Language Modeling with Power Low Rank Ensembles

Ankur P. Parikh
School of Computer Science
Carnegie Mellon University
apparikh@cs.cmu.edu

Avneesh Saluja
Electrical & Computer Engineering
Carnegie Mellon University
avneesh@cs.cmu.edu

Chris Dyer
School of Computer Science
Carnegie Mellon University
cdyer@cs.cmu.edu

Eric P. Xing
School of Computer Science
Carnegie Mellon University
epxing@cs.cmu.edu

Abstract

We present power low rank ensembles (PLRE), a flexible framework for n-gram language modeling where ensembles of low rank matrices and tensors are used to obtain smoothed probability estimates of words in context. Our method can be understood as a generalization of n-gram modeling to non-integer n, and includes standard techniques such as absolute discounting and Kneser-Ney smoothing as special cases. PLRE training is efficient and our approach outperforms state-of-the-art modified Kneser Ney baselines in terms of perplexity on large corpora as well as on BLEU score in a downstream machine translation task.

1 Introduction

Language modeling is the task of estimating the probability of sequences of words in a language and is an important component in, among other applications, automatic speech recognition (Rabiner and Juang, 1993) and machine translation (Koehn, 2010). The predominant approach to language modeling is the n-gram model, wherein the probability of a word sequence $P(w_1, \ldots, w_\ell)$ is decomposed using the chain rule, and then a Markov assumption is made: $P(w_1, \ldots, w_\ell) \approx \prod_{i=1}^{\ell} P(w_i | w_{i-n+1}^{i-1})$. While this assumption substantially reduces the modeling complexity, parameter estimation remains a major challenge. Due to the power-law nature of language (Zipf, 1949), the maximum likelihood estimator massively overestimates the probability of rare events and assigns zero probability to legitimate word sequences that happen not to have been observed in the training data (Manning and Schütze, 1999).

Many smoothing techniques have been proposed to address the estimation challenge. These reassign probability mass (generally from overestimated events) to unseen word sequences, whose probabilities are estimated by interpolating with or backing off to lower order n-gram models (Chen and Goodman, 1999).

Somewhat surprisingly, these widely used smoothing techniques differ substantially from techniques for coping with data sparsity in other domains, such as collaborative filtering (Koren et al., 2009; Su and Khoshgoftaar, 2009) or matrix completion (Candès and Recht, 2009; Cai et al., 2010). In these areas, *low rank* approaches based on matrix factorization play a central role (Lee and Seung, 2001; Salakhutdinov and Mnih, 2008; Mackey et al., 2011). For example, in recommender systems, a key challenge is dealing with the sparsity of ratings from a single user, since typical users will have rated only a few items. By projecting the low rank representation of a user's (sparse) preferences into the original space, an estimate of ratings for new items is obtained. These methods are attractive due to their computational efficiency and mathematical well-foundedness.

In this paper, we introduce **power low rank ensembles** (PLRE), in which low rank tensors are used to produce smoothed estimates for n-gram probabilities. Ideally, we would like the low rank structures to discover semantic and syntactic relatedness among words and n-grams, which are used to produce smoothed estimates for word sequence probabilities. In contrast to the few previous low rank language modeling approaches, PLRE is not orthogonal to n-gram models, but rather a general framework where existing n-gram smoothing methods such as Kneser-Ney smoothing are special cases. A key insight is that PLRE does not compute low rank approximations of the original

joint count matrices (in the case of bigrams) or tensors i.e. multi-way arrays (in the case of 3-grams and above), but instead altered quantities of these counts based on an element-wise power operation, similar to how some smoothing methods modify their lower order distributions.

Moreover, PLRE has two key aspects that lead to easy scalability for large corpora and vocabularies. First, since it utilizes the original n-grams, the ranks required for the low rank matrices and tensors tend to be remain tractable (e.g. around 100 for a vocabulary size $V \approx 1 \times 10^6$) leading to fast training times. This differentiates our approach over other methods that leverage an underlying latent space such as neural networks (Bengio et al., 2003; Mnih and Hinton, 2007; Mikolov et al., 2010) or soft-class models (Saul and Pereira, 1997) where the underlying dimension is required to be quite large to obtain good performance. Moreover, at test time, the probability of a sequence can be queried in time $O(\kappa_{max})$ where κ_{max} is the maximum rank of the low rank matrices/tensors used. While this is larger than Kneser Ney's virtually constant query time, it is substantially faster than conditional exponential family models (Chen and Rosenfeld, 2000; Chen, 2009; Nelakanti et al., 2013) and neural networks which require $O(V)$ for exact computation of the normalization constant. See Section 7 for a more detailed discussion of related work.

Outline: We first review existing n-gram smoothing methods (§2) and then present the intuition behind the key components of our technique: **rank** (§3.1) and **power** (§3.2). We then show how these can be interpolated into an ensemble (§4). In the experimental evaluation on English and Russian corpora (§5), we find that PLRE outperforms Kneser-Ney smoothing and all its variants, as well as class-based language models. We also include a comparison to the log-bilinear neural language model (Mnih and Hinton, 2007) and evaluate performance on a downstream machine translation task (§6) where our method achieves consistent improvements in BLEU.

2 Discount-based Smoothing

We first provide background on absolute discounting (Ney et al., 1994) and Kneser-Ney smoothing (Kneser and Ney, 1995), two common n-gram smoothing methods. Both methods can be formulated as back-off or interpolated models; we describe the latter here since that is the basis of our low rank approach.

2.1 Notation

Let $c(w)$ be the count of word w, and similarly $c(w, w_{i-1})$ for the joint count of words w and w_{i-1}. For shorthand we will define w_i^j to denote the word sequence $\{w_i, w_{i+1}, ..., w_{j-1}, w_j\}$. Let $\widehat{P}(w_i)$ refer to the maximum likelihood estimate (MLE) of the probability of word w_i, and similarly $\widehat{P}(w_i|w_{i-1})$ for the probability conditioned on a history, or more generally, $\widehat{P}(w_i|w_{i-n+1}^{i-1})$.

Let $N_-(w_i) := |\{w : c(w_i, w) > 0\}|$ be the number of distinct words that appear before w_i. More generally, let $N_-(w_{i-n+1}^i) = |\{w : c(w_{i-n+1}^i, w) > 0\}|$. Similarly, let $N_+(w_{i-n+1}^{i-1}) = |\{w : c(w, w_{i-n+1}^{i-1}) > 0\}|$. V denotes the vocabulary size.

2.2 Absolute Discounting

Absolute discounting works on the idea of interpolating higher order n-gram models with lower-order n-gram models. However, first some probability mass must be "subtracted" from the higher order n-grams so that the leftover probability can be allocated to the lower order n-grams. More specifically, define the following discounted conditional probability:

$$\widehat{P}_D(w_i|w_{i-n+1}^{i-1}) = \frac{\max\{c(w_i, w_{i-n+1}^{i-1}) - D, 0\}}{c(w_{i-n+1}^{i-1})}$$

Then absolute discounting $P_{\text{abs}}(\cdot)$ uses the following (recursive) equation:

$$P_{\text{abs}}(w_i|w_{i-n+1}^{i-1}) = \widehat{P}_D(w_i|w_{i-n+1}^{i-1}) + \gamma(w_{i-n+1}^{i-1}) P_{\text{abs}}(w_i|w_{i-n+2}^{i-1})$$

where $\gamma(w_{i-n+1}^{i-1})$ is the leftover weight (due to the discounting) that is chosen so that the conditional distribution sums to one: $\gamma(w_{i-n+1}^{i-1}) = \frac{D}{c(w_{i-n+1}^{i-1})} N_+(w_{i-n+1}^{i-1})$. For the base case, we set $P_{\text{abs}}(w_i) = \widehat{P}(w_i)$.

