\|^{-1}x$ only, see Figure 1. The main idea behind the present paper is to take advantage of this scale invariance for two tasks regarding sentiment analysis of text data: *(i)* Improved classification of a small but non negligible proportion of the data (namely, 25% in our experiments), *(ii)* Label preserving data augmentation, using the fact that the most probable label of an input x is unchanged when multiplying x by a scalar $\lambda > 1$.

How to use EVT in a machine learning framework has received increasing attention in the past few years. Learning tasks considered so far include anomaly detection Roberts (1999, 2000); Clifton et al. (2011); Goix et al. (2016); Thomas et al. (2017), anomaly clustering Chiapino et al. (2019a), unsupervised learning Goix et al. (2015), online learning Carpentier & Valko (2014); Achab et al. (2017), dimension reduction and support identification Goix et al. (2017); Chiapino & Sabourin (2016); Chiapino et al. (2019b). The present paper builds upon the methodological framework proposed by Jalalzai et al. (2018) for classification in extreme regions. The goal of Jalalzai et al. (2018) is to improve the performance of classifiers $\widehat{g}(x)$ 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 $\widehat{g}$ 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. (2018) 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 Jalalzai et al. (2018)'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 *Learning a Heavy Tailed Representation* (in short **`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 Goodfellow et al. (2016). In addition, we obtain a novel data augmentation mechanism **`GENELIEX`** which takes advantage of the scale invariance properties of Z to generate synthetic sentences that keep invariant the attribute of the original sentence.

Label preserving data augmentation is an effective solution to the data scarcity problem and is an efficient pre-processing step for moderate dimensional datasets Wang & Perez (2017); Wei & Zou (2019). 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: $y\big(h(x)\big) = y(x)$ Ratner et al. (2017). The dominant approaches for text data augmentation rely on word level transformations such as synonym replacement, slot filling, swap deletion Wei & Zou (2019) using external resources such as wordnet Miller (1995). Linguistic based approaches can also be combined with vectorial representations provided by language models Kobayashi (2018). 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 $y\big(h(x)\big)$ is unknown as soon as $h(x)$ does not belong to the training set, we address this issue by learning both an embedding φ and a classifier g satisfying a relaxed version of the problem above mentioned, namely $\forall \lambda \geq 1$

$$g\big(h_\lambda(\varphi(x))\big) = g\big(\varphi(x)\big). \tag{1}$$

For mathematical reasons which will appear clearly in Section 2.2, h_λ is chosen as the homothety with scale factor λ, $h_\lambda(x) = \lambda x$. In this paper, we work with output vectors issued by BERT Devlin et al. (2018). 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 C.2) Besides, there is no reason why a classifier g trained on such embedding would be scale invariant, *i.e.* would satisfy for a given sentence u, embedded as x, $g(h_\lambda(x)) = g(x)\ \forall \lambda \geq 1$. On the classification task, we demonstrate on four 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 sentences while preserving labels.

The rest of this paper is organized as follows. Section 2 introduces the necessary background in multivariate extremes and adversarial learning. 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. Additional comments and further experimental results are provided in the supplementary material.

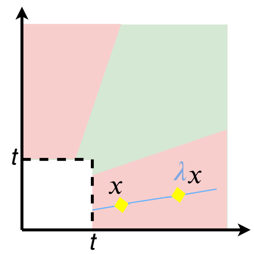


Figure 1: Example of angular classifier g dedicated to extreme samples $\{x, \|x\|_\infty \geq t\}$ in $\mathbb{R}^2_+$. The red and green truncated cones are the regions respectively labeled as $+1$ and -1 by g.

2 Background

2.1 Extreme values, heavy tails and regular variation

Extreme value analysis is a branch of statistics which 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 $x > 0$: $t\mathbb{P}\{X/b(t) > x\} \xrightarrow[t\to\infty]{} x^{-\alpha}$. 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. Then X is *standard multivariate regularly varying* if there exists a positive Radon measure μ on $[0, \infty]^d \backslash \{0\}$

$$t\mathbb{P}\left\{t^{-1}X \in A\right\} \xrightarrow[t\to\infty]{} \mu(A), \tag{2}$$

for any Borelian set $A \subset [0, \infty]^d$ which is bounded away from 0 and such that the limit measure μ of the boundary ∂A is zero. For a complete introduction to the theory of Regular Variation, the reader may refer to Resnick (2013). The measure μ may be understood as the limit distribution of tail events. In the standard case (2), μ 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 μ using the largest available data.

2.2 Classification in Extreme Regions

We now recall the classification setup for extremes as introduced in Jalalzai et al. (2018). Let $(X, Y) \in \mathbb{R}^d_+ \times \{-1, 1\}$ be a random pair. Jalalzai et al. (2018) assume standard regular variation for both classes, that is $t\mathbb{P}\{X \in tA \mid Y = \pm 1\} \to \mu_\pm(A)$, 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,

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

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 μ, under additional mild regularity assumptions concerning the regression function, namely uniform convergence to the limit at infinity, one can prove the following result (see Jalalzai et al. (2018), Theorem 1): there exists a classifier $g^\star_\infty$ depending on the pseudo-angle $\Theta(x) = \|x\|^{-1}x$ only, that is $g^\star_\infty(x) = g^\star_\infty\big(\Theta(x)\big)$, which is asymptotically optimal in terms of classification risk, *i.e.* $L_\infty(g^\star_\infty) = \inf_{g \text{ measurable}} L_\infty(g)$.

Notice that for $x \in \mathbb{R}^d_+ \setminus \{0\}$, the angle $\Theta(x)$ belongs to the positive orthant of the unit sphere, denoted by S in the sequel. 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 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.* $g(x) = g(\Theta(x)) = g(\Theta(\lambda x)), \forall x \in \{x, \|x\| \geq t\}$). The angular structure of the optimal classifier $g^\star_\infty$ is the basis for the following ERM strategy using the most extreme points of a dataset. Let $\mathcal{G}_S$ be a class of angular classifiers defined on the sphere S with finite VC dimension $V_{\mathcal{G}_S} < \infty$. By extension, for any $x \in \mathbb{R}^d_+$ and $g \in \mathcal{G}_S$, $g(x) = g\big(\Theta(x)\big) \in \{-1, 1\}$. Given a training dataset $\{(X_i, Y_i)\}_{i=1}^n$ made of n *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 $\widehat{L}_k(g) = \frac{1}{k}\sum_{i=1}^k \mathbf{1}\{Y_{(i)} \neq g(\Theta(X_{(i)}))\}$ is an empirical version of the risk $L_{t(k)}(g)$ as defined in (3) where $t(k)$ is a $(1 - k/n)$-quantile of the norm, $\mathbb{P}\{\|X\| > t(k)\} = k/n$. Selection of k is a bias-variance compromise, see Appendix B for further discussion. The strategy promoted by Jalalzai et al. (2018) is to use the solution of the ERM problem, $\widehat{g}_k = \arg\min_{g\in\mathcal{G}_S} \widehat{L}_k(g)$, for classification in the extreme region $\{x \in \mathbb{R}^d_+ : \|x\| > t(k)\}$. The following result provides guarantees concerning the excess risk of $\widehat{g}_k$ compared with the Bayes risk above level $t = t(k)$, $L^\star_t = \inf_{g \text{ measurable}} L_t(g)$.

