Experimenting with Power Divergences for Language Modeling

Matthieu Labeau **Shay B. Cohen**
Institute for Language, Cognition and Computation
School of Informatics, University of Edinburgh
{mlabeau,scohen}@inf.ed.ac.uk

Abstract

Neural language models are usually trained using Maximum-Likelihood Estimation (MLE). The corresponding objective function for MLE is derived from the Kullback-Leibler (KL) divergence between the empirical probability distribution representing the data and the parametric probability distribution output by the model. However, the word frequency discrepancies in natural language make performance extremely uneven: while the perplexity is usually very low for frequent words, it is especially difficult to predict rare words.

To address that, we experiment with several families (α, β and γ) of *power divergences*, generalized from the KL divergence, for learning language models with an objective different than standard MLE. Intuitively, these divergences should affect the way the probability mass is spread during learning, notably by prioritizing performance on high or low-frequency words. In addition, we implement and experiment with various sampling-based objectives, where the computation of the output layer is only done on a small subset of the vocabulary.

They are derived as *power* generalizations of a softmax approximated via Importance Sampling, and Noise Contrastive Estimation, for accelerated learning. Our experiments on the Penn Treebank and Wikitext-2 show that these power divergences can indeed be used to prioritize learning on the frequent or rare words, and lead to general performance improvements in the case of sampling-based learning.

1 Introduction

Language models are an important component in many NLP tasks, where they provide prior knowledge on the language used. They are conditional models that aim to predict the next token in a sequence: they can be applied to basic units ranging from individual characters to full words, each approach coming with its own benefits and limitations (Merity et al., 2018a). Word-level language models have traditionally been based on n-gram counts, obtaining good performance with smoothing techniques (Kneser and Ney, 1995; Goodman, 2001). Recently, neural networks have shown strong results in language modeling (Bengio et al., 2001), especially recurrent neural networks (Mikolov et al., 2011b). As previous approaches, like maximum entropy models (Berger et al., 1996), neural language models are trained via Maximum Likelihood Estimation (MLE). Thus, their training cost grows linearly with the number of words in the vocabulary, often making it prohibitively slow. This motivated a large amount of research work, bringing a variety of solutions (Chen et al., 2016).

The large vocabulary sizes encountered in training corpora arguably stem from the fact that the frequency distribution of words in a corpus of natural language follows Zipf's law (Powers, 1998). This also implies that the discrepancy between counts of high-frequency and low-frequency words increases with the size of the corpus, as well as the number of those low-frequency words. As a consequence, distributed word representations of low-frequency words are difficult to learn. Numerous approaches use decomposition of words with various sub-word units (Sennrich et al., 2016; Kim et al., 2016), but the same phenomenon exists for low-frequency subwords. In order to deal with this issue and to accelerate training, Grave et al. (2017a) implement a dependency between embedding sizes and word frequencies in the output layer, while Baevski and Auli (2019) apply it to the input layer, comparing the possible choices of which units to model. Using a different approach, Gong et al. (2018) attempt to learn word representations that are less affected by these large discrepancies in word frequencies, with an adver-

Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pages 4104–4114,
Hong Kong, China, November 3–7, 2019.

sarial training method to force the model to make frequent and rare word embeddings hard to differentiate based on word frequency alone.

These improvements have been obtained by explicitly incorporating in the model different ways of treating words according to their frequency. However, learning is always made via (or approximating) Maximum Likelihood Estimation, which finds the distribution that maximizes entropy subject to the constraints given by training examples. In this work, we explore the possibility of affecting how words are learned depending on their frequency by using alternative loss functions. We specifically explore *power divergences*, obtained through various generalizations of the Kullback-Leibler divergence, which is traditionally used to obtain the MLE objective function. This is motivated by the intuition that a well-suited power factor may direct learning towards prioritizing high or low-frequency words, instead of learning uniformly.

In this paper, we derive and experiment with the objective functions obtained from three power-divergences: the α, β and γ divergences. We also derive objective functions for the corresponding generalizations of two sampling-based methods: an Approximated Softmax obtained with importance sampling, and Noise Contrastive Estimation. We conduct a set of experiments comparing these objectives and their effect of various parts of the word frequency distribution, by training and evaluating models on two corpora: the Penn Treebank and Wikitext-2. Our experiments show that depending on the vocabulary used and the choice of power divergence, it is indeed possible to gear learning to focus on the most frequent or infrequent words. We also observe that, while the MLE gives the best overall performance for exact objectives, derived from the KL-divergence, our generalized objectives yield perplexity improvements compared to baselines for both sampling-based methods, up to 1 point in perplexity on both corpora.

2 Background

Language modeling aims to learn a probability distribution over a sequence of tokens from a finite target vocabulary $\mathcal{Y}$. Such a distribution is decomposed into a product of conditional distributions of tokens over $\mathcal{Y}$ given the previous tokens in the sequence. Hence, we learn a parametric model of the form $p_\theta(y|x)$, where $x \in \mathcal{X}$ represents the sequence of previous tokens, y is a target label belonging to $\mathcal{Y}$, and θ is the set of model parameters. They are obtained via maximum likelihood estimation (MLE), which consists in minimizing the *negative likelihood* objective function:

$$\mathrm{NLL}(\theta) = -\sum_{(x,y)\in\mathcal{D}} \log p_\theta(y|x)$$

over examples (x, y) corresponding to sequences of tokens drawn from the data $\mathcal{D}$. This can be seen as minimizing the *Kullback-Leibler* divergence between the parametrized probability distribution p_θ that we are learning and the distribution $p_\mathcal{D}$ described by our training data:

$$D_{KL}(p_\mathcal{D}||p_\theta) = \sum_{(x,y)\in\mathcal{X}\times\mathcal{Y}} p_\mathcal{D}(y|x) \log \frac{p_\mathcal{D}(y|x)}{p_\theta(y|x)},$$

since $p_\mathcal{D}(y|x) = 1$ if the sequence (x, y) appears in the training data, and equals 0 otherwise. Hence, the set of parameters θ^* minimizing $D_{KL}(p_\mathcal{D}||p_\theta)$ is the maximum likelihood distribution on the training data $\mathcal{D}$.

