信息科学与工程学院毕业设计(论文)

论文翻译

题　　目　用于 Skip-gram 的
　　稀疏非负矩阵语言建模

学　　生　罗　雅　文

指导教师　吴　建　辉

专　　业　通信工程

班　　级　通信 1303

日　　期　2017 年 3 月 29 日

信息科学与工程学院

用于 Skip-gram 的稀疏非负矩阵语言建模

Noam Shazeer,[1] Joris Pelemans[1,2], Ciprian Chelba
谷歌，山景城 1600，加利福尼亚
ESAT 部，鲁汶大学，阿伦贝尔城堡 10
B-3001 鲁汶，比利时

摘要： 我们提出了一种名为稀疏非负矩阵（SNM）评估的新型语言模型（LM）评估技术系列。在十亿字基准上经验性地评估这些技术的第一组实验[3]表明，使用 Skip-gram，SNMLM 能够匹配最先进的循环神经网络（RNN）LM；结合两种建模技术，能在基准测试中产生最为人所知的结果。SNM 相对于最大熵和 RNNLM 评估的计算优势可能是其主要优点，它有望在有效组合任意特征方面具有相同的灵活性，并应该和 n-gram LM 一样能优雅地扩展应用在更巨量的数据上。

关键词： 稀疏非负矩阵，语言模型，Skip-gram

1 简介

最近，神经网络（NN）平滑化[1]，[5]，[18]，特别是循环神经网络（RNN） [12]，[20]在语言建模[3]方面表现出出色的表现。它们的出色表现归功于利用长距离语境的组合，并训练词汇的向量表示。虽然这些模型目前是最先进的技术，但它们不能很好地扩展到非常大量的数据，训练时间大约是几周。

利用长距离语境的另一种方法是使用 skip-gram [8]，[14]，[17]。Skip-gram 是常规 n-gram 的泛化，除了允许相邻的单词序列，还允许跳过单词，因此覆盖更长的上下文而不受数据稀疏性的限制。以前的工作表明，训练具有 skip-gram 特征的模型能够与基于神经网络的模型竞争媲美[19]。

为了使用 skip-gram 特征构建目标词的概率评估，我们需要一种组合任意数量的这些特征的方法，这些特征不像常规的 n-gram 特征那样落入一个简单的层次结构中。

在本文中，我们提出了一种简单而又新颖的方法，称为稀疏非负矩阵（SNM）评估，以一种计算容易的方式组合这些预测因子，优雅地扩展到大量的数据，从建模的角度看，结果也是非常有效的。我们将在十亿字基准上，通过与其他一些流行的语言模型比较最终模型的困惑度来评估该方法[3]，并展示使用 skip-gram 特征的 SNMLM 能够匹配最先进的 RNNLM；结合两种建模技术，在基准测试中产生最为人所知的结果。

在本文的剩余部分，我们介绍了 skip-gram 语言建模（第 2 节），描述了 SNMLM 范例

（第 3 节），实验评估（第 4 节），并讨论了一些相关工作（第 5 节）。第 6 节是我们的最终结论和未来工作。

2 skip-gram 语言建模

在我们的方法中，用元组来描述从上下文 W_{k-1} 中提取的 skip-gram 特征 (r,s,a)，其中：

- r 表示长距离上下文单词的数量
- s 表示跳过的单词数
- a 表示相邻上下文单词的数量

和被预测的目标词 W_k 有相关联系。例如，在句子`<S> The quick brown fox jumps over the lazy dog </S>`中，对于目标词 `dog` 来说，一个 $(1,2,3)$ skip-gram 特征是：

```
[brown skip-2 over the lazy]
```

出于性能原因，建议限制 s 并限制 $(r+s)$，或限制 r 和 s；不设任何限制，将导致含有一些 skip-gram 特征，其总表现规模将会是句子长度的五分之一。

我们配置 skip-gram 特征提取器来生成所有特征 $\mathbf{f}$，由等价类 $\Phi(W_{k-1})$ 中所定义，即满足最小和约束最大值为：

- 使用的上下文字的单词数目 $r+a$
- 长距离单词的数量 r
- 相邻单词的数量 a
- 跳过长度 s

我们还允许在特征中表示不包括 s 的确切值的选项；通过共享各种跳过特征的计数，这可能有助于进行平滑。绑定的 skip-gram 特征将如下所示：

```
[curiousity skip-* the cat]
```