Theorem 1 *(Jalalzai et al. (2018), Theorem 2) If each class satisfies the regular variation assumption (2), under an additional regularity assumption concerning the regression function $\eta(x) = \mathbb{P}\{Y = +1 \mid x\}$ (see Equation (4) in Appendix B.3), for $\delta \in (0, 1)$, $\forall n \geq 1$, it holds with probability larger than $1 - \delta$ that*

$$L_{t(k)}(\widehat{g}_k) - L^\star_{t(k)} \leq \frac{1}{\sqrt{k}}\Big(\sqrt{2(1-k/n)\log(2/\delta)} \; + C\sqrt{V_{\mathcal{G}_S}\log(1/\delta)}\Big) + \frac{1}{k}\left(5 + 2\log(1/\delta) + \sqrt{\log(1/\delta)}(C\sqrt{V_{\mathcal{G}_S}} + \sqrt{2})\right) + \left\{\inf_{g\in\mathcal{G}_S} L_{t(k)}(g) - L^\star_{t(k)}\right\},$$

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 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.

2.3 Adversarial learning

Adversarial networks, introduced in Goodfellow et al. (2014), 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_{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_{input}. In short, the problem is a minmax game with value function $V(D, G)$

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim P_{\text{input}}}[\log D(x)] + \mathbb{E}_{z\sim P_G}[\log\big(1 - D(G(z))\big)].$$

Auto-encoders and derivations Goodfellow et al. (2016); Laforgue et al. (2018); Fard et al. (2018) 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 Makhzani et al. (2015)) 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.

3 Heavy-tailed sentence embeddings

3.1 Learning a heavy-tailed representation

We now introduce a novel algorithm *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 **GENELIEX** detailed in Section 3.2. **LHTR** proceeds by training an encoding function φ in such a way that *(i)* the marginal distribution $q(z)$ of the code Z be close to a user-specified heavy tailed target 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 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^{ext} on the extreme region of the latent space on the one hand, and g^{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 κ) 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^{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^{bulk} only serves as an intermediate step to learn an adequate representation φ.

Algorithm 1 LHTR

INPUT: Weighting coef. $\rho_1, \rho_2, \rho_3 > 0$, Training dataset $\mathcal{D}_n = \{(X_1, Y_1), \ \ldots, \ (X_n, Y_n)\}$, batch size m, proportion of extremes κ, heavy tailed prior P_Z.
Initialization: parameters $(\tau, \theta, \theta', \gamma)$ of the encoder φ_τ, classifiers C_θ^{ext}, $C_{\theta'}^{bulk}$ and discriminator D_γ
Optimization:
while $(\tau, \theta, \theta', \gamma)$ not converged **do**
Sample $\{(X_1, Y_1) \ldots, (X_m, Y_m)\}$ from $\mathcal{D}_n$ and define $\tilde{Z}_i = \varphi(X_i)$, $i \leq m$.
Sample $\{Z_1, \ldots, Z_m\}$ from the prior P_Z.
Update γ by ascending:

$$\frac{\rho_3}{m}\sum_{i=1}^m \log D_\gamma(Z_i) + \log(1 - D_\gamma(\tilde{Z}_i)).$$

Sort $\{\tilde{Z}_i\}_{i\in\{1,\ldots,m\}}$ by decreasing order of magnitude $||\tilde{Z}_{(1)}|| \geq \ldots \geq ||\tilde{Z}_{(m)}||$.
Update θ by descending:

$$\mathcal{L}^{\text{ext}}(\theta, \tau) \overset{\text{def}}{=} \frac{\rho_1}{\lfloor \kappa m \rfloor}\sum_{i=1}^{\lfloor \kappa m \rfloor} \ell\big(Y_{(i)}, C_\theta^{\text{ext}}(\tilde{Z}_{(i)})\big).$$

Update θ' by descending:

$$\mathcal{L}^{\text{bulk}}(\theta', \tau) \overset{\text{def}}{=} \frac{\rho_2}{m - \lfloor \kappa m \rfloor}\sum_{i=\lfloor \kappa m \rfloor + 1}^{m} \ell\big(Y_{(i)}, C_{\theta'}^{\text{bulk}}(\tilde{Z}_{(i)})\big).$$

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

Algorithm 2 GENELIEX: training step

INPUT: input of LHTR, $\mathcal{D}_{g_n} = \{U_1, \ \ldots, \ U_n\}$
Initialization: parameters of φ_τ, C_θ^{ext}, $C_{\theta'}^{\text{bulk}}$, D_γ and decoder G_ψ^{ext}
Optimization:
φ, C^{ext}, C^{bulk} = LHTR($\rho_1, \rho_2, \rho_3, \mathcal{D}_n, \kappa, m$)
while ψ not converged **do**
Sample $\{U_1 \ldots, U_m\}$ from the training set $\mathcal{D}_{g_n}$ 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 ψ by descending:

$$\mathcal{L}_g^{\text{ext}}(\psi) \overset{\text{def}}{=} \frac{\rho_1}{\lfloor \kappa m \rfloor}\sum_{i=1}^{\lfloor \kappa m \rfloor} \ell_{gen.}\big(U_{(i)}, G_\psi^{\text{ext}}(\tilde{Z}_{(i)})\big).$$

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 $\|\tilde{Z}_{(1)}\| \geq \ldots \|\tilde{Z}_{(k)}\| \geq \ldots \geq \|\tilde{Z}_{(n)}\|$.
OUTPUT: encoder φ, decoder G^{ext} applicable on the region $\{x : \|\varphi(x)\| \geq \|\tilde{Z}_{(\lfloor \kappa n \rfloor)}\|\}$