In this work, we will use several classes of divergences. A measure of discrepancy D between two probability densities p and q is a divergence if $D(p||q) \geq 0$ with equality if and only if $p = q$.[1] In the following, we will derive an objective function from a divergence D with the data distribution as first argument, and the second being the distribution of the parametric model:

$$Obj(\theta) = D(p_\mathcal{D}||p_\theta).$$

3 Power Divergences

A large number of divergence measures has been introduced for a variety of applications (Basseville, 2013). Several families of divergences are notably obtained from generalizing the Kullback-Leibler divergence by using a *generalized logarithm* function, which is a power function:

$$\log_\alpha(x) = \frac{1}{1-\alpha}(x^{1-\alpha} - 1), \tag{1}$$

defined by a parameter α, and that converges to the logarithm as $\alpha \to 1$. In this section, we will consider three families of divergences that can be generated from this function; see Cichocki and Amari

[1] Divergences are not necessarily metrics: they are defined only by non-negativity and positive definiteness.

(2010) for a review. For now, they are to be applied only to *normalized* probability densities, implying that $\sum_{y \in \mathcal{Y}} p_\theta(y|x) = 1$.

3.1 α-divergences

The notion of α-divergence was introduced by Csiszar (1967). The full expression of $D_\alpha(p||q)$ is shown in Table 1. It is a special case of f-divergence (Ali and Silvey, 1966), derived from a standardized version of the generalized logarithm[2] (Eq. 1). Applying α-divergences to parameter estimation generalizes MLE, and can be shown to have similar properties, as consistency and asymptotic normality of the estimation error (Keziou, 2003). Intuitively, the choice of α will impact the importance of the likelihood ratio p/q: while the limiting cases $\alpha \to 1$ and $\alpha \to 0$ are the Kullback-Leibler $D_{KL}(p||q)$ and the reverse Kullback-Leibler divergences $D_{KL}(q||p)$, having $\alpha \geq 1$ will make learning zero-avoiding[3] for q, and $\alpha \leq 0$ zero-forcing[4] (Minka, 2005). We should also note that $D_\alpha(p||q) = D_{1-\alpha}(q||p)$. When working with normalized densities, the α-divergence is linked to the Rényi divergence (Rényi, 1961), which has been used to measure domain similarity (Van Asch and Daelemans, 2010). Given that we are trying to learn from conditional distributions which are zero everywhere except for one target token, we are interested in experimenting with values of $\alpha > 1$, which should push the model towards generalizing, while having $\alpha \in (0.5, 1)$ should force the model to concentrate probability mass on training examples. Since we are here working with normalized distribution, we obtain the following objective:

$$Obj_\alpha(\theta) = \frac{1}{\alpha(\alpha-1)} \sum_{(x,y) \in \mathcal{D}} (p_\theta(y|x))^{1-\alpha}.$$

3.2 β-divergences

The β-divergence, also called *density power divergence* was introduced by Basu et al. (1998) as a robust estimation method, which showed it to be consistent for parameter estimation, with asymptotic normality of the estimation error (Basu et al., 1998, Theorem 2). The full expression of $D_\beta(p||q)$ is shown in Table 1.[5] It can be seen as a *Bregman* divergence (Bregman, 1967) also derived using the generalized logarithm[2] (Eq. 1). The motivation is to obtain divergences that are robust to outliers, which is the case for values of $\beta > 1$; choosing $\beta = 2$ gives us the L_2-loss, while $\beta \to 1$ gives the Kullback-Leibler divergence $D_{KL}(p||q)$ as a limiting case. Hence, choosing a value close to 1 while larger is supposed to provide a compromise between the efficiency of the Kullback-Leibler divergence and the robustness of the L_2-loss. Similarly, we can expect to give more importance to outliers (which we suppose to be the low-frequency tokens) by choosing $\beta < 1$. The objective becomes:

$$Obj_\beta(\theta) = \frac{1}{\beta(\beta-1)} \times \sum_{(x,y) \in \mathcal{D}} \left[(\beta - 1) \sum_{y' \in \mathcal{Y}} (p_\theta(y'|x))^\beta - \beta (p_\theta(y|x))^{\beta-1} \right].$$

3.3 γ-divergences

Eguchi and Kato (2010) introduced the γ-divergence as a modification of the β-divergence, with the specific goal of obtaining a *scale-invariant* version of the robust β-divergence,[6] also showing it to be consistent for parameter estimation, with asymptotic normality of the estimation error (Eguchi and Kato, 2010, Section 3). This divergence has notably been used for the estimation of parameters without normalizing the output probability distribution (Takenouchi and Kanamori, 2015).[7] The general expression of the γ-divergence is shown in Table 1. While the use of the log non-linearity makes this divergence non-separable, which at first glance could be thought to complicate computation in practice, our motivation for using it is its scale-invariance, which we will discuss in Section 4.3. Applied to our prob-

[2] α and β-divergences are related to the *Tsallis* entropy, which can be seen as a deformation of the Shannon entropy using this same generalized logarithm: see Appendix A from Cichocki and Amari (2010) . Recently, Blondel et al. (2018) showed how to build Fenchel-Young Losses from the same Tsallis entropies.