为了在上下文 W_{k-1} 中为目标词 ω_k 建立良好的概率评估，我们需要一种组合任意数量的跳过特征 $\mathbf{f}_{k-1}$ 的方法，其不属于简单的层次结构，如常规的 n-gram 特征。以下部分描述了这种预测的方式，为便于计算结合了简单而新颖的方法，将其优雅地扩展应用在巨量的数据上，大量的数据和事实也证明，从建模的角度来看，这也是非常有效的。

3 稀疏非负矩阵语言模型

在本节中，我们将介绍我们的新范式而无需演算所有的推导。感兴趣的读者可以在[15]中找到这些。

3.1 模型定义

在稀疏非负矩阵（SNM）范式中，我们将训练数据表示为事件序列 $E = e_1, e_2, \ldots$，其中每个事件 $e \in E$ 由稀疏非负特征向量 $\mathbf{f}$ 和稀疏非负目标词向量 $\mathbf{t}$。两个向量都是二进制值，分别表示特征或目标词的存在或不存在。虽然 SNM 没有强制要求，但为了语言建模，事件通常具有多个特征，但只有一个单一目标词才能有效地使 t 成为一个大小为 $|\nu|$ 的一位有效编码词汇库 ν。训练数据由 $|E||Pos(\mathbf{f})||\nu|$ 训练样例组成，其中 $Pos(\mathbf{f})$ 表示向量 $\mathbf{f}$ 中的正元素集合。其中，$|E||Pos(\mathbf{f})|$ 为正（存在目标词），$|E||Pos(\mathbf{f})|(|\nu|-1)$ 为负（不存在目标字）。

语言模型由非负矩阵 $\mathbf{M}$ 表示，当应用于给定特征向量 $\mathbf{f}$ 时，产生密集预测向量 $\mathbf{y}$：

$$\mathbf{y} = \mathbf{M}\mathbf{f} \approx \mathbf{t} \tag{1}$$

在评估时，我们对 $\mathbf{y}$ 进行归一化，使得我们得到一个模型 $\mathbf{M}$ 的条件概率分布 $P_{\mathbf{M}}(\mathbf{t}|\mathbf{f})$。对于对应于 $\mathbf{t}$ 中的索引 j 的每个单词 $\omega \in \nu$，以及对应于特征向量 $\mathbf{f}$ 的历史，条件概率 $P_{\mathbf{M}}(t_j|\mathbf{f})$ 变为：

$$\begin{aligned} P_{\mathbf{M}}(t_j|\mathbf{f}) &= \frac{y_j}{\sum_{u=1}^{|\nu|} y_u} \\ &= \frac{\sum_{i \in Pos(\mathbf{f})} M_{ij}}{\sum_{i \in Pos(\mathbf{f})} \sum_{u=1}^{|\nu|} M_{iu}} \end{aligned} \tag{2}$$

为方便起见，我们将在本文的其余部分中写出 $P(t_j|\mathbf{f})$ 而不是 $P_{\mathbf{M}}(t_j|\mathbf{f})$。

根据式（2）中的分母的要求，该计算涉及对于整个词汇表的所有当前特征的求和。然而，如果我们预先计算行总和 $\sum_{u=1}^{|\nu|} M_{iu}$ 并将其与模型一起存储在一起，则可以在 $|Pos(\mathbf{f})|$ 时间内非常有效地进行评估。还要注意，由于 $\mathbf{M}$ 的稀疏性，预先计算行和仅涉及很少的项。

3.2 调整函数和元特征

我们让 M 的表达稍微修改一下相对频率：

$$M_{ij} = e^{A(i,j)} \frac{C_{ij}}{C_{i*}} \tag{3}$$

其中 $A(i,j)$ 是实值函数，称为调整函数，$\mathbf{C}$ 是在整个训练语料库上计算的特征目标计数矩

阵。

C_{ij} 表示特征 f_i 和目标 t_j 的同现频率，而 C_{i*} 表示特征 f_i 的总出现频率，即在所有目标上求和。

对于每个特征目标对 (f_i, t_j)，调整函数计算对应于 k 个新特征的权重 $\theta_k(i, j)$ 的和，称为元特征：

$$A(i, j) = \sum_k \theta_k(i, j) \tag{4}$$

从给定的输入特征（例如常规的 n-gram 和 skipgram）中，我们构造元特征作为以下基本元素中的任何一个或全部的连接：

- 特征标识，如`[the quick brown]`
- 特征类型，如 4-grams
- 特征计数，如 C_{i*}
- 目标标识，如 `fox`
- 特征-目标计数，如 C_{ij}