Remark 1 Recall from Section 2.2 that the optimal classifier in the extreme region as $t \to \infty$ depends on the angular component $\theta(x)$ 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 our preliminary experiments (not shown here), the resulting classifier was less performant 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

$$R(\varphi, g^{\text{ext}}, g^{\text{bulk}}) \overset{\text{def}}{=} \rho_1 \mathbb{P}\left\{Y \neq g^{\text{ext}}(Z), \|Z\| \geq t\right\} + \rho_2 \mathbb{P}\left\{Y \neq g^{\text{bulk}}(Z), \|Z\| < t\right\} + \rho_3 \mathfrak{D}(q(z), p(z)),$$

where $Z = \varphi(X)$, $\mathfrak{D}$ is the Jensen-Shannon distance between the heavy tailed target distribution p and the code distribution q, and ρ_1, ρ_2, ρ_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 $\widehat{L}(q,p) = \sup_{D\in\Gamma} \widehat{L}(q,p,D)$, with $\widehat{L}(q,p,D) = \frac{1}{m}\sum_{i=1}^{m} \log D(Z_i) + \log\big(1 - D(\tilde{Z}_i)\big)$, where Γ is a wide class of discriminant functions valued in $[0,1]$, and where independent samples $Z_i, \tilde{Z}_i$ are respectively sampled from the target distribution and the code distribution q. The classifiers g^{ext}, g^{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 $\inf_{C^{\text{ext}},C^{\text{bulk}},\varphi} \sup_D \widehat{R}(\varphi, C^{\text{ext}}, C^{\text{bulk}}, D)$ with

$$\widehat{R}(\varphi, C^{\text{ext}}, C^{\text{bulk}}, D) = \frac{\rho_1}{k}\sum_{i=1}^{k} \ell(Y_{(i)}, C^{\text{ext}}(Z_{(i)})) \; + \frac{\rho_2}{n-k}\sum_{i=k+1}^{n-k} \ell(Y_{(i)}, C^{\text{bulk}}(Z_{(i)})) + \rho_3\, \hat{L}(q,p,D),$$

where $\{Z_{(i)} = \varphi(X_{(i)}), i = 1,\dots,n\}$ 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 $\ell(y, C(x)) = -(y \log C(x) + (1-y)\log(1 - C(x)), y \in \{0,1\}$ is the negative log-likelihood of the dicriminant function $C(x) \in (0,1)$. **`LHTR`** is summarized in Algorithm 1 and its workflow is illustrated in Appendix A.

3.2 A heavy-tailed representation for dataset augmentation

We now introduce **`GENELIEX`** (Generating Label Invariant sentences 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 Sutskever et al. (2014). It trains a Transformer Decoder Vaswani et al. (2017) G^{ext} on the extreme regions.

For an input sentence $U = (u_1,\dots,u_T)$ 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^{ext} M sentences U'_j where $j \in \{1,\dots,M\}$. The M decoded sequences correspond to the codes $\{\lambda_j Z, j \in \{1,\dots,M\}\}$ 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 the decoder generates either a stop symbol or the length of the generated sequence reaches the maximum sequence length ($T_{\max}$).

To train the decoder $G^{\text{ext}} : \mathbb{R}^{d'} \to \big[1,\dots,|\mathcal{V}|\big]^{T_{\max}}$ where $\mathcal{V}$ is the vocabulary on the extreme regions, **`GENELIEX`** requires an additional dataset $\mathcal{D}_{g_n} = (U_1,\dots,U_n)$ (not necessarily labeled) with associated representation *via* BERT $(X_{U,1},\dots,X_{U,n})$. Learning is carried out by optimising the classical negative log-likelihood of individual tokens ℓ_{gen}. The latter is defined as $\ell_{gen}\big(U, G^{\text{ext}}(\varphi(X))\big) \overset{\text{def}}{=} \sum_{t=1}^{T_{\max}} \sum_{v\in\mathcal{V}} \mathbb{1}\{u_t = v\} \log\big(p_{v,t}\big)$, where $p_{v,t}$ is the probability predicted by G^{ext} that the t^{th} word is equal to v. A detailed description of the training step of **`GENELIEX`** is provided in Algorithm 2, see also Appendix A for an illustrative diagram.

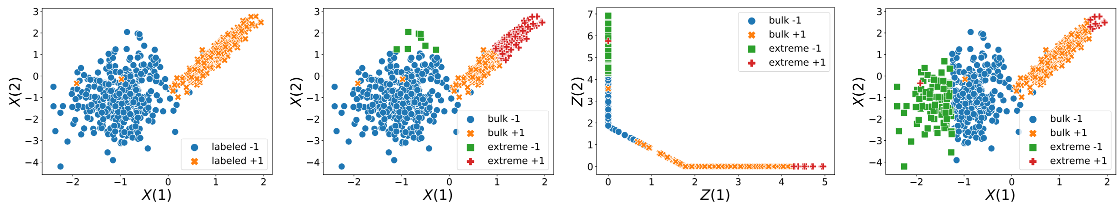
Remark 2 Note that the proposed method only augments data on the extreme regions. A general data augmentation algorithm can be obtained by combining this approach with any other algorithm on the original input data X whose latent code $Z = \varphi(X_U)$ does not lie in the extreme regions.

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 $t = \|\tilde{Z}_{(\lfloor \kappa n \rfloor)}$ as returned by **`LHTR`**. We denote by $\mathcal{T}_{\text{test}}$ and $\mathcal{T}_{\text{train}}$ 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 Apendix C.1 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 δ. This distribution is widely used in the context of extreme values analysis Chiapino et al. (2019b); Thomas et al. (2017); Goix et al. (2016) and should not be confused with the classical logistic distribution which is well known in the machine learning community.

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 φ 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).



(a) (b) (c) (d)

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.

Labeled samples are simulated from a Gaussian mixture distribution with two components of identical weight. The label indicates the component from which the point is generated. **`LHTR`** is trained on 2250 examples and a testing set of size 750 is shown in Figure 2. The testing samples in the input space (Figure 2a) are mapped onto the latent space *via* φ (Figure 2c) In Figure 2b, the extreme raw observations are selected according to their norm after a component-wise standardisation of X_i, refer to the 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. Here the standardisation step is unnecessary because all components of the code are identically distributed due to our prior choice. Figure 2d is similar to Figure 2b except for the selection of extremes is performed in the latent space as in Figure 2c.

On this toy example, the GAN 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 meaningful selection of extreme samples than the input representation. Indeed, the former representation ensures class balance among extremes while the latter yields imbalanced classes. Namely, selection of extremes based on the heavy tailed representation yields 108 extreme points in the test set, among which 23 points from class 1 while selection in the input space yields 182 extremes in the test set among which only 9 examples from class 0. Another useful feature of the heavy tailed representation is that it separates the two classes in the latent space, thus facilitating classification.

4.2 Application to positive *vs.* negative classification of sentences

In this section, we investigate the relevance 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, following Jalalzai et al. (2018). 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 the raw input (BERT). Regarding the classification task, the purpose is to predict a binary label attributed to a text review.