[3] Having $p_i > 0$ will enforce $q_i > 0$.

[4] Having $p_i = 0$ will enforce $q_i = 0$.

[5] For readability, we keep the notation and values used in Cichocki and Amari (2010) instead of Basu et al. (1998).

[6] 'Scale-invariance' here means that the divergence should remain unchanged when one or both of the measures it is applied to are multiplied by scalars.

[7] However, this particular method is not suitable in our case because of the sparsity of our data.

lem, the objective becomes:

$$Obj_\gamma(\theta) = \sum_{(x,y)\in\mathcal{D}} \left[\log p_\theta(y|x) - \frac{1}{\gamma} \log \sum_{y'\in\mathcal{Y}} (p_\theta(y'|x))^\gamma \right]. \tag{2}$$

4 Accelerating Learning

In order to use the previously defined objectives, we need to compute the model probabilities $p_\theta(y|x)$, which are usually obtained using a softmax function:

$$p_\theta(y|x) = \frac{\exp(s_\theta(x,y))}{\sum_{y'\in\mathcal{Y}} \exp(s_\theta(x,y'))}$$

applied on scores (or logits) $s_\theta(x, y')$ output by the model for every possible target token $y' \in \mathcal{Y}$. However, as explained earlier, $\mathcal{Y}$ can be very large, hence computing all the scores and summing them is extremely slow. We choose to follow the strategies employed by Jozefowicz et al. (2016): approximating the softmax using importance sampling (Bengio and Sénécal, 2003, 2008), and Noise Contrastive Estimation (NCE; Gutmann and Hyvärinen 2010; Mnih and Teh 2012). Besides, the divergences presented in Section 3 can be applied to *positive measures*. Hence, a possible third direction is to instead approximate the objectives to the *un-normalized* model distributions. All the objectives presented in this section are explained in Appendix A.4.

4.1 Approximated softmax

Plugging the first solution into our objectives is straightforward: using self-importance sampling to approximate directly the multinomial classification probability, we compute p_θ via an *approximated softmax*:[8]

$$p_\theta(y|x) \approx \frac{\frac{\exp(s_\theta(x,y))}{p_n(y)}}{\frac{\exp(s_\theta(x,y))}{p_n(y)} + \sum_{\substack{i=1 \\ \hat{y}_i \sim p_n}}^{k} \frac{\exp(s_\theta(x,\hat{y}_i))}{p_n(\hat{y}_i)}}$$

where p_n is an auxiliary distribution chosen to reflect the training data while still being easy to sample from, and $k \ll |\mathcal{Y}|$ is the number of samples drawn.

[8]The same objective is called 'Ranking NCE' by Ma and Collins (2018).

4.2 Adapting Noise-Contrastive Estimation

Noise-Contrastive Estimation consists of training the model for a surrogate binary classification task where, given examples from the mixture:

$$\frac{1}{k+1} p_\mathcal{D} + \frac{k}{k+1} p_n$$

we learn to discriminate between examples coming from data and samples coming from the auxiliary noise distribution p_n. Minimizing the NCE loss function has been shown to be equivalent to minimizing a Bregman divergence (Gutmann and Hirayama, 2011); however, the transformation that we need to apply to the associated generating function (Pihlaja et al., 2012) in order to obtain a power divergence is not straigthforward. Instead, with the posterior classification probability:

$$p_\theta(C = \text{True}|y, x) = \frac{p_\theta(y|x)}{p_\theta(y|x) + kp_n(y)} \tag{3}$$

Noting p_θ^C the positive measure on $\mathcal{X} \times \mathcal{Y}$:

$$p_\theta^C : (x, y) \rightarrow p_\theta(C = \text{True}|y, x)$$

we can show that the NCE objective function can be derived from the following expression:[9]

$$D_{KL}(p_\mathcal{D}^C || p_\theta^C) + D_{KL}(1 - p_\mathcal{D}^C || 1 - p_\theta^C)$$

Knowing this, we simply need to replace D_{KL} by the other divergences to derive the α, β and γ objective functions associated to this surrogate classification task. It is interesting to note that the power transformations will here be applied on the posterior classification probabilities p_θ^C instead of categorical probabilities p_θ.

4.3 Working With Positive Measures

The three divergences presentend in Section 3 are defined on positive measures: in theory, we can simply use the exp function on the scores s_θ and do not need to normalize them:

$$Obj(\theta) = D(p_\mathcal{D} || \exp(s_\theta))$$

However, neither the α and β divergences are scale invariant (see right column of Table 1 and Cichocki and Amari 2010). We can show that working with an un-normalized model distribution will, in both those cases, give an objective proportional to the scale of the model, allowing the divergence

[9]The full derivation is given in Appendix A.2.