Discontinuity: Note that if $c(w_{i-n+1}^{i-1}) = 0$, then $\gamma(w_{i-n+1}^{i-1}) = \frac{0}{0}$, in which case $\gamma(w_{i-n+1}^{i-1})$ is set to 1. We will see that this discontinuity appears in PLRE as well.

2.3 Kneser Ney Smoothing

Ideally, the smoothed probability should preserve the observed unigram distribution:

$$\widehat{P}(w_i) = \sum_{w_{i-n+1}^{i-1}} P_{\text{sm}}(w_i|w_{i-n+1}^{i-1})\widehat{P}(w_{i-n+1}^{i-1}) \quad (1)$$

where $P_{\text{sm}}(w_i|w_{i-n+1}^{i-1})$ is the smoothed conditional probability that a model outputs. Unfortunately, absolute discounting does not satisfy this property, since it exclusively uses the unaltered MLE unigram model as its lower order model. In practice, the lower order distribution is only utilized when we are unsure about the higher order distribution (i.e., when $\gamma(\cdot)$ is large). Therefore, the unigram model should be altered to condition on this fact.

This is the inspiration behind Kneser-Ney (KN) smoothing, an elegant algorithm with robust performance in n-gram language modeling. KN smoothing defines alternate probabilities $P^{\text{alt}}(\cdot)$:

$$P_D^{\text{alt}}(w_i|w_{i-n'+1}^{i-1}) = \begin{cases} \widehat{P}_D(w_i|w_{i-n'+1}^{i-1}), & \text{if } n' = n \\ \frac{\max\{N_-(w_{i-n'+1}^{i})-D,0\}}{\sum_{w_i} N_-(w_{i-n'+1}^{i})}, & \text{if } n' < n \end{cases}$$

The base case for unigrams reduces to $P^{\text{alt}}(w_i) = \frac{N_-(w_i)}{\sum_{w_i} N_-(w_i)}$. Intuitively $P^{\text{alt}}(w_i)$ is proportional to the number of unique words that precede w_i. Thus, words that appear in many different contexts will be given higher weight than words that consistently appear after only a few contexts. These alternate distributions are then used with absolute discounting:

$$P_{\text{kn}}(w_i|w_{i-n+1}^{i-1}) = P_D^{\text{alt}}(w_i|w_{i-n+1}^{i-1}) + \gamma(w_{i-n+1}^{i-1})P_{\text{kn}}(w_i|w_{i-n+2}^{i-1}) \quad (2)$$

where we set $P_{\text{kn}}(w_i) = P^{\text{alt}}(w_i)$. By definition, KN smoothing satisfies the marginal constraint in Eq. 1 (Kneser and Ney, 1995).

3 Power Low Rank Ensembles

In n-gram smoothing methods, if a bigram count $c(w_i, w_{i-1})$ is zero, the unigram probabilities are used, which is equivalent to assuming that w_i and w_{i-1} are independent (and similarly for general n). However, in this situation, instead of backing off to a 1-gram, we may like to back off to a "1.5-gram" or more generally an order between 1 and 2 that captures a coarser level of dependence between w_i and w_{i-1} and does not assume full independence.

Inspired by this intuition, our strategy is to construct an ensemble of matrices and tensors that not only consists of MLE-based count information, but also contains quantities that represent levels of dependence in-between the various orders in the model. We call these combinations power low rank ensembles (PLRE), and they can be thought of as n-gram models with non-integer n. Our approach can be recursively formulated as:

$$\begin{aligned} P_{\text{plre}}(w_i|w_{i-n+1}^{i-1}) &= P_{\boldsymbol{D}_0}^{\text{alt}}(w_i|w_{i-n+1}^{i-1}) \\ &+ \gamma_0(w_{i-n+1}^{i-1})\Big(\boldsymbol{Z}_{\boldsymbol{D}_1}(w_i|w_{i-n+1}^{i-1}) + ... \\ &+ \gamma_{\eta-1}(w_{i-n+1}^{i-1})\Big(\boldsymbol{Z}_{\boldsymbol{D}_\eta}(w_i|w_{i-n+1}^{i-1}) \\ &+ \gamma_\eta(w_{i-n+1}^{i-1})\Big(P_{\text{plre}}(w_i|w_{i-n+2}^{i-1})\Big)\Big)...\Big) \end{aligned} \quad (3)$$

where $\boldsymbol{Z}_1, ..., \boldsymbol{Z}_\eta$ are conditional probability matrices that represent the intermediate n-gram orders[1] and $\boldsymbol{D}$ is a discount function (specified in §4).

This formulation begs answers to a few critical questions. How to construct matrices that represent conditional probabilities for intermediate n? How to transform them in a way that generalizes the altered lower order distributions in KN smoothing? How to combine these matrices such that the marginal constraint in Eq. 1 still holds? The following propose solutions to these three queries:

1. **Rank** (Section 3.1): *Rank* gives us a concrete measurement of the dependence between w_i and w_{i-1}. By constructing low rank approximations of the bigram count matrix and higher-order count tensors, we obtain matrices that represent coarser dependencies, with a rank one approximation implying that the variables are independent.

2. **Power** (Section 3.2): In KN smoothing, the lower order distributions are not the original counts but rather altered estimates. We propose a continuous generalization of this alteration by taking the element-wise *power* of the counts.

[1] with a slight abuse of notation, let $\boldsymbol{Z}_{\boldsymbol{D}_j}$ be shorthand for $\boldsymbol{Z}_{j,\boldsymbol{D}_j}$

3. **Creating the Ensemble** (Section 4): Lastly, PLRE also defines a way to interpolate the specifically constructed intermediate n-gram matrices. Unfortunately a constant discount, as presented in Section 2, will not in general preserve the lower order marginal constraint (Eq. 1). We propose a generalized discounting scheme to ensure the constraint holds.

3.1 Rank

We first show how rank can be utilized to construct quantities between an n-gram and an $n-1$-gram. In general, we think of an n-gram as an n^{th} order tensor i.e. a multi-way array with n indices $\{i_1, ..., i_n\}$. (A vector is a tensor of order 1, a matrix is a tensor of order 2 etc.) Computing a special rank one approximation of slices of this tensor produces the $n-1$-gram. Thus, taking rank κ approximations in this fashion allows us to represent dependencies between an n-gram and $n-1$-gram.

Consider the bigram count matrix $\boldsymbol{B}$ with N counts which has rank V. Note that $\widehat{P}(w_i|w_{i-1}) = \frac{\boldsymbol{B}(w_i,w_{i-1})}{\sum_w \boldsymbol{B}(w,w_{i-1})}$. Additionally, $\boldsymbol{B}$ can be considered a random variable that is the result of sampling N tuples of (w_i, w_{i-1}) and agglomerating them into a count matrix. Assuming w_i and w_{i-1} are independent, the expected value (with respect to the empirical distribution) $\mathbb{E}[\boldsymbol{B}] = NP(w_i)P(w_{i-1})$, which can be rewritten as being proportional to the outer product of the unigram probability vector with itself, and is thus rank one.