Datasets: Our analysis relies on four text datasets: *Amazon dataset 1*, *Yelp dataset 1*, *Amazon dataset 2* and *Yelp dataset 2*. *Amazon dataset 2* is a large dataset introduced in McAuley & Leskovec (2013) containing 231,780 reviews[2]. *Yelp dataset 2* is another commonly used large dataset Yu et al. (2014); Liu et al. (2015) containing 1,450,000 reviews after preprocessing [3]. Reviews' ratings are used to obtain labels. Reviews with a rating greater than or equal to $^4/_5$ are labeled as $+1$, while those with a rating smaller or equal to $^2/_5$ are labeled as -1. The gap in reviews' rating is to avoid any potential overlap between labels of different contents. *Amazon dataset 1* and *Yelp dataset 1* are balanced binary labeled datasets introduced by Kotzias et al. (2015) and commonly used in the NLP community. Both datasets are composed of 1000 sentences extracted from positive and negative reviews of the corresponding large dataset and manually labeled positive or negative by four annotators. The classification task is known to be easier on those selected data.
Table 1 summarizes the sizes of the four datasets together with the proportion of training and testing data used in our experiments. Note that each sample of *Amazon dataset 1* and *Yelp dataset 1* is a sentence while each sample of *Amazon dataset 2* and *Yelp dataset 2* is is a review, potentially composed of several sentences.
Performance comparison: We compare the performance of three classifiers. The baseline **`NN model`** is a MLP

[2]We work with the video games subdataset from `http://jmcauley.ucsd.edu/data/amazon/`.
[3]Data can be found at `https://www.yelp.com/dataset`.

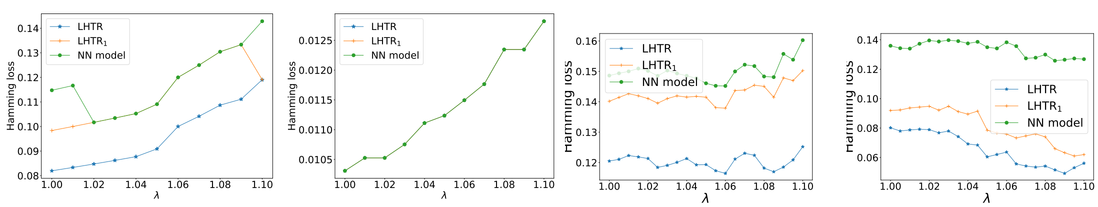
	Amazon		*Yelp*	
	dataset 1	dataset 2	dataset 1	dataset 2
$\lvert\mathcal{D}_n^{Train}\rvert$	750	210k	750	1400k
$\lvert\mathcal{D}_n^{Test}\rvert$	250	22k	250	22k
$\lvert\mathcal{T}_{\text{test}}\rvert$	61	6.9k	97	6.1k

Table 1: Sizes of the train $\mathcal{D}_n^{Train}$, test $\mathcal{D}_n^{Test}$ and the extreme test set $\mathcal{T}_{\text{test}}$. The extreme train tests constitute $1/4$ of the full train tests. These datasets are used to validate **`LHTR`**.

trained on BERT. The second classifier **`LHTR`**$_1$ is a variant of **`LHTR`** where a single MLP C is trained on the output of the encoder φ, using all the available data, both extreme and non extreme ones, so that $C^{\text{ext}} = C^{\text{bulk}}$ using the notations from Section 3. A diagram for this variant is provided in Appendix A. The third classifier results from the approach we promote, which is to use the extreme classifier C^{ext} trained by **`LHTR`** on the tail region. For simplicity, we denote by **`LHTR`** this classifier. All classifiers take the same training inputs and have identical structure, see Appendix C.1 for additional details concerning the network settings. Since the core contribution of our method concerns the distributional tails, the performance (as defined by the Hamming loss) is measured on the extreme test samples, *i.e.* those which norm exceeds the extreme threshold determined by **`LHTR`** as the $(1-\kappa)$ empirical quantile of the norm on the training set. Comparing **`LHTR`**$_1$ with **`NN model`** assesses the relevance of working with heavy-tailed embeddings. Since **`LHTR`**$_1$ is obtained by using **`LHTR`** with $C^{\text{ext}} = C^{\text{bulk}}$, comparing **`LHTR`**$_1$ with **`LHTR`** validates the use of two separate classifiers so that extremes are handled in a specific manner.

Performance metric: To illustrate the generalization ability of the proposed classifier in the extreme regions we consider nested subsets of the extreme test set $\mathcal{T}_{\text{test}}$, $\mathcal{T}^{\lambda} = \{z \in \mathcal{T}_{\text{test}}, \|z\| \geq \lambda t\}$, $\lambda \geq 1$. Note that for all factor $\lambda \geq 1$, $\mathcal{T}^{\lambda} \subseteq \mathcal{T}_{\text{test}}$. The greater λ, the fewer the samples retained for evaluation and the greater their norms.

Results: Figure 3 gathers the results obtained by the three considered classifiers on the four datasets mentioned above. The three competing methods obtain the exact same scores on *Yelp dataset 1* (Figure 3b), which may possibly be explained by the fact that the classification task is known to be easy on this dataset. Indeed, the two classes are well separated in the three representations except for rare counter-examples, leading to identical performances. On the three other datasets (Figures 3a, 3c, 3d), **`LHTR`**$_1$ outperforms the baseline **`NN model`**, even though the improvement is moderate on the small size dataset *Amazon dataset 1*. 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`** that we propose. This demonstrates the importance of working with two separate classifiers.



(a) (b) (c) (d)

Figure 3: Hamming loss of **`LHTR`**, **`LHTR`**$_1$ and **`NN model`** on the extreme test set $\{x \in \mathcal{T}, ||x|| \geq \lambda t\}$ for increasing values of λ (X-axis), on several datasets: (3a) *Amazon dataset 1*, (3b) *Yelp dataset 1*, (3c) *Amazon dataset 2*, (3d) *Yelp dataset 2*. Note that in Figure 3b the classifiers' performance are equal and losses are overlapping.

Scale invariance: On all datasets, the extreme classifier g^{ext} verifies Equation (1) for each sample of the test set, $g^{\text{ext}}(\lambda Z) = g^{\text{ext}}(Z)$ with λ ranging from 1 to 20. This demonstrates that g^{ext} is scale invariant 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 when varying the value of λ. Therefore, none of them are scale invariant. Additional experimental details are gathered in Appendix C.2.

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 & Zou (2019) and Kobayashi (2018). 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 proposed by Kobayashi (2018) with a BERT language model[4] (see Appendix

[4]We use the library Tansformers Wolf et al. (2019)

Model	*Amazon 2*			
	Medium		Large	
	F1	dist1/dist2	F1	dist1/dist2
Raw Data	84.0	X	93.3	X
Kobayashi (2018)	85.0	0.10/0.47	92.9	0.14/0.53
Wei & Zou (2019)	85.2	0.11/0.50	93.2	0.14/0.54
GENELIEX	**86.3**	**0.14/0.52**	**94.0**	**0.18/0.58**

Model	*Yelp 2*			
	Medium		Large	
	F1	dist1/dist2	F1	dist1/dist2
Raw Data	86.7	X	94.1	X
Kobayashi (2018)	87.0	0.15/0.53	94.0	0.14/0.58
Wei & Zou (2019)	87.0	0.15/0.52	**94.2**	0.16/0.59
GENELIEX	**88.4**	**0.18/0.62**	**94.2**	**0.16/0.60**

Table 2: Quantitative Evaluation. Algorithms are compared according to **C3** and **C4**. dist1 and dist2 stand respectively for distinct 1 and 2 and measure the diversity of generated sentences in terms of unigrams and bigrams. F1 is the F1-score achieved by a FastText classifier trained on an augmented labelled training set.