	Expression	Scaling properties
$\alpha \in \mathbb{R} \setminus \{0,1\}$	$\frac{1}{\alpha(\alpha-1)} \sum_{i=1}^{n} \left[p_i^\alpha q_i^{1-\alpha} - \alpha p_i + (\alpha-1) q_i\right]$	$D_\alpha(c \times p \| c \times q) = c \times D_\alpha(p\|q)$
$\beta \in \mathbb{R} \setminus \{0,1\}$	$\frac{1}{\beta(\beta-1)} \sum_{i=1}^{n} \left[p_i^\beta + (\beta-1) q_i^\beta - \beta p_i q_i^{\beta-1}\right]$	$D_\alpha(c \times p \| c \times q) = c^\beta \times D_\alpha(p\|q)$
$\gamma \in \mathbb{R} \setminus \{0,1\}$	$\frac{1}{\gamma(\gamma-1)} \log \sum_{i=1}^{n} p_i^\gamma + \frac{1}{\gamma} \log \sum_{i=1}^{n} q_i^\gamma - \frac{1}{\gamma-1} \log \sum_{i=1}^{n} p_i q_i^{\gamma-1}$	$D_\gamma(c_1 \times p \| c_2 \times q) = D_\gamma(p\|q)$

Table 1: Complete expressions of α, β, and γ divergences between two positive measures $p = (p_i)_{i=1}^n$ and $q = (q_i)_{i=1}^n$ on $\mathbb{R}_+^n$, as well as their scaling properties. Note that the α-divergence simplifies if we restrict them to the set of normalized probability densities (if $\sum_{i=1}^n p_i = \sum_{i=1}^n q_i = 1$).

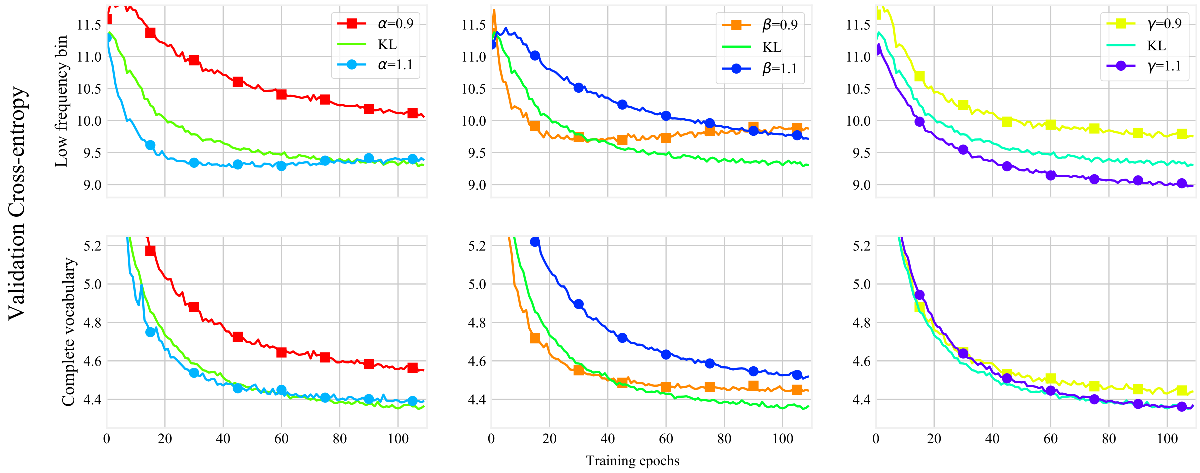

Figure 1: Validation cross-entropy values obtained during the beginning of training models with Obj_α (left), Obj_β (center) and Obj_γ (right) on the PTB with a full vocabulary. Words are grouped into 5 buckets of equal size, following their frequencies. On the top is shown the cross-entropy obtained on the bucket of lowest frequency words, while the global cross-entropy is displayed on the bottom.

to be minimized by minimizing the scale alone, with no actual learning.[10] This is not an issue for the γ-divergence, since $D_\gamma(p_\mathcal{D} \| \exp(s_\theta)) = D_\gamma(p_\mathcal{D} \| p_\theta)$. Hence, the objective of Eq. 2 may be simplified as following:

$$Obj_\gamma(\theta) = \sum_{(x,y)\in\mathcal{D}} \left[s_\theta(x,y) - \frac{1}{\gamma} \log \sum_{y'\in\mathcal{Y}} \exp\left(\gamma s_\theta(x,y')\right) \right].$$

We can further accelerate learning by using importance sampling on the second term, and because of the logarithm applied to the sum, we obtain an objective that is equivalent to composing the approximated softmax with γ-divergence, as can be seen in Appendix A.4.

[10]The proof is given in Appendix A.1.

5 Experiments

Our goal is to compare the effect of the various objective functions we proposed in Sections 3 and 4, and especially study how the values of α, β and γ affect learning and performance, overall as well as on low-frequency words. Since each model is trained with a different objective function, the training scores are not comparable. Hence, we use the validation cross-entropy and perplexity at each epoch as a way to track progress during training.

5.1 Datasets

We perform our experiments on two widely used, reasonably sized datasets: the Penn Treebank (PTB; Mikolov et al. 2011a) and WikiText-2 (WT2; Merity et al. 2017). The PTB, heavily preprocessed, retains the $10k$ most frequent words in its vocabulary, while the others tokens are replaced by a common <*unk*> token. The WT2, about two times larger, retains words that appear at least three times in the training data in its vocabulary, which makes it $33,278$ words. To study the impact of our various objective functions on a vocabulary containing very rare words, we also experiment with a version on the PTB to which we only apply limited preprocessing, allowing us to keep its full training vocabulary, which contains 39030 words.

5.2 Experimental setup

We based our setup on the ASGD weight-dropped LSTM (AWD-LSTM) models of Merity et al. (2018b), since to the best of our knowledge they give state-of-the-art results on the PTB and WT2 for models that are build with a softmax output layer and do not use any adaptative method (as the pointer sentinel LSTM (Merity et al., 2017), the neural cache (Grave et al., 2017b) and the dynamic evaluation (Krause et al., 2018)). Our models[11], replications of their 3-layer LSTMs with tied input and output representations, were built using their implementation.[12] For each dataset, we follow their choice of hyperparameters, only modifying the objective functions.