This observation extends to higher order n-grams as well. Let $\boldsymbol{C}^n$ be the n^{th} order tensor where $\boldsymbol{C}^n(w_i, ..., w_{i-n+1}) = c(w_i, ..., w_{i-n+1})$. Furthermore denote $\boldsymbol{C}^n(:, \tilde{w}_{i-n+2}^{i-1}, :)$ to be the $V \times V$ matrix slice of $\boldsymbol{C}^n$ where $w_{i-n+2}, ..., w_{i-1}$ are held fixed to a particular sequence $\tilde{w}_{i-n+2}, ..., \tilde{w}_{i-1}$. Then if w_i is conditionally independent of w_{i-n+1} given w_{i-n+2}^{i-1}, then $\mathbb{E}[\boldsymbol{C}^n(:, \tilde{w}_{i-n+2}^{i-1}, :)]$ is rank one $\forall \tilde{w}_{i-n+2}^{i-1}$.

However, it is rare that these matrices are actually rank one, either due to sampling variance or the fact that w_i and w_{i-1} are not independent. What we would really like to say is that the *best* rank one approximation $\boldsymbol{B}^{(1)}$ (under some norm) of $\boldsymbol{B}$ is $\propto \widehat{P}(w_i)\widehat{P}(w_{i-1})$. While this statement is not true under the ℓ_2 norm, it is true under generalized KL divergence (Lee and Seung, 2001): $gKL(\boldsymbol{A}||\boldsymbol{B}) = \sum_{ij}\Big(\boldsymbol{A}_{ij}\log(\frac{\boldsymbol{A}_{ij}}{\boldsymbol{B}_{ij}}) - \boldsymbol{A}_{ij} + \boldsymbol{B}_{ij})\Big)$.

In particular, generalized KL divergence preserves row and column sums: *if $\boldsymbol{M}^{(\kappa)}$ is the best rank κ approximation of $\boldsymbol{M}$ under gKL then the row sums and column sums of $\boldsymbol{M}^{(\kappa)}$ and $\boldsymbol{M}$ are equal* (Ho and Van Dooren, 2008). Leveraging this property, it is straightforward to prove the following lemma:

Lemma 1. *Let $\boldsymbol{B}^{(\kappa)}$ be the best rank κ approximation of $\boldsymbol{B}$ under gKL. Then $\boldsymbol{B}^{(1)} \propto \widehat{P}(w_i)\widehat{P}(w_{i-1})$ and $\forall w_{i-1}$ s.t. $c(w_{i-1}) \neq 0$:*

$$\widehat{P}(w_i) = \frac{\boldsymbol{B}^{(1)}(w_i, w_{i-1})}{\sum_w \boldsymbol{B}^{(1)}(w, w_{i-1})}$$

For more general n, let $\boldsymbol{C}^{n,(\kappa)}_{i-1,...,i-n+2}$ be the best rank κ approximation of $\boldsymbol{C}^n(:, \tilde{w}_{i-n+2}^{i-1}, :)$ under gKL. Then similarly, $\forall w_{i-n+1}^{i-1}$ s.t. $c(w_{i-n+1}^{i-1}) > 0$:

$$\widehat{P}(w_i|w_{i-1}, ..., w_{i-n+2}) = \frac{\boldsymbol{C}^{n,(1)}_{i-1,...,i-n+2}(w_i, w_{i-n+1}^{i-1})}{\sum_w \boldsymbol{C}^{n,(1)}_{i-1,...,i-n+2}(w, w_{i-n+1}^{i-1})} \quad (4)$$

Thus, by selecting $1 < \kappa < V$, we obtain count matrices and tensors between n and $n-1$-grams. The condition that $c(w_{i-n+1}^{i-1}) > 0$ corresponds to the discontinuity discussed in §2.2.

3.2 Power

Since KN smoothing alters the lower order distributions instead of simply using the MLE, varying the rank is not sufficient in order to generalize this suite of techniques. Thus, PLRE computes low rank approximations of altered count matrices. Consider taking the elementwise power ρ of the bigram count matrix, which is denoted by $\boldsymbol{B}^{\cdot\rho}$. For example, the observed bigram count matrix and associated row sum:

$$\boldsymbol{B}^{\cdot 1} = \begin{pmatrix} 1.0 & 2.0 & 1.0 \\ 0 & 5.0 & 0 \\ 2.0 & 0 & 0 \end{pmatrix} \overset{\text{row sum}}{\rightarrow} \begin{pmatrix} 4.0 \\ 5.0 \\ 2.0 \end{pmatrix}$$

As expected the row sum is equal to the unigram counts (which we denote as $\boldsymbol{u}$). Now consider $\boldsymbol{B}^{\cdot 0.5}$:

$$\boldsymbol{B}^{\cdot 0.5} = \begin{pmatrix} 1.0 & 1.4 & 1.0 \\ 0 & 2.2 & 0 \\ 1.4 & 0 & 0 \end{pmatrix} \overset{\text{row sum}}{\rightarrow} \begin{pmatrix} 3.4 \\ 2.2 \\ 1.4 \end{pmatrix}$$

Note how the row sum vector has been altered. In particular since w_1 (corresponding to the first

row) has a more diverse history than w_2, it has a higher row sum (compared to in $\boldsymbol{u}$ where w_2 has the higher row sum). Lastly, consider the case when $p = 0$:

$$\boldsymbol{B}^{\cdot 0} = \begin{pmatrix} 1.0 & 1.0 & 1.0 \\ 0 & 1.0 & 0 \\ 1.0 & 0 & 0 \end{pmatrix} \overset{\text{row sum}}{\rightarrow} \begin{pmatrix} 3.0 \\ 1.0 \\ 1.0 \end{pmatrix}$$

The row sum is now the number of unique words that precede w_i (since $\boldsymbol{B}^0$ is binary) and is thus equal to the (unnormalized) Kneser Ney unigram. This idea also generalizes to higher order n-grams and leads us to the following lemma:

Lemma 2. *Let $\boldsymbol{B}^{(\rho,\kappa)}$ be the best rank κ approximation of $\boldsymbol{B}^{\cdot\rho}$ under gKL. Then $\forall w_{i-1}$ s.t. $c(w_{i-1}) \neq 0$:*

$$P^{alt}(w_i) = \frac{\boldsymbol{B}^{(0,1)}(w_i, w_{i-1})}{\sum_w \boldsymbol{B}^{(0,1)}(w, w_{i-1})}$$

For more general n, let $\boldsymbol{C}^{n,(\rho,\kappa)}_{i-1,\ldots,i-n+2}$ be the best rank κ approximation of $\boldsymbol{C}^{n,(\rho)}(:, \tilde{w}^{i-1}_{i-n+2}, :)$ under gKL. Similarly, $\forall w^{i-1}_{i-n+1}$ s.t. $c(w^{i-1}_{i-n+1}) > 0$:

$$P^{alt}(w_i|w_{i-1}, ..., w_{i-n+2}) = \frac{\boldsymbol{C}^{n,(0,1)}_{i-1,\ldots,i-n+2}(w_i, w^{i-1}_{i-n+1})}{\sum_w \boldsymbol{C}^{n,(0,1)}_{i-1,\ldots,i-n+2}(w, w^{i-1}_{i-n+1})} \quad (5)$$

4 Creating the Ensemble

Recall our overall formulation in Eq. 3; a naive solution would be to set $\boldsymbol{Z}_1, ..., \boldsymbol{Z}_\eta$ to low rank approximations of the count matrices/tensors under varying powers, and then interpolate through constant absolute discounting. Unfortunately, the marginal constraint in Eq. 1 will generally not hold if this strategy is used. Therefore, we propose a generalized discounting scheme where each nonzero n-gram count is associated with a different discount $\boldsymbol{D}_j(w_i, w^{i-1}_{i-n'+1})$. The low rank approximations are then computed on the discounted matrices, leaving the marginal constraint intact.