C.3 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 (**C1**, **C2**, **C3**,**C4**). **C1** measures Cohesion Crossley & McNamara (2010) (*Are the generated sentences grammatically and semantically consistent?*). **C2** (named Sent. in Table 3) evaluates label conservation (*Does the expressed sentiment in the generated sentence match the sentiment of the input sentence?*). **C3** measures the diversity Li et al. (2015) (corresponding to dist1 or dist2 in Table 3[5]) of the sentences (*Does the augmented dataset contain diverse sentences?*). Augmenting the training set with very diverse sentences can lead to better classification performance. **C4** measures the improvement in terms of F1 score when training a classifier (fastText Joulin et al. (2016)) on the augmented training set (*Does the augmented dataset improve classification performance?*).

Datasets. We evaluate our data augmentation algorithm using two data sets, a medium one and a large one (see Silfverberg et al. (2017)) which consist respectively of 1k and 10k labeled samples. In both cases, we have access to $\mathcal{D}_{g_n}$ a dataset of size 80k of unlabeled samples. Datasets are randomly sampled from *Amazon dataset 2* and *Yelp dataset 2*.

Experiment description. We augment extreme regions of each dataset according to three algorithms: **GENELIEX** (with scaling factor λ ranging from 1 to 1.5), Kobayashi (2018), and Wei & Zou (2019). For each sentence considered as extreme in the train set, 10 new sentences are generated using each algorithm. Additional details are given in Appendix C.3. For experiment **C4** the test set contains 10000 sentences.

5.2 Automatic measures

The results of **C3** and **C4** evaluation are reported in Table 2. Augmented data with **GENELIEX** are more diverse than the one augmented with Kobayashi (2018) and Wei & Zou (2019). 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 sentences in the extreme regions and higher F1 score: the more diverse the augmented dataset, the higher the F1 score. More diverse sequences are thus more likely to lead to better improvement on downstream tasks (*e.g.* classification).

5.3 Perceptive Measures

To evaluate **C1**, **C2**, three turkers were asked to annotate the cohesion and the sentiment of 100 generated sentences for each algorithm and for the raw data. F1 scores of this evaluation are reported in Table 3. Grammar evaluation confirms the findings of Wei & Zou (2019) showing that random swaps and deletions do not always maintain the cohesion of the sentence. In contrast, **GENELIEX** and Kobayashi (2018) which use vectorial representations produce more coherent sentences. Concerning sentiment label preservation, on Yelp 2, **GENELIEX** achieves the highest score which confirms the observed improvement reported in Table 2. On Amazon 2, turker annotations obtain a lower F1-score using data

[5] *Distn* is obtained by calculating the number of distinct n-grams divided by the total number of generated tokens to avoid favoring long sentences.

from **GENELIEX** than from Kobayashi (2018). This does not correlate with what is observed in Table 2 and may be explained by a lower Krippendorff Alpha[6] on Amazon ($\alpha = 0.20$) than on Yelp ($\alpha = 0.57$) .

Model	Medium			
	Amazon 2		*Yelp 2*	
	Sent.	Cohesion	Sent.	Cohesion
Raw Data	83.6	78.3	80.6	0.71
Kobayashi (2018)	**80.0**	**84.2**	82.9	0.72
Wei & Zou (2019)	69.0	67.4	80.0	0.60
GENELIEX	78.4	73.2	**85.7**	**0.77**

Table 3: Qualitative evaluation. Qualitative evaluation is performed on Amazon Mechanical Turk. Sent. stands for sentiment label preservation. Each sentence is annotated by three turkers. 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.

[6] Krippendorff Alpha is a measure for determining inter-rater reliability. Values vary from 0 to 1, where 0 is perfect disagreement and 1 is perfect agreement.

References

Achab, M., Clémençon, S., Garivier, A., Sabourin, A., and Vernade, C. Max k-armed bandit: On the extremehunter algorithm and beyond. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 389–404. Springer, 2017.

Carpentier, A. and Valko, M. Extreme bandits. In *Advances in Neural Information Processing Systems 27*, pp. 1089–1097. Curran Associates, Inc., 2014.

Chiapino, M. and Sabourin, A. Feature clustering for extreme events analysis, with application to extreme stream-flow data. In *International Workshop on New Frontiers in Mining Complex Patterns*, pp. 132–147. Springer, 2016.

Chiapino, M., Clémençon, S., Feuillard, V., and Sabourin, A. A multivariate extreme value theory approach to anomaly clustering and visualization. *Computational Statistics*, pp. 1–22, 2019a.

Chiapino, M., Sabourin, A., and Segers, J. Identifying groups of variables with the potential of being large simultaneously. *Extremes*, 22(2):193–222, 2019b.

Clifton, D. A., Hugueny, S., and Tarassenko, L. Novelty detection with multivariate extreme value statistics. *J Signal Process Syst.*, 65:371–389, 2011.

Coles, S. G. and Tawn, J. A. Statistical methods for multivariate extremes: an application to structural design. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 43(1):1–31, 1994.

Crossley, S. and McNamara, D. Cohesion, coherence, and expert evaluations of writing proficiency. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 32, 2010.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Fard, M. M., Thonet, T., and Gaussier, E. Deep k-means: Jointly clustering with k-means and learning representations. *arXiv preprint arXiv:1806.10069*, 2018.

Goix, N., Sabourin, A., and Clémençon, S. Learning the dependence structure of rare events: a non-asymptotic study. In *Conference on Learning Theory*, pp. 843–860, 2015.

Goix, N., Sabourin, A., and Clémençon, S. Sparse representation of multivariate extremes with applications to anomaly ranking. In *Artificial Intelligence and Statistics*, pp. 75–83, 2016.

Goix, N., Sabourin, A., and Clémençon, S. Sparse representation of multivariate extremes with applications to anomaly detection. *Journal of Multivariate Analysis*, 161:12–31, 2017.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

Goodfellow, I., Bengio, Y., and Courville, A. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

Holtzman, A., Buys, J., Forbes, M., and Choi, Y. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*, 2019.

Jalalzai, H., Clémençon, S., and Sabourin, A. On binary classification in extreme regions. In *Advances in Neural Information Processing Systems*, pp. 3092–3100, 2018.

Joulin, A., Grave, E., Bojanowski, P., and Mikolov, T. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*, 2016.