For our sampling-based objectives, we use $k = 1024$ samples, and the unigram distribution as p_n. In the case of objectives derived from binary NCE, to avoid issues with the phenomenon of *self-normalization* and consistency issues (Ma and Collins, 2018) we chose to use *blackout* (Ji et al., 2015). Indeed, this method amounts to using a noise distribution which depends on the model probabilities, making the normalization term disappear from the posterior classification probabilities of Eq. 3. Hence, we have an objective function that is fast to compute without any supplementary assumption, and the negative samples are still drawn from the unigram distribution. For both AS and NCE, in our setting, the computation time is reduced to about 40% of the time taken by MLE on the WT2, and 50% on the PTB.

6 Results and discussion

We turn to describe the results of our experiments.

[11] https://github.com/MatthieuLabeau/power-divergences-LM

[12] https://github.com/salesforce/awd-lstm-lm.

6.1 Qualitative results on the full vocabulary PTB

To observe the behavior of the proposed objectives for various choices of α, β and γ, we plot the validation cross-entropy at the beginning of learning of models on PTB equipped with a full training vocabulary. We choose values of 0.9 and 1.1 for the power parameters, to experiment with an objective on each side of the baseline MLE objective. We split the words according to frequency into 5 buckets, in order for the buckets to represent have equal size based on word counts, and display both the global cross-entropy and the cross-entropy on the lowest frequency bin in Figure 1. A value of $\alpha > 1$ seems to initially behave better, especially for rare words. This could be expected: intuitively, these values of α should make the model 'stretch' the probability mass. However, as learning progresses, this phenomenon lessens, and the performance on rare words gets worse. A value of $\beta < 1$, supposed to make the model less robust to outliers, seems to make learning faster initially, particularly on rare words, but again improvements lessen. Choosing $\alpha < 1$ or $\beta > 1$ gives a worse cross-entropy overall. This 'inverted' similarity between the behaviors induced by choices of α and β, and the links between the two divergences, have been explored by Patra et al. (2013). Finally, choosing $\gamma > 1$ gives very interesting results, allowing for a better cross-entropy on rare words while retaining the same overall performance than MLE.

6.2 Penn Treebank and WikiText-2

In this section, we present the results of exploratory experiments. We fully train models with the proposed objectives, for a variety of power parameters. For the PTB, the final validation perplexities are presented in Table 2, while we present the final validation cross-entropies for the objectives derived from MLE, on 5 frequency buckets, in Figure 2. For the WT2, they are shown in Table 3 and Figure 3.

With exact objectives: for both corpora, the results for the high and low-frequency buckets seem to confirm that values of α, γ that are smaller and larger than 1 can help prioritizing learning towards the frequent and rare words. The effect of β depending on frequency seems lighter, especially on the PTB: we hypothesize that this is due to the vocabulary being cut short, and containing no very

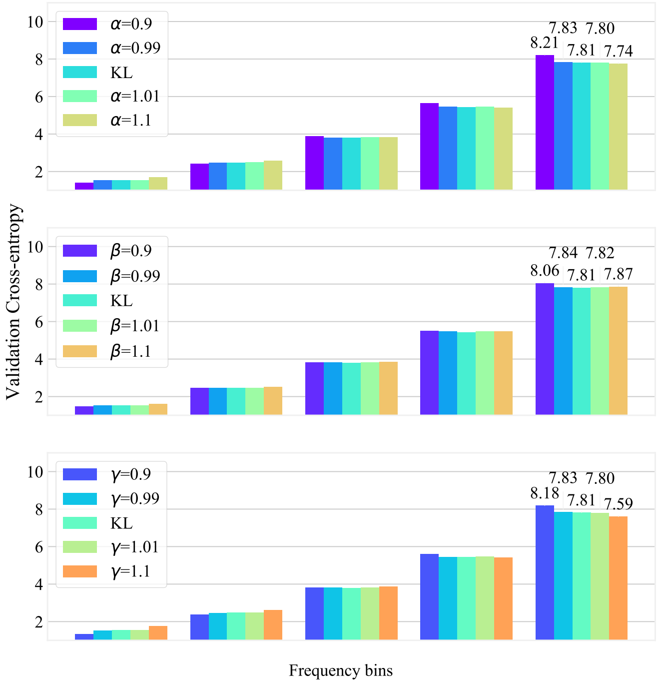

Figure 2: Validation cross-entropy values for the best epoch obtained for models trained with Obj_α (top), Obj_β (middle) and Obj_γ (bottom) on the PTB. Words are grouped into 5 buckets of equal size, following their frequencies. We display values for each bucket from the most frequent words (left) to less frequent ones (right).

Objective		Obj_α	Obj_β	Obj_γ
MLE	0.9	65.8	63.3	62.3
	0.99	60.7	61.0	60.6
	1.0		60.1	
	1.01	60.8	61.1	61.0
	1.1	63.0	63.9	63.3
AS	0.5	147.0	87.1	113.8
	0.9	61.3	62.8	61.0
	0.99	61.7	61.7	61.6
	1.0		61.7	
	1.01	61.8	61.6	62.0
	1.1	65.6	61.6	64.1
	1.5	97.7	127.6	81.7
NCE	0.5	1407.5	88.0	117.2
	0.9	61.4	62.6	61.8
	0.99	61.1	61.4	61.7
	1.0		61.2	
	1.01	61.3	61.1	61.8
	1.1	64.4	61.0	64.4
	1.5	97.9	123.7	81.2

Table 2: Best validation perplexities obtained on the PTB with Obj_α, Obj_β, and Obj_γ, derived from MLE, and approximated objectives AS and NCE, on single models with the same intialization.