For clarity of exposition, we focus on the special case where $n = 2$ with only one low rank matrix before stating our general algorithm:

$$P_{\text{plre}}(w_i|w_{i-1}) = \widehat{P}_{\boldsymbol{D}_0}(w_i|w_{i-1}) + \gamma_0(w_{i-1})\Big(\boldsymbol{Z}_{\boldsymbol{D}_1}(w_i|w_{i-1}) + \gamma_1(w_{i-1})P^{\text{alt}}(w_i)\Big) \quad (6)$$

Our goal is to compute $\boldsymbol{D}_0, \boldsymbol{D}_1$ and $\boldsymbol{Z}_1$ so that the following lower order marginal constraint holds:

$$\widehat{P}(w_i) = \sum_{w_{i-1}} P_{\text{plre}}(w_i|w_{i-1})\widehat{P}(w_{i-1}) \quad (7)$$

Our solution can be thought of as a two-step procedure where we compute the discounts $\boldsymbol{D}_0, \boldsymbol{D}_1$ (and the $\gamma(w_{i-1})$ weights as a by-product), followed by the low rank quantity $\boldsymbol{Z}_1$. First, we construct the following intermediate ensemble of powered, but full rank terms. Let $\boldsymbol{Y}^{\rho_j}$ be the matrix such that $\boldsymbol{Y}^{\rho_j}(w_i, w_{i-1}) := c(w_i, w_{i-1})^{\rho_j}$. Then define

$$P_{\text{pwr}}(w_i|w_{i-1}) := \boldsymbol{Y}^{(\rho_0=1)}_{\boldsymbol{D}_0}(w_i|w_{i-1}) + \gamma_0(w_{i-1})\Big(\boldsymbol{Y}^{(\rho_1)}_{\boldsymbol{D}_1}(w_i|w_{i-1}) + \gamma_1(w_{i-1})\boldsymbol{Y}^{(\rho_2=0)}(w_i|w_{i-1})\Big) \quad (8)$$

where with a little abuse of notation:

$$\boldsymbol{Y}^{\rho_j}_{\boldsymbol{D}_j}(w_i|w_{i-1}) = \frac{c(w_i, w_{i-1})^{\rho_j} - \boldsymbol{D}_j(w_i, w_{i-1})}{\sum_{w_i} c(w_i, w_{i-1})^{\rho_j}}$$

Note that $P^{\text{alt}}(w_i)$ has been replaced with $\boldsymbol{Y}^{(\rho_2=0)}(w_i|w_{i-1})$, based on Lemma 2, and will equal $P^{\text{alt}}(w_i)$ once the low rank approximation is taken as discussed in § 4.2).

Since we have only combined terms of different power (but all full rank), it is natural choose the discounts so that the result remains unchanged i.e., $P_{\text{pwr}}(w_i|w_{i-1}) = \widehat{P}(w_i|w_{i-1})$, since the low rank approximation (not the power) will implement smoothing. Enforcing this constraint gives rise to a set of linear equations that can be solved (in closed form) to obtain the discounts as we now show below.

4.1 Step 1: Computing the Discounts

To ensure the constraint that $P_{\text{pwr}}(w_i|w_{i-1}) = \widehat{P}(w_i|w_{i-1})$, it is sufficient to enforce the following two local constraints:

$$\boldsymbol{Y}^{(\rho_j)}(w_i|w_{i-1}) = \boldsymbol{Y}^{(\rho_j)}_{\boldsymbol{D}_j}(w_i|w_{i-1}) + \gamma_j(w_{i-1})\boldsymbol{Y}^{(\rho_{j+1})}(w_i|w_{i-1}) \text{ for } j = 0, 1 \quad (9)$$

This allows each $\boldsymbol{D}_j$ to be solved for independently of the other $\{\boldsymbol{D}_{j'}\}_{j' \neq j}$. Let $c_{i,i-1} = c(w_i, w_{i-1})$, $c^j_{i,i-1} = c(w_i, w_{i-1})^{\rho_j}$, and $d^j_{i,i-1} =$

$\boldsymbol{D}_j(w_i, w_{i-1})$. Expanding Eq. 9 yields that $\forall w_i, w_{i-1}$:

$$\frac{c^j_{i,i-1}}{\sum_i c^j_{i,i-1}} = \frac{c^j_{i,i-1} - d^j_{i,i-1}}{\sum_i c^j_{i,i-1}} + \left(\frac{\sum_i d^j_{i,i-1}}{\sum_i c^j_{i,i-1}}\right) \frac{c^{j+1}_{i,i-1}}{\sum_i c^{j+1}_{i,i-1}} \quad (10)$$

which can be rewritten as:

$$-d^j_{i,i-1} + \left(\sum_i d^j_{i,i-1}\right) \frac{c^{j+1}_{i,i-1}}{\sum_i c^{j+1}_{i,i-1}} = 0 \quad (11)$$

Note that Eq. 11 decouples across w_{i-1} since the only $d^j_{i,i-1}$ terms that are dependent are the ones that share the preceding context w_{i-1}.

It is straightforward to see that setting $d^j_{i,i-1}$ proportional to $c^{j+1}_{i,i-1}$ satisfies Eq. 11. Furthermore it can be shown that all solutions are of this form (i.e., the linear system has a null space of exactly one). Moreover, we are interested in a particular subset of solutions where a single parameter d_* (independent of w_{i-1}) controls the scaling as indicated by the following lemma:

Lemma 3. *Assume that $\rho_j \geq \rho_{j+1}$. Choose any $0 \leq d_* \leq 1$. Set $d^j_{i,i-1} = d_* c^{j+1}_{i,i-1}\ \forall i, j$. The resulting discounts satisfy Eq. 11 as well as the inequality constraints $0 \leq d^j_{i,i-1} \leq c^j_{i,i-1}$. Furthermore, the leftover weight γ_j takes the form:*

$$\gamma_j(w_{i-1}) = \frac{\sum_i d^j_{i,i-1}}{\sum_i c^j_{i,i-1}} = \frac{d_* \sum_i c^{j+1}_{i,i-1}}{\sum_i c^j_{i,i-1}}$$

Proof. Clearly this choice of $d^j_{i,i-1}$ satisfies Eq. 11. The largest possible value of $d^j_{i,i-1}$ is $c^{j+1}_{i,i-1}$. $\rho_j \geq \rho_{j+1}$, implies $c^j_{i,i-1} \geq c^{j+1}_{i,i-1}$. Thus the inequality constraints are met. It is then easy to verify that γ takes the above form. □

The above lemma generalizes to longer contexts (i.e. $n > 2$) as shown in Algorithm 1. Note that if $\rho_j = \rho_{j+1}$ then Algorithm 1 is equivalent to scaling the counts e.g. deleted-interpolation/Jelinek Mercer smoothing (Jelinek and Mercer, 1980). On the other hand, when $\rho_{j+1} = 0$, Algorithm 1 is equal to the absolute discounting that is used in Kneser-Ney. Thus, depending on ρ_{j+1}, our method generalizes different types of interpolation schemes to construct an ensemble so that the marginal constraint is satisfied.

Algorithm 1 Compute $\boldsymbol{D}$

In: Count tensor $\boldsymbol{C}^n$, powers ρ_j, ρ_{j+1} such that $\rho_j \geq \rho_{j+1}$, and parameter d_*.

Out: Discount $\boldsymbol{D}_j$ for powered counts $\boldsymbol{C}^{n,(\rho_j)}$ and associated leftover weight γ_j

1: Set $\boldsymbol{D}_j(w_i, w_{i-n+1}^{i-1}) = d_* c(w_i, w_{i-n+1}^{i-1})^{\rho_{j+1}}$.