注意，貌似缺失的特征目标标识由特征标识和目标标识的结合来表示。由于元特征可能涉及特征计数和特征目标计数，在本文的其余部分，我们将在必要时写入 $A(i, j, C_{i*}, C_{ij})$。这将在第 3.5 节中重要，我们将在此讨论留一培训。

每个基本元特征与其他元素结合起来形成更复杂的元特征，其又与所有其他基本和复杂元特征相结合，最终以元特征的所有 2^5-1 种可能组合结束。

由于同一数量级的数量元特征具有相似的信息，因此我们将它们分组，以便它们可以共享相同的权重。我们这样做是通过根据它们的 $\log 2$ 值（向下取整）来计数元特征。

3.3 模型评估

评估模型 $\mathbf{M}$ 对应于根据一些损失函数 L，将目标矢量 $\mathbf{t}$ 与预测矢量 $\mathbf{y}$ 之间的所有事件的平均损失最小化，为所有事件找到所有事件的所有脉冲的最佳权重 θ_k。

在[15]中，我们提出了基于泊松分布的损失函数：我们认为 $\mathbf{t}$ 中的每个 t_j 都是参数 y_j 的泊松分布。$P_{Poisson}(\mathbf{t} \mid \mathbf{f})$ 的条件概率为：

$$P_{Poisson}(\mathbf{t} \mid \mathbf{f}) = \prod_{j \in \mathbf{t}} \frac{y_j^{t_j} e^{-y_j}}{t_j!} \tag{5}$$

相应的泊松损失函数为：

$$\begin{aligned} L_{Poisson}(\mathbf{y},\mathbf{t}) &= -\log(P_{Poisson}(\mathbf{t}\,|\,\mathbf{f})) \\ &= -\sum_{j\in\mathbf{t}}[t_j\log(y_j)-y_j-\log(t_j!)] \\ &= \sum_{j\in\mathbf{t}}y_j-\sum_{j\in\mathbf{t}}t_j\log(y_j) \end{aligned} \tag{6}$$

我们丢弃了最后一项，因为t_j是二进制值[1]。虽然这种选择在语言建模的上下文中并不明显，但它非常适合基于梯度的优化，我们将看到，实验结果实际上非常好。此外，泊松损失本身也适用于可能有用的多目标预测，例如在子建模中。

$$\frac{\partial(L_{Poisson}(\mathbf{Mf},\mathbf{t}))}{\partial(A(i,j))} = f_iM_{ij}(1-\frac{t_j}{y_j}) \tag{7}$$

通过对损失函数应用随机梯度下降来学习调整函数。也就是说，对于每个事件中的每个特征目标对(f_i,t_j)，我们需要通过计算相对于调整函数的梯度来更新元信息的权重。

对于完整的推导，我们参考[15]。

然后我们使用 Adagrad [4]自适应学习率程序来更新元特征权重。Adagrad 不是使用单一的固定学习率，而是在(f_i,t_j)的第N个出现时对每个权重$\theta_k(i,j)$使用单独的自适应学习率η_k，$N(i,j)$。

$$\eta_{k,N}(i,j) = \frac{\gamma}{\sqrt{\Delta_0+\sum_{n=1}^{N}\partial_n(ij)^2}} \tag{8}$$

其中γ是所有学习率的常数缩放因子，Δ_0是初始累加器常数，$\partial_n(ij)$是相对于$A(i,j)$的损失的第N个梯度的短手符号。

3.4 优化

如果我们将等式（7）中的梯度应用于每个（正和负）训练样例，则计算代价太高，因为即使所有负训练样例的第二项为零，对于所有$|E||Pos(\mathbf{f})||\nu|$训练样例，都需要计算第一项。

然而，由于第一个术语不依赖于y_j，所以我们可以通过在$f_i=1$但$t_j=0$的事件中添加梯度来分配负样例的更新。特别地，我们添加C_{i*}，而不是添加术语f_iM_{ij}，这使我们只能在正样例上更新梯度。这是基于在整个训练集上这样是等价的观察。对于完整的推导，我们参考[15]。