Objective	Obj_α		Obj_β		Obj_γ	
Dataset	PTB	WT2	PTB	WT2	PTB	WT2
0.9	65.8	75.8	63.3	68.3	63.2	70.1
	39.7	42.4	107.9	130.0	62.4	69.0
0.99	60.7	66.5	61.0	65.6	60.6	65.9
	57.9	63.0	64.0	69.3	60.6	65.9
1.0	60.1	64.9	60.1	64.9	60.1	64.9
1.01	60.8	65.6	61.1	65.4	61.0	66.0
	63.8	69.3	58.3	61.8	61.0	66.0
1.1	63.0	66.7	63.9	68.3	63.3	68.1
	99.8	114.3	40.7	45.2	62.5	67.0

Table 4: Final validation results obtained on the PTB and WT2, with the exact objectives Obj_α, Obj_β, and Obj_γ, derived from MLE. In each cell we give on top the validation perplexity, and below, the 'counterpart' to perplexity corresponding to the training objective — which is the value being optimized. Each color corresponds to a different objective: values cannot be compared for varying α, β and γ.

rare words, which reduces the discrepancies between higher and lower scoring examples. However, no objective seems to be able to improve on MLE for overall performance.

To explain these results, we may argue that perplexity is a biased measure as MLE directly optimizes it. Hence, we compute the 'counterparts' to perplexity corresponding to each objective. Equivalently to perplexity for MLE, they are directly optimized by their respective objectives, and should decrease to 1 as the model distribution gets closer to the data distribution. Again, since they vary for each objective, they are not comparable between themselves and with perplexity. We display these values for our models trained with exact objectives in Table 4. As α, β and γ are closer to 1, these values get closer to the perplexity of the model. Besides, they are especially close across all values of γ: we can assume that this indicates that the corresponding Obj_γ are 'closer' to the MLE objective. However, tracking these values during training[13] shows that they all behave very similarly to perplexity.

With approximated objectives: for objectives derived from AS and NCE, we observe far less impact of the choice of the power parameter on frequent or rare words,[14] which is probably due to the fact that only a small subset of the vocab-

[13] See Appendix A.3.
[14] See figures in Appendix A.5.

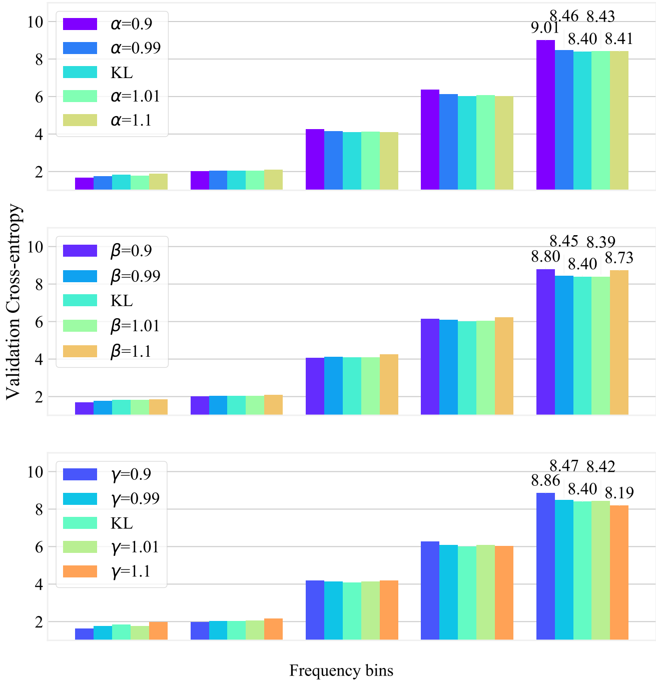

Figure 3: Validation cross-entropy values for the best epoch obtained for models trained with Obj_α (top), Obj_β (middle) and Obj_γ (bottom) on the WT2. Words are grouped into 5 buckets of equal size, following their frequencies. We display values for each bucket from the most frequent words (left) to less frequent ones (right).

Objective		Obj_α	Obj_β	Obj_γ
MLE	0.9	75.8	68.3	70.1
	0.99	66.5	65.6	65.9
	1.0		64.9	
	1.01	65.6	65.4	66.0
	1.1	66.7	68.3	68.1
AS	0.5	203.6	82.1	191.6
	0.9	69.9	65.6	68.0
	0.99	67.4	65.0	65.1
	1.0		65.3	
	1.01	67.5	65.1	65.0
	1.1	69.8	67.8	64.8
	1.5	100.7	211.1	78.1
NCE	0.5	2401.2	84.1	193.1
	0.9	70.6	66.6	69.7
	0.99	66.8	65.7	67.5
	1.0		65.8	
	1.01	65.8	65.9	65.8
	1.1	64.7	68.3	65.4
	1.5	83.8	196.4	77.3

Table 3: Best validation perplexities obtained on the WT2 with Obj_α, Obj_β, and Obj_γ, derived from MLE, and approximated objectives AS and NCE, on single models with the same initialization.