Kobayashi, S. Contextual augmentation: Data augmentation by words with paradigmatic relations. *arXiv preprint arXiv:1805.06201*, 2018.

Kotzias, D., Denil, M., De Freitas, N., and Smyth, P. From group to individual labels using deep features. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 597–606. ACM, 2015.

Laforgue, P., Clémençon, S., and d'Alché Buc, F. Autoencoding any data through kernel autoencoders. *arXiv preprint arXiv:1805.11028*, 2018.

Li, J., Galley, M., Brockett, C., Gao, J., and Dolan, B. A diversity-promoting objective function for neural conversation models. *arXiv preprint arXiv:1510.03055*, 2015.

Liu, J., Shang, J., Wang, C., Ren, X., and Han, J. Mining quality phrases from massive text corpora. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, pp. 1729–1744. ACM, 2015.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., and Frey, B. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.

McAuley, J. and Leskovec, J. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM conference on Recommender systems*, pp. 165–172. ACM, 2013.

Miller, G. A. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. Deep contextualized word representations. In *Proc. of NAACL*, 2018.

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf*, 2018.

Ratner, A. J., Ehrenberg, H., Hussain, Z., Dunnmon, J., and Ré, C. Learning to compose domain-specific transformations for data augmentation. In *Advances in neural information processing systems*, pp. 3236–3246, 2017.

Resnick, S. I. *Extreme values, regular variation and point processes*. Springer, 2013.

Roberts, S. Novelty detection using extreme value statistics. *IEE P-VIS IMAGE SIGN*, 146:124–129, Jun 1999.

Roberts, S. Extreme value statistics for novelty detection in biomedical data processing. *IEE P-SCI MEAS TECH*, 147: 363–367, 2000.

Silfverberg, M., Wiemerslage, A., Liu, L., and Mao, L. J. Data augmentation for morphological reinflection. *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, pp. 90–99, 2017.

Stephenson, A. Simulating multivariate extreme value distributions of logistic type. *Extremes*, 6(1):49–59, 2003.

Sutskever, I., Vinyals, O., and Le, Q. V. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pp. 3104–3112, 2014.

Thomas, A., Clémençon, S., Gramfort, A., and Sabourin, A. Anomaly detection in extreme regions via empirical mv-sets on the sphere. In *AISTATS*, pp. 1011–1019, 2017.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

Wang, J. and Perez, L. The effectiveness of data augmentation in image classification using deep learning. *Convolutional Neural Networks Vis. Recognit*, 2017.

Wei, J. and Zou, K. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. pp. 6383–6389, 2019.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., and Brew, J. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771, 2019.

Yu, X., Ren, X., Sun, Y., Gu, Q., Sturt, B., Khandelwal, U., Norick, B., and Han, J. Personalized entity recommendation: A heterogeneous information network approach. In *Proceedings of the 7th ACM international conference on Web search and data mining*, pp. 283–292. ACM, 2014.

APPENDIX

A Models

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**$_1$ 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 not used in the workflow.

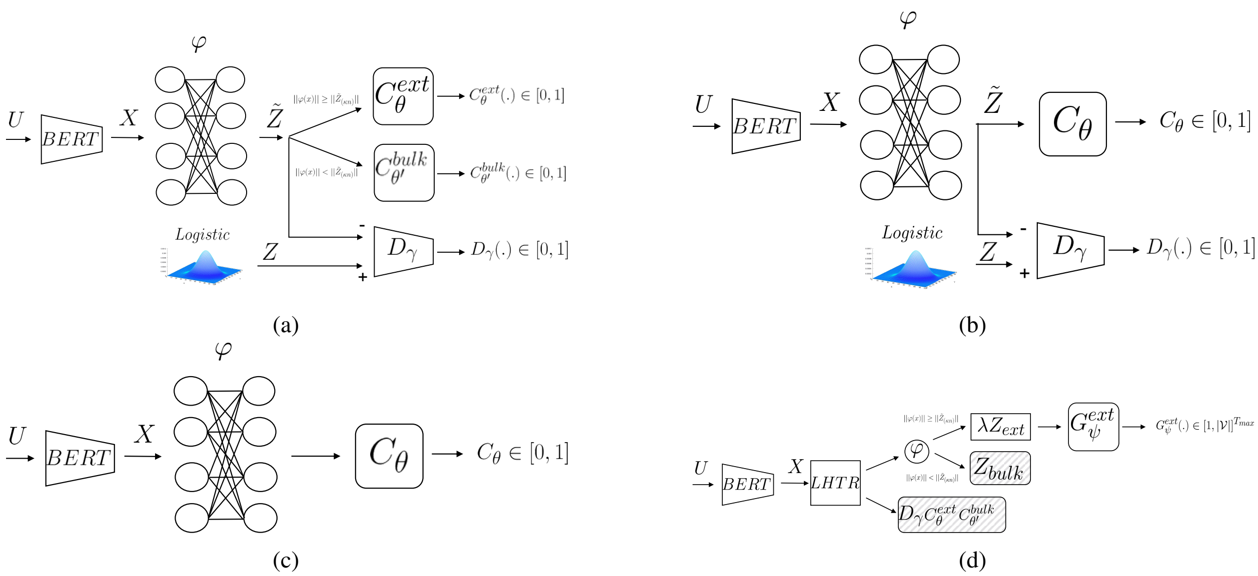


Figure 4: Illustrative pipelines

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 2b 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 $(X_i)_{i\in\{1,\dots,n\}}$ by applying the rank-transformation:

$$\widehat{T}(x) = \left(1/\left(1 - \widehat{F}_j(x)\right)\right)_{j=1,\dots,d}$$

for all $x = (x^1, \dots, x^d) \in \mathbb{R}^d$ where $\widehat{F}_j(x) \overset{\text{def}}{=} \frac{1}{n+1}\sum_{i=1}^{n} \mathbb{1}\{X_i^j \leq x\}$ is the j^{th} empirical marginal distribution. Denoting by V_i the standardized variables, $\forall i \in \{1,\dots,n\}, V_i = \hat{T}(|X_i|)$. 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 $\widehat{T}$ instead of the genuine marginal *c.d.f.*'s F_j. After this standardization step, the selected extreme samples are $\{V_i, \|V_i\| \geq V_{(\lfloor \kappa n \rfloor)}\}$.