---

[1] 事实上，即使在t_j可以取任何非负值的一般情况下，该项也将梯度消失，因为它独立于$\mathbf{M}$。

我们注意到，本次更新只适用于单批训练是严格正确的，而不包括在线培训，因为M_{ij}在每次更新后会变化。尽管如此，我们发现这样做会产生很好的效果，并且可以大大降低计算成本。施加到每个训练样例的在线梯度变为：

$$\frac{\partial(L_{Poission}(\mathbf{Mf},\mathbf{t}))}{\partial(A(i,j))} = f_i t_j \frac{C_{i*}-C_{ij}}{C_{ij}} M_{ij} + f_i t_j (1-\frac{1}{y_j}) M_{ij} \tag{9}$$

对于正训练样例而言，它不是零，因此加速了$|\nu|$因子的计算。

3.5 留一训练

具有大量参数的模型容易过度拟合训练数据。处理此问题的首选方法是使用保留数据来评估参数。不幸的是，等式（9）中的聚合梯度不允许我们使用额外的数据来训练调整函数，因为它们将更新计算与训练数据中的相对频率$\frac{C_{i*}}{C_{ij}}$相结合。相反，我们必须求助留一训练法，以防止模型过度配套。我们通过排除生成梯度的事件来计算这些梯度的计数。因此，对于每个事件$e=(\mathbf{f},\mathbf{t})$的每个正例$(f_i,t_j)$，我们计算梯度，不包括$C_{i*}$和$C_{ij}$中的1。对于负样例的梯度，我们只从$C_{i*}$中排除1，因为我们没有观察到$t_j$。为了保持负样本的梯度的总计算，我们将其均匀分布在所有具有相同特征的正例中；然后每个C_{ij}正例将计算$\frac{C_{i*}-C_{ij}}{C_{ij}}$负例的梯度。

总而言之，当我们进行单批训练时，我们对所有正训练样例应用以下梯度更新规则：

$$\begin{aligned}
&\frac{\partial(L_{Poission}(\mathbf{Mf},\mathbf{t}))}{\partial(A(i,j))} \\
&= f_i t_j \frac{C_{i*}-C_{ij}}{C_{ij}} e^{A(i,j,C_{i*}-1,C_{ij})} \frac{C_{ij}}{C_{i*}-1} \\
&+ f_i t_j (1-\frac{1}{y'_j}) M_{ij} e^{A(i,j,C_{i*}-1,C_{ij}-1)} \frac{C_{ij}-1}{C_{i*}-1}
\end{aligned} \tag{10}$$

其中y'_j是留一法的所有相关特征的产物：

$$y'_j = (\mathbf{M}'\mathbf{f})_j$$
$$M'_{ij} = e^{A(i,j,C_{i*}-1,C_{ij}-1)} \frac{C_{ij}-1}{C_{i*}-1}$$

4 实验

我们的实验设置使用[3]提供的十亿字基准语料库[2]。

为了完整，这里是简短的语料库，仅包含单语英文资料：

- 训练标识总数约 8 亿
- 提供的词汇包括 793471 个单词，包括句子边界标记<S>，</ S>，并且通过丢 弃计数低于 3 的所有单词来构造
- 词库之外的词被映射到一个<UNK>令牌，也是词库的一部分
- 句子顺序是随机的
- 测试数据共 159658 个词（不包括语言模型从未预测的句子开始标记<S>）
- 测试集上的超出词汇（OoV）率为 0.28%。

在第一组实验中，我们研究如何将 skip-gram 特征与常规的 n-gram 特征结合起来。所有提到的 n-gram 模型都使用插值的 Kneser-Ney（KN）平滑[11]进行训练，而不计算截止值，其中减益不随模型的顺序而变化。