2: $$\gamma_j(w_i, w_{i-n+1}^{i-1}) = \frac{d_* \sum_{w_i} c(w_i, w_{i-n+1}^{i-1})^{\rho_{j+1}}}{\sum_{w_i} c(w_i, w_{i-n+1}^{i-1})^{\rho_j}}$$

Algorithm 2 Compute $\boldsymbol{Z}$

In: Count tensor $\boldsymbol{C}^n$, power ρ, discounts $\boldsymbol{D}$, rank κ

Out: Discounted low rank conditional probability table $\boldsymbol{Z}_{\boldsymbol{D}}^{(\rho,\kappa)}(w_i|w_{i-n+1}^{i-1})$ (represented implicitly)

1: Compute powered counts $\boldsymbol{C}^{n,(\cdot\rho)}$.

2: Compute denominators $\sum_{w_i} c(w_i, w_{i-n+1}^{i-1})^\rho$ $\forall w_{i-n+1}^{i-1}$ s.t. $c(w_{i-n+1}^{i-1}) > 0$.

3: Compute discounted powered counts $\boldsymbol{C}_{\boldsymbol{D}}^{n,(\cdot\rho)} = \boldsymbol{C}^{n,(\cdot\rho)} - \boldsymbol{D}$.

4: For each slice $\boldsymbol{M}_{\tilde{w}_{i-n+2}^{i-1}} := \boldsymbol{C}_{\boldsymbol{D}}^{n,(\cdot\rho)}(:, \tilde{w}_{i-n+2}^{i-1}, :)$ compute

$$\boldsymbol{M}^{(\kappa)} := \min_{\boldsymbol{A} \geq 0 : rank(\boldsymbol{A}) = \kappa} \|\boldsymbol{M}_{\tilde{w}_{i-n+2}^{i-1}} - \boldsymbol{A}\|_{KL}$$

(stored implicitly as $\boldsymbol{M}^{(\kappa)} = \boldsymbol{LR}$)

Set $\boldsymbol{Z}_{\boldsymbol{D}}^{(\rho,\kappa)}(:, \tilde{w}_{i-n+2}^{i-1}, :) = \boldsymbol{M}^{(\kappa)}$

5: Note that

$$\boldsymbol{Z}_{\boldsymbol{D}}^{(\rho,\kappa)}(w_i|w_{i-n+1}^{i-1}) = \frac{\boldsymbol{Z}_{\boldsymbol{D}}^{(\rho,\kappa)}(w_i, w_{i-n+1}^{i-1})}{\sum_{w_i} c(w_i, w_{i-n+1}^{i-1})^\rho}$$

4.2 Step 2: Computing Low Rank Quantities

The next step is to compute low rank approximations of $\boldsymbol{Y}_{\boldsymbol{D}_j}^{(\rho_j)}$ to obtain $\boldsymbol{Z}_{\boldsymbol{D}_j}$ such that the intermediate marginal constraint in Eq. 7 is preserved. This constraint trivially holds for the intermediate ensemble $P_{\text{pwr}}(w_i|w_{i-1})$ due to how the discounts were derived in § 4.1. For our running bigram example, define $\boldsymbol{Z}_{\boldsymbol{D}_j}^{(\rho_j,\kappa_j)}$ to be the best rank κ_j approximation to $\boldsymbol{Y}_{\boldsymbol{D}_j}^{(\rho_j,\kappa_j)}$ according to gKL and let

$$\boldsymbol{Z}_{\boldsymbol{D}_j}^{\rho_j,\kappa_j}(w_i|w_{i-1}) = \frac{\boldsymbol{Z}_{\boldsymbol{D}_j}^{\rho_j,\kappa_j}(w_i, w_{i-1})}{\sum_{w_i} c(w_i, w_{i-1})^{\rho_j}}$$

Note that $\boldsymbol{Z}_{\boldsymbol{D}_j}^{\rho_j,\kappa_j}(w_i|w_{i-1})$ is a valid (discounted) conditional probability since gKL preserves row/column sums so the denominator remains unchanged under the low rank approximation. Then

using the fact that $\boldsymbol{Z}^{(0,1)}(w_i|w_{i-1}) = P^{\text{alt}}(w_i)$ (Lemma 2) we can embellish Eq. 6 as

$$P_{\text{plre}}(w_i|w_{i-1}) = P_{\boldsymbol{D}_0}(w_i|w_{i-1}) + \gamma_0(w_{i-1})\left(\boldsymbol{Z}_{\boldsymbol{D}_1}^{(\rho_1,\kappa_1)}(w_i|w_{i-1}) + \gamma_1(w_{i-1})P_{\text{alt}}(w_i)\right)$$

Leveraging the form of the discounts and row/column sum preserving property of gKL, we then have the following lemma (the proof is in the supplementary material):

Lemma 4. *Let $P_{plre}(w_i|w_{i-1})$ indicate the PLRE smoothed conditional probability as computed by Eq. 6 and Algorithms 1 and 2. Then, the marginal constraint in Eq. 7 holds.*

4.3 More general algorithm

In general, the principles outlined in the previous sections hold for higher order n-grams. Assume that the discounts are computed according to Algorithm 1 with parameter d_* and $\boldsymbol{Z}_{\boldsymbol{D}_j}^{(\rho_j,\kappa_j)}$ is computed according to Algorithm 2. Note that, as shown in Algorithm 2, for higher order n-grams, the $\boldsymbol{Z}_{\boldsymbol{D}_j}^{(\rho_j,\kappa_j)}$ are created by taking low rank approximations of slices of the (powered) count tensors (see Lemma 2 for intuition). Eq. 3 can now be embellished:

$$\begin{aligned} P_{\text{plre}}(w_i|w_{i-n+1}^{i-1}) &= P_{\boldsymbol{D}_0}^{\text{alt}}(w_i|w_{i-n+1}^{i-1}) \\ &+ \gamma_0(w_{i-n+1}^{i-1})\left(\boldsymbol{Z}_{\boldsymbol{D}_1}^{(\rho_1,\kappa_1)}(w_i|w_{i-n+1}^{i-1}) + ...\right. \\ &+ \gamma_{\eta-1}(w_{i-n+1}^{i-1})\left(\boldsymbol{Z}_{\boldsymbol{D}_\eta}^{(\rho_\eta,\kappa_\eta)}(w_i|w_{i-n+1}^{i-1})\right. \\ &\left.\left. + \gamma_\eta(w_{i-n+1}^{i-1})\left(P_{\text{plre}}(w_i|w_{i-n+2}^{i-1})\right)\right)...\right) \quad (12) \end{aligned}$$

Lemma 4 also applies in this case and is given in Theorem 1 in the supplementary material.

4.4 Links with KN Smoothing

In this section, we explicitly show the relationship between PLRE and KN smoothing. Rewriting Eq. 12 in the following form:

$$\begin{aligned} P_{\text{plre}}(w_i|w_{i-n+1}^{i-1}) &= P_{\text{plre}}^{\text{terms}}(w_i|w_{i-n+1}^{i-1}) \\ &+\gamma_{0:\eta}(w_{i-n+1}^{i-1})P_{\text{plre}}(w_i|w_{i-n+2}^{i-1}) \quad (13) \end{aligned}$$

where $P_{\text{plre}}^{\text{terms}}(w_i|w_{i-n+1}^{i-1})$ contains the terms in Eq. 12 except the last, and $\gamma_{0:\eta}(w_{i-n+1}^{i-1}) = \prod_{h=0}^{\eta}\gamma_h(w_{i-n+1}^{i-1})$, we can leverage the form of the discount, and using the fact that $\rho_{\eta+1} = 0$[2]:

$$\gamma_{0:\eta}(w_{i-n-1}^{i-1}) = \frac{{d_*}^{\eta+1}N_+(w_{i-n+1}^{i-1})}{c(w_{i-n+1}^{i-1})}$$

With this form of $\gamma(\cdot)$, Eq. 13 is remarkably similar to KN smoothing (Eq. 2) if KN's discount parameter D is chosen to equal $(d_*)^{\eta+1}$.