B.3 Enforcing regularity assumptions in Theorem 1

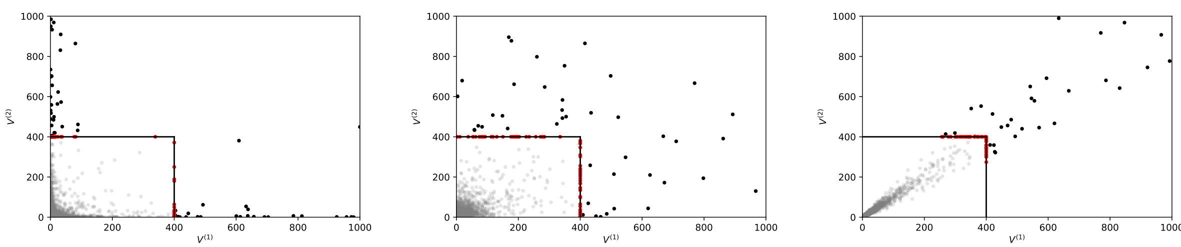
The methodology proposed in the present paper consists in learning a representation Z for text data *via* **`LHTR`** satisfying the regular variation condition (2). This condition is weaker than the assumptions from Theorem 1 for two reasons: first, it does not imply that each class (conditionally to the label Y) is regularly varying, only that the distribution of Z (unconditionally to the label) is. Second, in Jalalzai et al. (2018), it is additionally required that the regression function $\eta(z) = \mathbb{P}\{Y+1 \mid Z=z\}$ converges uniformly as $\|z\| \to \infty$. Getting into details, one needs to introduce a limit random pair (Z_∞, Y_∞) which distribution is the limit of $\mathbb{P}\left\{Y = \cdot\,, t^{-1}Z \in \cdot \;\middle|\; \|Z\| > t\right\}$ as $t \to \infty$. Denote by η_∞ the limiting regression function, $\eta_\infty(z) = \mathbb{P}\{Y_\infty = +1 \mid Z_\infty = z\}$. The required assumption is that

$$\sup_{\{z \in \mathbb{R}^d_+ : \|z\| > t\}} \left|\eta(z) - \eta_\infty(z)\right| \xrightarrow[t\to\infty]{} 0. \tag{4}$$

Uniform convergence (4) is not enforced in **`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.* $F(x) = \exp\big\{-\big(\sum_{j=1}^d x^{(j)\,\frac{1}{\delta}}\big)^{\delta}\big\}$. Samples from the logistic distribution can be simulated according to the algorithm proposed in Stephenson (2003). Figure 5 illustrates this distribution with various values of δ. Values of δ close to 1 yield non concommittant extremes, *i.e.* the probability of a simultaneous exceedance of a high threshold is negligible. Conversely, for small values of δ, extreme values tend to occur simultaneously. These two distinct tail dependence structures are respectively called 'asymptotic independence' and 'asymptotic dependence' in the EVT terminology.



(a) near tail independence (b) moderate tail dependence (c) high tail dependence

Figure 5: Illutration 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.

C Experiments

C.1 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 $5 * 10^{-4}$. The weight parameter ρ_3 in the loss function (Jensen-Shannon divergence from the target) is set to 10^{-3}. Each component φ, C^{bulk} and C^{ext} is made of 3 fully connected layers, the sizes of which are reported in Table 4.

BERT representation for text data. We use BERT pretrained models and code from the library *Transformers* [7]. 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.

[7] `https://github.com/huggingface/transformers`

	Layers' sizes
φ	[2,4,2]
$C^{bulk}_{\theta'}$	[2,8,1]
C^{ext}_{θ}	[2,8,1]

Table 4: Sizes of the successive layers in each component of **LHTR** used in the toy example.

Network architectures Tables 5 and 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 $\rho_1 = (1 - \hat{\mathbb{P}}(||Z|| \geq ||Z_{(\lfloor \kappa n \rfloor)}||))^{-1}$ and $\rho_2 = \hat{\mathbb{P}}(||Z|| \geq ||Z_{(\lfloor \kappa n \rfloor)}||)^{-1}$.

	NN model	**LHTR**$_1$	**LHTR**
Sizes of the layers φ	[768,384,200,50,8,1]	[768,384,200,100]	[768,384,200,150]
Sizes of the layers $C^{bulk}_{\theta'}$	X	[100,50,8,1]	[150,75,8,1]
Sizes of the layers C^{ext}_{θ}	X	X	[150,75,8,1]
ρ_3	X	X	0.001

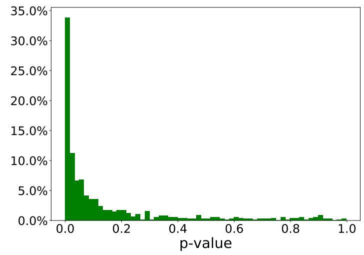
Table 5: Network architectures for *Amazon dataset 1* and *Yelp dataset 1*. The weight decay is set to 10^5, the learning rate is set to $5 * 10^{-4}$, the number of epochs is set to 500 and the batch size is set to 64.

	NN model	**LHTR**$_1$	**LHTR**
Sizes of the layers φ	[768,384,200,50,8,1]	[768,384,200,100]	[768,384,200,150]
Sizes of the layers of $C^{bulk}_{\theta'}$	[150,75,8,1]	[100,50,8,1]	[150,75,8,1]
Sizes of the layers of C^{ext}_{θ}	X	X	[150,75,8,1]
ρ_3	X	X	0.01

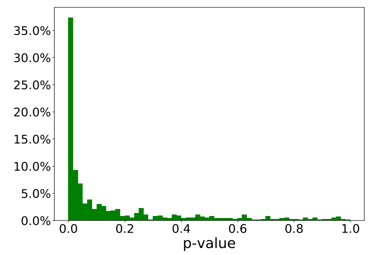
Table 6: Network architectures for *Amazon dataset 2* and *Yelp dataset 2*. The weight decay is set to 10^5, the learning rate is set to $1 * 10^{-4}$, the number of epochs is set to 500 and the batch size is set to 256.

C.2 Scale invariance: comparison of BERT and LHTR

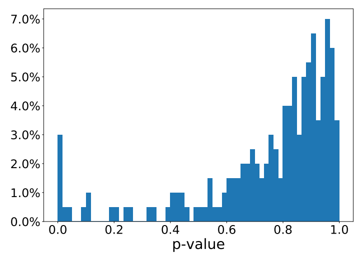
BERT is not regularly varying: In order to show that X is not regularly varying, independence between $\|X\|$ and a margin of $\Theta(X)$ can be tested Coles & Tawn (1994), which is easily done *via* correlation tests. Pearson correlation tests were run on the extreme samples of BERT and **LHTR** embeddings of *Amazon dataset 1* and *Yelp dataset 1*. The tests were performed between all margins of $\big(\Theta(X_i)\big)_{1\geq i \geq n}$ and $\big(\|X_i\|\big)_{1\geq i\geq n}$.