为了引入 skip-gram 特征，我们可以构建一个“纯粹”的 skip-gram 式 SNM，它不包含常规的 n-gram 特征（除了 unigrams），并用 KN 内插这个模型，或者我们可以构建一个具有常规 n-gram 特征和 skip-gram 特征的 SNM。我们通过选择 skip-gram 特征来比较两种方法，该特征可以被认为是 5-grams 的 skip 等效值，即它们最多包含 4 个字。特别是，我们使用 skip-gram 特征，其中长距离跨度限制在 1 到 3 之间，最多 3 个字（$r=[1..3], s=[1..3], r+a=[1..4]$）和其中所有超过 4 的 skip 都被绑定并受到最多 2 个字的长距离跨距限制（$r=[1..2], s=[1..*], r+a=[1..4]$）。然后，我们构建了一个使用这些特征和常规 5-grams（SNM5-skip）的模型，以及仅使用 skip-gram 特征（SNM5-skip（no n-gram））的模型。

Model	Params	PPL
SNM5-skip(no n-grams)	61B	69.8
SNM5-skip	62B	54.2
KN5+SNM5-skip (no n-grams)		56.5
KN5+SNM5-skip		53.6

表 1：具有和不具有 n-gram 的 SNM5-skip 模型的参数数量（百亿级）和困惑度结果，以及使用 KN5 插值的困惑度结果。

事实证明，从表 1 可以看出，最好将所有特征纳入一个单一的 SNM 模型，而不是用 KN 5-grams 模型（KN5）插值。使用 KN5 插入全留式 SNM5-skip 几乎不会产生额外的增益。这是不奇怪的，因为线性内插使用固定权重来评估每个单词序列，而 SNM 模型应用了依赖于上下文和目标单词的可变权重。迄今为止最好的 SNM 结果是使用 10-grams（SNM10-skip），连同最多 5 个字的无缝 skip-gram 特征，跳过 1 个字（$s=1, r+a=[1..5]$）以及最多 4 个字的连接 skip-gram 特征，其中只有 1 个字是长距离的，但最多可以跳过 10 个字（如

[2] http://www.statmt.org/lm-benchmark

$r=1, s=[1..10], r+a=[1..4]$）。

Model	Params	PPL	interpolation weights		
KN5	1.76B	67.6		0.06	0.00
HSME	6B	101.3		0.00	0.00
SBO	1.13B	87.9		0.20	0.04
SNM5-skip	62B	54.2			0.10
SNM10-skip	33B	52.9	0.4		0.27
RNNME-256	20B	58.2		0.00	0.00
RNNME-512	20B	54.6		0.13	0.07
RNNME-1024	20B	51.3	0.6	0.61	0.53
SNM10-skip+RNNME-1024			41.3		
Previous best				43.8	
ALL					41.0

表 2：所有调查模型的参数数量（百亿）和复杂度结果，以及插值结果和权重。

丰富的短距离和浅长距离特征的混合使得该模型能够获得最先进的结果。表 2 将其对 KN5 的困惑以及以下语言模型进行了比较：

- 分层软最大熵 LM（HSME）[7]，[13]
- 傻瓜式倒退 LM（SBO）[2]
- 具有最大熵的反复神经网络（RNNME）[12]

然而，描述这些模型超出了本文的范围。相反，我们将读者引导至[3]，其中包含表 2 中所有模型的详细描述。

当我们将 SNM10-skip 的困惑与隐藏层（RNNME-1024）中的 1024 个神经元的最先进的 RNNLM 进行比较时，发现差异仅为 3％。此外，虽然我们的模型具有比 RNN 更多的参数（33 vs 200 亿），但训练只需要大约十分之一的时间（24 小时 vs 240 小时）。有趣的是，当我们插入这两个模型时，我们增加了 20％的增益，据我们所知，41.3 的困惑已经是这个数据库中最好的报告，最高达 6％。

最后，当我们优化所有模型的插值权重时，额外模型的贡献以及困惑度的降低可以忽略不计。

5 相关工作

SNM 评估与依赖于各种顺序混合相对频率的所有 n-gram LM 平滑技术密切相关。与大多数这些不同，它将各种指令的预测变量相结合，而不依赖于上下文的分层嵌套，使其更接近于最大熵（ME）[17]或指数模型。

我们不是第一个强调在更长时间内捕获依赖关系的 skip-gram 效果的功能，类似于 RNNLM；最近，[16]也表明使用单跳的退避泛化产生了显着的困惑度下降。我们注意到，我们的 SNM 模型是使用单个和更长的跳过进行训练的，据我们所知，我们评估特征权重的方法是完全原始的。