The difference is that $P^{\text{alt}}(\cdot)$ has been replaced with the alternate estimate $P_{\text{plre}}^{\text{terms}}(w_i|w_{i-n+1}^{i-1})$, which have been enriched via the low rank structure. Since these alternate estimates were constructed via our ensemble strategy they contain both very fine-grained dependencies (the original n-grams) as well as coarser dependencies (the lower rank n-grams) and is thus fundamentally different than simply taking a single matrix/tensor decomposition of the trigram/bigram matrices.

Moreover, it provides a natural way of setting d_* based on the Good-Turing (GT) estimates employed by KN smoothing. In particular, we can set d_* to be the $(\eta+1)^{\text{th}}$ root of the KN discount D that can be estimated via the GT estimates.

4.5 Computational Considerations

PLRE scales well even as the order n increases. To compute a low rank bigram, one low rank approximation of a $V \times V$ matrix is required. For the low rank trigram, we need to compute a low rank approximation of each slice $\boldsymbol{C}_{\boldsymbol{D}}^{n,(\cdot p)}(:, \tilde{w}_{i-1}, :)\ \forall \tilde{w}_{i-1}$. While this may seem daunting at first, in practice the *size* of each slice (number of non-zero rows/columns) is usually much, much smaller than V, keeping the computation tractable.

Similarly, PLRE also evaluates conditional probabilities at evaluation time efficiently. As shown in Algorithm 2, the normalizer can be precomputed on the sparse powered matrix/tensor. As a result our test complexity is $\mathcal{O}(\sum_{i=1}^{\eta_{\text{total}}}\kappa_i)$ where η_{total} is the total number of matrices/tensors in the ensemble. While this is larger than Kneser Ney's practically constant complexity of $\mathcal{O}(n)$, it is much faster than other recent methods for language modeling such as neural networks and conditional exponential family models where exact computation of the normalizing constant costs $O(V)$.

5 Experiments

To evaluate PLRE, we compared its performance on English and Russian corpora with several vari-

[2] for derivation see proof of Lemma 4 in the supplementary material

ants of KN smoothing, class-based models, and the log-bilinear neural language model (Mnih and Hinton, 2007). We evaluated with perplexity in most of our experiments, but also provide results evaluated with BLEU (Papineni et al., 2002) on a downstream machine translation (MT) task. We have made the code for our approach publicly available [3].

To build the hard class-based LMs, we utilized `mkcls`[4], a tool to train word classes that uses the maximum likelihood criterion (Och, 1995) for classing. We subsequently trained trigram class language models on these classes (corresponding to 2nd-order HMMs) using SRILM (Stolcke, 2002), with KN-smoothing for the class transition probabilities. SRILM was also used for the baseline KN-smoothed models.

For our MT evaluation, we built a hierarchical phrase translation (Chiang, 2007) system using `cdec` (Dyer et al., 2010). The KN-smoothed models in the MT experiments were compiled using KenLM (Heafield, 2011).

5.1 Datasets

For the perplexity experiments, we evaluated our proposed approach on 4 datasets, 2 in English and 2 in Russian. In all cases, the singletons were replaced with "<unk>" tokens in the training corpus, and any word not in the vocabulary was replaced with this token during evaluation. There is a general dearth of evaluation on large-scale corpora in morphologically rich languages such as Russian, and thus we have made the processed Large-Russian corpus available for comparison [3].

- **Small-English**: APNews corpus (Bengio et al., 2003): Train - 14 million words, Dev - 963,000, Test - 963,000. Vocabulary- 18,000 types.
- **Small-Russian**: Subset of Russian news commentary data from 2013 WMT translation task[5]: Train- 3.5 million words, Dev - 400,000 Test - 400,000. Vocabulary - 77,000 types.
- **Large-English**: English Gigaword, Training - 837 million words, Dev - 8.7 million, Test - 8.7 million. Vocabulary- 836,980 types.
- **Large-Russian**: Monolingual data from WMT 2013 task. Training - 521 million words, Validation - 50,000, Test - 50,000. Vocabulary- 1.3 million types.

[3] http://www.cs.cmu.edu/∼apparikh/plre.html

[4] http://code.google.com/p/giza-pp/

[5] http://www.statmt.org/wmt13/training-monolingual-nc-v8.tgz

For the MT evaluation, we used the parallel data from the WMT 2013 shared task, excluding the Common Crawl corpus data. The newstest2012 and newstest2013 evaluation sets were used as the development and test sets respectively.

5.2 Small Corpora

For the class-based baseline LMs, the number of classes was selected from $\{32, 64, 128, 256, 512, 1024\}$ (Small-English) and $\{512, 1024\}$ (Small-Russian). We could not go higher due to the computationally laborious process of hard clustering. For Kneser-Ney, we explore four different variants: back-off (BO-KN) interpolated (int-KN), modified back-off (BO-MKN), and modified interpolated (int-MKN). Good-Turing estimates were used for discounts. All models trained on the small corpora are of order 3 (trigrams).

For PLRE, we used one low rank bigram and one low rank trigram in addition to the MLE n-gram estimates. The powers of the intermediate matrices/tensors were fixed to be 0.5 and the discounts were set to be square roots of the Good Turing estimates (as explained in § 4.4). The ranks were tuned on the development set. For Small-English, the ranges were $\{1e-3, 5e-3\}$ (as a fraction of the vocabulary size) for both the low rank bigram and low rank trigram models. For Small-Russian the ranges were $\{5e-4, 1e-3\}$ for both the low rank bigram and the low rank trigram models.

The results are shown in Table 1. The best class-based LM is reported, but is not competitive with the KN baselines. PLRE outperforms all of the baselines comfortably. Moreover, PLRE's performance over the baselines is highlighted in Russian. With larger vocabulary sizes, the low rank approach is more effective as it can capture linguistic similarities between rare and common words.

Next we discuss how the maximum n-gram order affects performance. Figure 1 shows the relative percentage improvement of our approach over int-MKN as the order is increased from 2 to 4 for both methods. The Small-English dataset has a rather small vocabulary compared to the number of tokens, leading to lower data sparsity in the bigram. Thus the PLRE improvement is small for order $= 2$, but more substantial for order $= 3$. On the other hand, for the Small-Russian dataset, the vocabulary size is much larger and consequently the bigram counts are sparser. This leads to sim-

Dataset	class-1024(3)	BO-KN(3)	int-KN(3)	BO-MKN(3)	int-MKN(3)	PLRE(3)
Small-English Dev	115.64	99.20	99.73	99.95	95.63	**91.18**
Small-English Test	119.70	103.86	104.56	104.55	100.07	**95.15**
Small-Russian Dev	286.38	281.29	265.71	287.19	263.25	**241.66**
Small-Russian Test	284.09	277.74	262.02	283.70	260.19	**238.96**

Table 1: Perplexity results on small corpora for all methods.