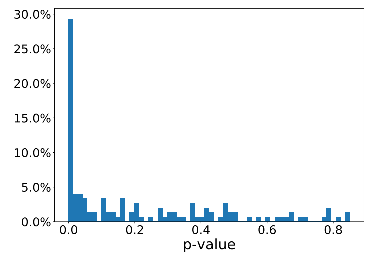
(a) *Yelp dataset 1* - BERT



(b) *Amazon dataset 1* - BERT



(c) *Yelp dataset 1* - **LHTR**



(d) *Amazon dataset 1* - **LHTR**

Figure 6: Histograms of the p-values for the non-correlation test between $\left(\Theta(X_i)\right)_{1\geq i\geq n}$ and $\left(\|X_i\|\right)_{1\geq i\geq n}$ 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 dataset 1* and *Yelp dataset 1* 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 dataset 1* than with *Amazon dataset 1*. 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 sentence 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 dataset 1*. The color (white or black respectively) indicates the predicted class (respectively -1 and $+1$). For values of λ close to 1, the predicted class is -1 but the prediction shifts to class $+1$ for larger values of λ.

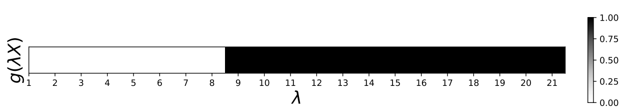


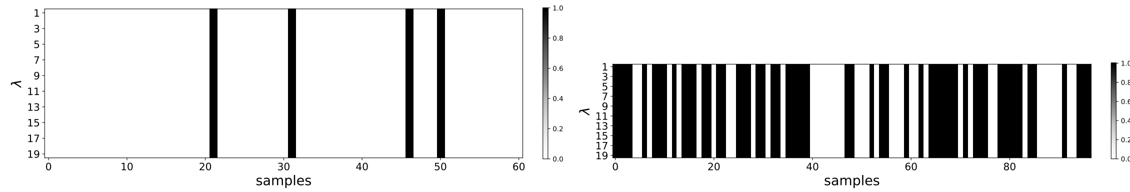
Figure 7: Lack of scale invariance of the classifier trained on BERT: evolution of the predicted label $g(\lambda X)$ from -1 to $+1$ for increasing values of λ, for one particular example X.

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 $g^{\text{ext}}(\lambda Z_i)$ for increasing values of the scale factor $\lambda \geq 1$ and Z_i belonging to $\mathcal{T}_{\text{test}}$, the set of samples considered as extreme in the learnt representation. For any such sample Z, the predicted label remains constant as λ varies, *i.e.* it is scale invariant, $g^{\text{ext}}(\lambda Z) = g^{\text{ext}}(Z)$, for all $\lambda \geq 1$.

C.3 Experiments for data generation

C.3.1 experimental setting

As mentioned in Section 5.1 hyperparameters for dataset augmentation are detailed in Table 7. For the Transformer Decoder we use 2 layers with 8 heads, the dimension of the key and value is set to 64 Vaswani et al. (2017) and the inner dimension is set to 512. The architectures for the models proposed by Wei & Zou (2019) and Kobayashi (2018)



(a) *Amazon dataset 1* (b) *Yelp dataset 1*

Figure 8: Scale invariance of g^{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 λ, for samples Z_i from the extreme test set $\mathcal{T}_{\text{test}}$ from *Amazon dataset 1* (Figure 8a) and *Yelp dataset 1* (Figure 8b)

.

	LHTR
Sizes of the layers φ	[768,384,200,150]
Sizes of the layers of $C_{\theta'}^{bulk}$	[150,75,8,1]
Sizes of the layers of C_{θ}^{ext}	[150,75,8,1]
ρ_3	0.01

Table 7: For *Amazon dataset 2* and *Yelp dataset 2*, the weight decay is set to 10^5, the learning rate is set to $1 * 10^{-4}$, the number of epochs is set to 100 and the batch size is set to 256.

are chosen according to the original papers. For a fair comparison with Kobayashi (2018), we update the language model with a BERT model, the labels are embedded in $\mathbb{R}^{10}$ and fed to a single MLP layer. The new model is trained using AdamW Loshchilov & Hutter (2017).

C.3.2 Influence of the scaling factor on the linguistic content

Table 8 gathers some extreme sentences generated by **GENELIEX** for λ ranging from 1 to 1.5. No major linguistic change appears when λ varies. The generated sentences are grammatically correct and share the same polarity (positive or negative sentiment) as the input sentence. Note that for greater values of λ, a repetition phenomenon appears. The resulting sentences keep the label and polarity of the input sentence but repeat some words Holtzman et al. (2019).

input	(it wasn't busy either), the building was cold.
$\lambda = 1$	(it was not occupied either), the building was cold.
$\lambda = 1.1$	(i wasn't busy either), the building was frozen.
$\lambda = 1.3$	also, the building was freezing.
$\lambda = 1.5$	plus, the building was colder than ice.
input	food quality has been horrible.
$\lambda = 1$	food quality has been terrible.
$\lambda = 1.1$	the quality of the food was horrible.
$\lambda = 1.3$	the quality of the food has been horrible.
$\lambda = 1.5$	the quality of food was terrible.
input	overall , i like there food and the service .
$\lambda = 1$	i love food and the service.
$\lambda = 1.1$	on the whole, i like food and service.
$\lambda = 1.3$	in general, i like the food and the service.
$\lambda = 1.5$	in general, i like food and service.
input	the desserts were a bit strange.
$\lambda = 1$	the desserts were a little weird.
$\lambda = 1.1$	the desserts were very strange.
$\lambda = 1.3$	the desserts were terrible.
$\lambda = 1.5$	the desserts were terrible.
input	we definately enjoyed ourselves.
$\lambda = 1$	we enjoyed ourselves.
$\lambda = 1.1$	we had a lot of fun.
$\lambda = 1.3$	we've really enjoyed each other.
$\lambda = 1.5$	we certainly had fun.

input	seriously killer hot chai latte.
$\lambda = 1$	-it's a real killer.
$\lambda = 1.2$	he is a real killer.
$\lambda = 1.3$	he likes to kill.
$\lambda = 1.5$	i loves murders.
input	all of the tapas dishes were delicious!
$\lambda = 1$	all the tapas was delicious.
$\lambda = 1.1$	all tapas dishes were delicious!
$\lambda = 1.3$	all the tapas dishes were delicious!
$\lambda = 1.5$	the tapas were great!
input	there was hardly any meat.
$\lambda = 1$	there was almost no meat.
$\lambda = 1.1$	there was practically no meat.
$\lambda = 1.3$	there was almost no meat.
$\lambda = 1.5$	there was no meat.
input	waiter was a jerk.
$\lambda = 1$	the waiter was a jerk.
$\lambda = 1.1$	awaiter was a poor guy.
$\lambda = 1.3$	waiter was an idiot.
$\lambda = 1.5$	waiter was such an idiot.
input	i 'm not eating here!
$\lambda = 1$	i don't eat here.
$\lambda = 1.1$	i don't eat here!
$\lambda = 1.3$	i'm not going to eat here!
$\lambda = 1.5$	i will never going to eat here!

Table 8: Sentences generated by **GENELIEX** for extreme embeddings implying label (sentiment polarity) invariance for generated sentence. λ is the scale factor.