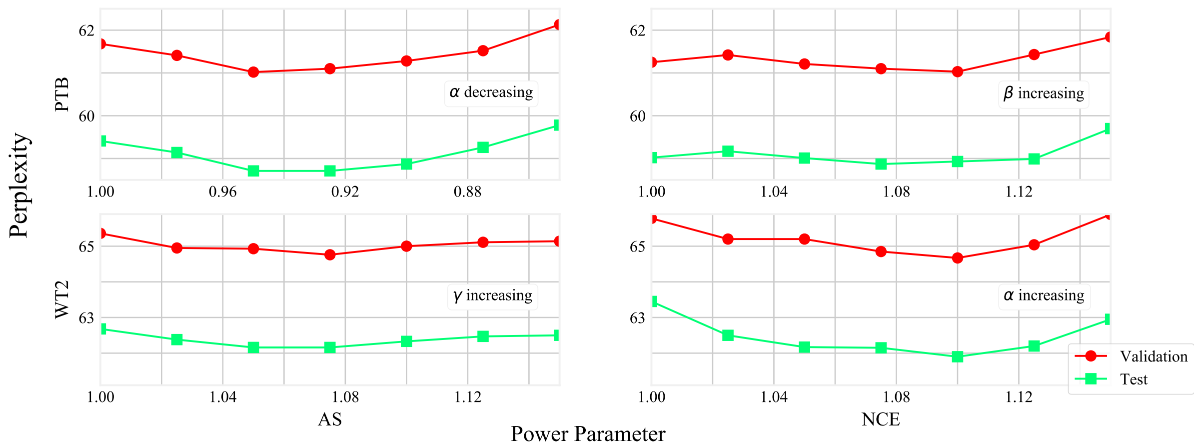

Figure 4: Validation and test perplexities obtained for particular values of α, β or γ with sampling-based objectives in 4 possible pairings of AS and NCE-based objectives trained on the PTB and WT2, on single models with the same initialization.

ulary is updated during learning — and this subset is drawn according to word frequency. However, some objectives provide modest improvements over the corresponding KL-based baseline.

6.3 Searching for the Optimal Power Parameter

In order to verify the potential benefits of our generalization of sampling-based objectives, we use these preliminary results to search for the 'best' power parameter, and check that improvements are consistent for several versions of the model,

		Objective	Validation	Test
PTB		MLE	60.7	58.6
	AS	KL	62.2	59.8
		$\alpha = 0.95$	61.2	58.9
	NCE	KL	61.5	59.2
		$\beta = 1.1$	61.2	59.0
WT2		MLE	65.5	62.8
	AS	KL	65.1	62.6
		$\gamma = 1.075$	64.8	62.1
	NCE	KL	65.8	63.4
		$\alpha = 1.1$	64.7	62.0

Table 5: Best validation and test perplexities obtained on the best performing configurations of power parameter for both corpora and each category of objectives, averaged over 5 models with different initializations.

initialized with different seeds. This search is shown in Figure 4 for 4 different configurations chosen using the preliminary results presented earlier. In each case, we can see that the perplexity seems to decrease, reach a minimum, and then increase monotonously. While verifying this would require a deeper investigation, we can make the hypothesis that the sampling procedure induces noise that some of our proposed objectives can be better suited for. In Table 5, we present perplexity results averaged for different seeds, in order to check that model initialization does not discernibly affect these improvements. They show gains of up to 1 point in perplexity, and a slight improvement over the MLE baseline for WT2, confirming the results obtained on a single model. However, it is tedious to find the optimal power parameter for a given configuration, and we can expect it to shift sensibly when hyper-parameters are modified. The size of the vocabulary and the distribution of word frequencies should intuitively have the biggest influence, as we can see that the behavior of α, β and γ is quite different on the PTB and WT2. In general, finding the optimal divergence with the optimal parameter for a given problem is very difficult. Methods for doing so have only been recently explored; for example, Dikmen et al. (2015) use the Tweedie Likelihood to find the optimal β, which can lead to the optimal α and γ.

7 Conclusion

We explore the use of generalizations of KL-divergence into power (α, β and γ) divergences for training language models. We derive exact but also approximated objectives, based on an approximated softmax using importance sampling, and noise contrastive estimation. We show that in the case of exact objectives, a well-chosen γ-divergence can be used to prioritize learning low or high-frequency words. In the case of approximated objective, we show that our proposed objectives can improve on perplexity, and demonstrate it is the case for several configurations. Further research should investigate the potential gains of using γ-divergences for appropriate downstream tasks.

Acknowledgments

We thank the anonymous reviewers for helpful feedback. We gratefully acknowledge the support of Huawei Technologies.

References

S. M. Ali and S. D. Silvey. 1966. A general class of coefficients of divergence of one distribution from another. *Journal of the Royal Statistical Society. Series B (Methodological)*, 28(1):131–142.

Alexei Baevski and Michael Auli. 2019. Adaptive input representations for neural language modeling. In *International Conference on Learning Representations*.

MichèLe Basseville. 2013. Review: Divergence measures for statistical data processing-an annotated bibliography. *Signal Process.*, 93(4):621–633.

Ayanendranath Basu, Ian R. Harris, Nils L. Hjort, and M. C. Jones. 1998. Robust and efficient estimation by minimising a density power divergence. *Biometrika*, 85(3):549–559.

Yoshua Bengio, Réjean Ducharme, and Pascal Vincent. 2001. A neural probabilistic language model. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 932–938. MIT Press.

Yoshua Bengio and Jean-Sébastien Sénécal. 2003. Quick training of probabilistic neural nets by importance sampling. In *Proceedings of the conference on Artificial Intelligence and Statistics (AISTATS)*.

Yoshua Bengio and Jean-Sébastien Sénécal. 2008. Adaptive importance sampling to accelerate training of a neural probabilistic language model. *IEEE Trans. Neural Networks*, 19(4):713–722.