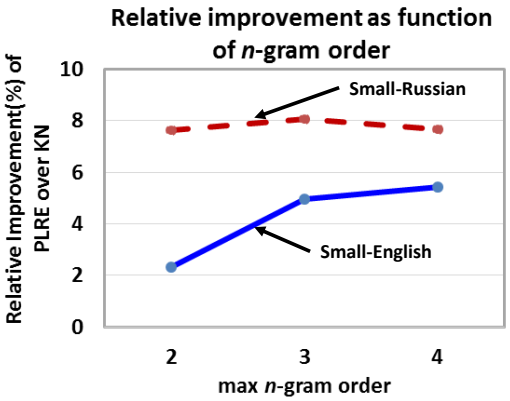

Figure 1: Relative percentage improvement of PLRE over int-MKN as the maximum n-gram order for both methods is increased.

ilar improvements for all orders (which are larger than that for Small-English).

On both these datasets, we also experimented with tuning the discounts for int-MKN to see if the baseline could be improved with more careful choices of discounts. However, this achieved only marginal gains (reducing the perplexity to 98.94 on the Small-English test set and 259.0 on the Small-Russian test set).

Comparison to LBL (Mnih and Hinton, 2007): Mnih and Hinton (2007) evaluate on the Small-English dataset (but remove end markers and concatenate the sentences). They obtain perplexities 117.0 and 107.8 using contexts of size 5 and 10 respectively. With this preprocessing, a 4-gram (context 3) PLRE achieves 108.4 perplexity.

5.3 Large Corpora

Results on the larger corpora for the top 2 performing methods "PLRE" and "int-MKN" are presented in Table 2. Due to the larger training size, we use 4-gram models in these experiments. However, including the low rank 4-gram tensor provided little gain and therefore, the 4-gram PLRE only has additional low rank bigram and low rank trigram matrices/tensors. As above, ranks were tuned on the development set. For Large-English, the ranges were $\{1e-4, 5e-4, 1e-3\}$ (as a fraction of the vocabulary size) for both the low rank bigram and low rank trigram models. For Small-Russian the ranges were $\{1e-5, 5e-5, 1e-4\}$ for both the low rank bigram and the low rank trigram models. For statistical validity, 10 test sets of size equal to the original test set were generated by randomly sampling sentences with replacement from the original test set. Our method outperforms "int-MKN" with gains similar to that on the smaller datasets. As shown in Table 3, our method obtains fast training times even for large datasets.

Dataset	int-MKN(4)	PLRE(4)
Large-English Dev	73.21	**71.21**
Large-English Test	77.90 $\pm$ 0.203	**75.66** $\pm$ 0.189
Large-Russian Dev	326.9	**297.11**
Large-Russian Test	289.63 $\pm$ 6.82	**264.59** $\pm$ 5.839

Table 2: Mean perplexity results on large corpora, with standard deviation.

Dataset	PLRE Training Time
Small-English	3.96 min (order 3) / 8.3 min (order 4)
Small-Russian	4.0 min (order 3) / 4.75 min (order 4)
Large-English	3.2 hrs (order 4)
Large-Russian	8.3 hrs (order 4)

Table 3: PLRE training times for a fixed parameter setting[6]. 8 Intel Xeon CPUs were used.

Method	BLEU
int-MKN(4)	17.63 $\pm$ 0.11
PLRE(4)	17.79 $\pm$ 0.07
Smallest Diff	PLRE+0.05
Largest Diff	PLRE+0.29

Table 4: Results on English-Russian translation task (mean $\pm$ stdev). See text for details.

6 Machine Translation Task

Table 4 presents results for the MT task, translating from English to Russian[7]. We used MIRA (Chiang et al., 2008) to learn the feature weights. To control for the randomness in MIRA, we avoid retuning when switching LMs - the set of feature weights obtained using int-MKN is the same, only the language model changes. The

[6] As described earlier, only the ranks need to be tuned, so only 2-3 low rank bigrams and 2-3 low rank trigrams need to be computed (and combined depending on the setting).

[7] the best score at WMT 2013 was 19.9 (Bojar et al., 2013)

procedure is repeated 10 times to control for optimizer instability (Clark et al., 2011). Unlike other recent approaches where an additional feature weight is tuned for the proposed model and used in conjunction with KN smoothing (Vaswani et al., 2013), our aim is to show the improvements that PLRE provides as a substitute for KN. On average, PLRE outperforms the KN baseline by 0.16 BLEU, and this improvement is consistent in that PLRE never gets a worse BLEU score.

7 Related Work

Recent attempts to revisit the language modeling problem have largely come from two directions: Bayesian nonparametrics and neural networks. Teh (2006) and Goldwater et al. (2006) discovered the connection between interpolated Kneser Ney and the hierarchical Pitman-Yor process. These have led to generalizations that account for domain effects (Wood and Teh, 2009) and unbounded contexts (Wood et al., 2009).

The idea of using neural networks for language modeling is not new (Miikkulainen and Dyer, 1991), but recent efforts (Mnih and Hinton, 2007; Mikolov et al., 2010) have achieved impressive performance. These methods can be quite expensive to train and query (especially as the vocabulary size increases). Techniques such as noise contrastive estimation (Gutmann and Hyvärinen, 2012; Mnih and Teh, 2012; Vaswani et al., 2013), subsampling (Xu et al., 2011), or careful engineering approaches for maximum entropy LMs (which can also be applied to neural networks) (Wu and Khudanpur, 2000) have improved training of these models, but querying the probability of the next word given still requires explicitly normalizing over the vocabulary, which is expensive for big corpora or in languages with a large number of word types. Mnih and Teh (2012) and Vaswani et al. (2013) propose setting the normalization constant to 1, but this is approximate and thus can only be used for downstream evaluation, not for perplexity computation. An alternate technique is to use word-classing (Goodman, 2001; Mikolov et al., 2011), which can reduce the cost of exact normalization to $O(\sqrt{V})$. In contrast, our approach is much more scalable, since it is trivially parallelized in training and does not require explicit normalization during evaluation.

There are a few low rank approaches (Saul and Pereira, 1997; Bellegarda, 2000; Hutchinson et al., 2011), but they are only effective in restricted settings (e.g. small training sets, or corpora divided into documents) and do not generally perform comparably to state-of-the-art models. Roark et al. (2013) also use the idea of marginal constraints for re-estimating back-off parameters for heavily-pruned language models, whereas we use this concept to estimate n-gram specific discounts.

8 Conclusion

We presented power low rank ensembles, a technique that generalizes existing n-gram smoothing techniques to non-integer n. By using ensembles of sparse as well as low rank matrices and tensors, our method captures both the fine-grained and coarse structures in word sequences. Our discounting strategy preserves the marginal constraint and thus generalizes Kneser Ney, and under slight changes can also extend other smoothing methods such as deleted-interpolation/Jelinek-Mercer smoothing. Experimentally, PLRE convincingly outperforms Kneser-Ney smoothing as well as class-based baselines.

Acknowledgements

This work was supported by NSF IIS1218282, NSF IIS1218749, NSF IIS1111142, NIH R01GM093156, the U. S. Army Research Laboratory and the U. S. Army Research Office under contract/grant number W911NF-10-1-0533, the NSF Graduate Research Fellowship Program under Grant No. 0946825 (NSF Fellowship to APP), and a grant from Ebay Inc. (to AS).

References

Jerome R. Bellegarda. 2000. Large vocabulary speech recognition with multispan statistical language models. *IEEE Transactions on Speech and Audio Processing*, 8(1):76–84.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3:1137–1155, March.

Ondřej Bojar, Christian Buck, Chris Callison-Burch, Christian Federmann, Barry Haddow, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. 2013. Findings of the 2013 Workshop on Statistical Machine Translation. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 1–44, Sofia, Bulgaria, August. Association for Computational Linguistics.

Jian-Feng Cai, Emmanuel J Candès, and Zuowei Shen. 2010. A singular value thresholding algorithm for

matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982.

Emmanuel J Candès and Benjamin Recht. 2009. Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717–772.

Stanley F. Chen and Joshua Goodman. 1999. An empirical study of smoothing techniques for language modeling. *Computer Speech & Language*, 13(4):359–393.

Stanley F Chen and Ronald Rosenfeld. 2000. A survey of smoothing techniques for me models. *Speech and Audio Processing, IEEE Transactions on*, 8(1):37–50.

Stanley F. Chen. 2009. Shrinking exponential language models. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, pages 468–476, Stroudsburg, PA, USA. Association for Computational Linguistics.

David Chiang, Yuval Marton, and Philip Resnik. 2008. Online large-margin training of syntactic and structural translation features. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 224–233. Association for Computational Linguistics.

David Chiang. 2007. Hierarchical phrase-based translation. *Comput. Linguist.*, 33(2):201–228, June.

Jonathan H. Clark, Chris Dyer, Alon Lavie, and Noah A. Smith. 2011. Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2*, HLT '11, pages 176–181.

Chris Dyer, Jonathan Weese, Hendra Setiawan, Adam Lopez, Ferhan Ture, Vladimir Eidelman, Juri Ganitkevitch, Phil Blunsom, and Philip Resnik. 2010. cdec: A decoder, alignment, and learning framework for finite-state and context-free translation models. In *Proceedings of the ACL 2010 System Demonstrations*, pages 7–12. Association for Computational Linguistics.

Sharon Goldwater, Thomas Griffiths, and Mark Johnson. 2006. Interpolating between types and tokens by estimating power-law generators. In *Advances in Neural Information Processing Systems*, volume 18.

Joshua Goodman. 2001. Classes for fast maximum entropy training. In *Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01). 2001 IEEE International Conference on*, volume 1, pages 561–564. IEEE.

Michael Gutmann and Aapo Hyvärinen. 2012. Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *Journal of Machine Learning Research*, 13:307–361.

Kenneth Heafield. 2011. KenLM: faster and smaller language model queries. In *Proceedings of the EMNLP 2011 Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland, United Kingdom, July.

Ngoc-Diep Ho and Paul Van Dooren. 2008. Non-negative matrix factorization with fixed row and column sums. *Linear Algebra and its Applications*, 429(5):1020–1025.

Brian Hutchinson, Mari Ostendorf, and Maryam Fazel. 2011. Low rank language models for small training sets. *Signal Processing Letters, IEEE*, 18(9):489–492.

Frederick Jelinek and Robert Mercer. 1980. Interpolated estimation of markov source parameters from sparse data. *Pattern recognition in practice*.

Reinhard Kneser and Hermann Ney. 1995. Improved backing-off for m-gram language modeling. In *Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on*, volume 1, pages 181–184. IEEE.

Philipp Koehn. 2010. *Statistical Machine Translation*. Cambridge University Press, New York, NY, USA, 1st edition.

Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37.

Daniel D. Lee and H. Sebastian Seung. 2001. Algorithms for non-negative matrix factorization. *Advances in Neural Information Processing Systems*, 13:556–562.

Lester Mackey, Ameet Talwalkar, and Michael I Jordan. 2011. Divide-and-conquer matrix factorization. *arXiv preprint arXiv:1107.0789*.

Christopher D Manning and Hinrich Schütze. 1999. *Foundations of statistical natural language processing*, volume 999. MIT Press.

Risto Miikkulainen and Michael G. Dyer. 1991. Natural language processing with modular pdp networks and distributed lexicon. *Cognitive Science*, 15:343–399.

Tom Mikolov, Martin Karafit, Luk Burget, Jan ernock, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Proceedings of the 11th Annual Conference of the International Speech Communication Association (INTERSPEECH 2010)*, volume 2010, pages 1045–1048. International Speech Communication Association.

Tomas Mikolov, Stefan Kombrink, Lukas Burget, JH Cernocky, and Sanjeev Khudanpur. 2011. Extensions of recurrent neural network language model. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 5528–5531. IEEE.

Andriy Mnih and Geoffrey Hinton. 2007. Three new graphical models for statistical language modelling. In *Proceedings of the 24th international conference on Machine learning*, pages 641–648. ACM.

Andriy Mnih and Yee Whye Teh. 2012. A fast and simple algorithm for training neural probabilistic language models. In *Proceedings of the International Conference on Machine Learning*.

Anil Kumar Nelakanti, Cedric Archambeau, Julien Mairal, Francis Bach, and Guillaume Bouchard. 2013. Structured penalties for log-linear language models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 233–243, Seattle, Washington, USA, October. Association for Computational Linguistics.

Hermann Ney, Ute Essen, and Reinhard Kneser. 1994. On Structuring Probabilistic Dependencies in Stochastic Language Modelling. *Computer Speech and Language*, 8:1–38.

Franz Josef Och. 1995. Maximum-likelihood-schätzung von wortkategorien mit verfahren der kombinatorischen optimierung. Bachelor's thesis (Studienarbeit), University of Erlangen.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. pages 311–318.

Lawrence Rabiner and Biing-Hwang Juang. 1993. Fundamentals of speech recognition.

Brian Roark, Cyril Allauzen, and Michael Riley. 2013. Smoothed marginal distribution constraints for language modeling. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 43–52.

Ruslan Salakhutdinov and Andriy Mnih. 2008. Bayesian probabilistic matrix factorization using Markov chain Monte Carlo. In *Proceedings of the 25th international conference on Machine learning*, pages 880–887. ACM.

Lawrence Saul and Fernando Pereira. 1997. Aggregate and mixed-order markov models for statistical language processing. In *Proceedings of the second conference on empirical methods in natural language processing*, pages 81–89. Somerset, New Jersey: Association for Computational Linguistics.

Andreas Stolcke. 2002. SRILM - An Extensible Language Modeling Toolkit. In *Proceedings of the International Conference in Spoken Language Processing*.

Xiaoyuan Su and Taghi M Khoshgoftaar. 2009. A survey of collaborative filtering techniques. *Advances in artificial intelligence*, 2009:4.

Yee Whye Teh. 2006. A hierarchical bayesian language model based on pitman-yor processes. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 985–992. Association for Computational Linguistics.

Ashish Vaswani, Yinggong Zhao, Victoria Fossum, and David Chiang. 2013. Decoding with large-scale neural language models improves translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1387–1392, Seattle, Washington, USA, October. Association for Computational Linguistics.

Frank Wood and Yee Whye Teh. 2009. A hierarchical nonparametric Bayesian approach to statistical language model domain adaptation. In *Artificial Intelligence and Statistics*, pages 607–614.

Frank Wood, Cédric Archambeau, Jan Gasthaus, Lancelot James, and Yee Whye Teh. 2009. A stochastic memoizer for sequence data. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1129–1136. ACM.

Jun Wu and Sanjeev Khudanpur. 2000. Efficient training methods for maximum entropy language modeling. In *Interspeech*, pages 114–118.

Puyang Xu, Asela Gunawardana, and Sanjeev Khudanpur. 2011. Efficient subsampling for training complex language models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1128–1136, Stroudsburg, PA, USA. Association for Computational Linguistics.

George Zipf. 1949. Human behaviour and the principle of least-effort. Addison-Wesley, Cambridge, MA.