Adam L. Berger, Vincent J. Della Pietra, and Stephen A. Della Pietra. 1996. A maximum entropy approach to natural language processing. *Comput. Linguist.*, 22(1):39–71.

Mathieu Blondel, André F. T. Martins, and Vlad Niculae. 2018. Learning classifiers with fenchel-young losses: Generalized entropies, margins, and algorithms. Cite arxiv:1805.09717.

L.M. Bregman. 1967. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics*, 7(3):200 – 217.

Wenlin Chen, David Grangier, and Michael Auli. 2016. Strategies for training large vocabulary neural language models. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1975–1985, Berlin, Germany. Association for Computational Linguistics.

Andrzej Cichocki and Shun Amari. 2010. Families of alpha- beta- and gamma- divergences: Flexible and robust measures of similarities. *Entropy*, 12(6):1532–1568.

I. Csiszar. 1967. Information-type measures of difference of probability distributions and indirect observation. *Studia Scientiarum Mathematicarum Hungarica*, 2:229–318.

Onur Dikmen, Zhirong Yang, and Erkki Oja. 2015. Learning the information divergence. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(7):1442–1454.

Shinto Eguchi and Shogo Kato. 2010. Entropy and divergence associated with power function and the statistical application. *Entropy*, 12(2):262–274.

Chengyue Gong, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tie-Yan Liu. 2018. Frage: Frequency-agnostic word representation. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 1334–1345. Curran Associates, Inc.

Joshua T. Goodman. 2001. A bit of progress in language modeling. *Comput. Speech Lang.*, 15(4):403–434.

Edouard Grave, Armand Joulin, Moustapha Cissé, David Grangier, and Hervé Jégou. 2017a. Efficient softmax approximation for gpus. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pages 1302–1310.

Edouard Grave, Armand Joulin, and Nicolas Usunier. 2017b. Improving neural language models with a continuous cache. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Michael Gutmann and Junichiro Hirayama. 2011. Bregman divergence as general framework to estimate unnormalized statistical models. In *UAI*.

Michael Gutmann and Aapo Hyvärinen. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2010, Chia Laguna Resort, Sardinia, Italy, May 13-15, 2010*, pages 297–304.

Shihao Ji, S. V. N. Vishwanathan, Nadathur Satish, Michael J. Anderson, and Pradeep Dubey. 2015. Blackout: Speeding up recurrent neural network language models with very large vocabularies. *CoRR*, abs/1511.06909.

Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. 2016. Exploring the limits of language modeling.

Amor Keziou. 2003. Dual representation of phi-divergences and applications. *Comptes Rendus Mathematique*, 336(10):857 – 862.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M. Rush. 2016. Character-aware neural language models. In *30th AAAI Conference on Artificial Intelligence, AAAI 2016*, pages 2741–2749. AAAI press.

Reinhard Kneser and Hermann Ney. 1995. Improved backing-off for m-gram language modeling. In *In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, volume I, pages 181–184, Detroit, Michigan.

Ben Krause, Emmanuel Kahembwe, Iain Murray, and Steve Renals. 2018. Dynamic evaluation of neural sequence models. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2766–2775, Stockholmsmässan, Stockholm Sweden. PMLR.

Zhuang Ma and Michael Collins. 2018. Noise contrastive estimation and negative sampling for conditional models: Consistency and statistical efficiency. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3698–3707. Association for Computational Linguistics.

Stephen Merity, Nitish Shirish Keskar, and Richard Socher. 2018a. An analysis of neural language modeling at multiple scales. *CoRR*, abs/1803.08240.

Stephen Merity, Nitish Shirish Keskar, and Richard Socher. 2018b. Regularizing and optimizing LSTM language models. In *International Conference on Learning Representations*.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. Pointer sentinel mixture models. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Tomas Mikolov, , Stefan Kombrink, Lukas Burget, and Jan Honza Cernocky. 2011a. Empirical evaluation and combination of advanced language modeling techniques. In *Interspeech*. ISCA.

Tomas Mikolov, Stefan Kombrink, Lukás Burget, Jan Cernocký, and Sanjeev Khudanpur. 2011b. Extensions of recurrent neural network language model. In *ICASSP*, pages 5528–5531. IEEE.

Thomas Minka. 2005. Divergence measures and message passing. Technical report.

Andriy Mnih and Yee Whye Teh. 2012. A fast and simple algorithm for training neural probabilistic language models. In *Proceedings of the 29th International Conference on Machine Learning, ICML 2012, Edinburgh, Scotland, UK, June 26 - July 1, 2012*.

Sujayendu Patra, Avijit Maji, Ayanendranath Basu, and Leandro Pardo. 2013. The power divergence and the density power divergence families: The mathematical connection. *Sankhya B*, 75.

Miika Pihlaja, Michael Gutmann, and Aapo Hyvärinen. 2012. A family of computationally efficient and simple estimators for unnormalized statistical models. *CoRR*, abs/1203.3506.

David M. W. Powers. 1998. Applications and explanations of Zipf's law. In *New Methods in Language Processing and Computational Natural Language Learning*.

Alfréd Rényi. 1961. On measures of entropy and information. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*, pages 547–561, Berkeley, Calif. University of California Press.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Takashi Takenouchi and Takafumi Kanamori. 2015. Empirical localization of homogeneous divergences on discrete sample spaces. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 820–828. Curran Associates, Inc.

Vincent Van Asch and Walter Daelemans. 2010. Using domain similarity for performance estimation. In *Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing*, DANLP 2010, pages 31–36, Stroudsburg, PA, USA. Association for Computational Linguistics.