通过分层预测输出层[7]和子采样[21]提供的 ME 和 RNN LM 训练的加速仍然需要在词

汇大小中线性化的更新与训练数据中的单词数量相关，而对于更小的调整函数，式（10）中的 SNM 更新消除了对词汇大小的依赖。

6 结论与未来的工作

我们已经提出了 SNM，一种新的 LM 评估技术系列。对十亿字基准的第一次实证评估[3]表明，使用 skip-gram 特征，SNMLM 能够匹配最先进的 RNN LM;结合两种建模技术，在基准测试中产生最为人所知的结果。

SNMLMs 与最大熵和 RNNLM 评估的计算优势有一定的方法在有效地组合任意特征方面具有相同的灵活性，但是应该像 n-gram LM 一样优雅地扩展到非常大量的数据。

未来的工作项目包括模型修剪，探索类似于[6]的更丰富的功能，以及调整模型中更丰富的元数据，混合了针对各种数据源进行培训的 SNM 模型，使其在给定的开发集上表现最好，在这方面更灵活的评估技术。

7 参考文献

[1] Yoshua Bengio, Rejean Ducharme, Pascal Vincent, and Christian Janvin. “A Neural Probabilistic Language Model,” Journal of Machine Learning Research, 3, pp. 1137–1155, 2003.

[2] Thorsten Brants, Ashok C. Popat, Peng Xu, Franz J. Och, and Jeffrey Dean. “Large Language Models in Machine Translation,” Proceedings of EMNLP, pp. 858–867, 2007.

[3] Ciprian Chelba, Toma´s Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. “One Billion Word Benchmark for Measuring Progress in Statistical Language Modeling,” Proceedings of Interspeech, pp. 2635–2639, 2014.

[4] John Duchi, Elad Hazan and Yoram Singer. “Adaptive Subgradient Methods for Online Learning and Stochastic Optimization,” Journal of Machine Learning Research, 12, pp. 2121–2159, 2011.

[5] Ahmad Emami. “A Neural Syntactic Language Model,” Ph.D. Thesis, Johns Hopkins University, 2006.

[6] Joshua T. Goodman. “A Bit of Progress in Language Modeling, Extended Version,” Technical Report MSR-TR-2001-72, 2001.

[7] Joshua T. Goodman. “Classes for Fast Maximum Entropy Training,” Proceedings of ICASSP, pp. 561–564, 2001.

[8] Xuedong Huang, Fileno Alleva, Mei-Yuh Hwang, and Ronald Rosenfeld. “An Overview of the SPHINX-II Speech Recognition System,” Computer Speech and Language, 2, pp. 137–148, 1993.

[9] Frederick Jelinek. “Information Extraction From Speech And Text,” MIT Press, 1997.

[10] Slava M. Katz. “Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer,” IEEE Transactions on Acoustics, Speech and Signal Processing, ASSP-35, 3, pp. 400–401, 1987.

[11] Reinhard Kneser and Hermann Ney. “Improved Backing-Off for M-Gram Language Modeling,” Proceedings of ICASSP, pp. 181–184, 1995.

[12] Toma´s Mikolov. “Statistical Language Models Based on Neural Networks,” Ph.D. Thesis, Brno University of Technology, 2012.
[13] Frederic Morin and Yoshua Bengio. “Hierarchical Probabilistic Neural Network Language Model,” Proceedings of AISTATS, pp. 246–252, 2005.
[14] Hermann Ney, Ute Essen, and Reinhard Kneser. “On Structuring Probabilistic Dependences in Stochastic Language Modeling,” Computer Speech and Language, 8, pp. 1–38, 1994.
[15] Noam Shazeer, Joris Pelemans and Ciprian Chelba. “Skip-gramLanguage Modeling Using Sparse Non-negative Matrix Probability Estimation,” CoRR, abs/1412.1454, 2014. [Online]. Available: http://arxiv.org/abs/1412.1454.
[16] Rene Pickhardt, Thomas Gottron, Martin Korner, Paul G. Wagner,¨ Till Speicher, and Steffen Staab. “A Generalized Language Model as the Combination of Skipped n-grams and Modified Kneser-Ney Smoothing,” Proceedings of ACL, pp. 1145–1154, 2014.
[17] Ronald Rosenfeld. “Adaptive Statistical Language Modeling: A Maximum Entropy Approach,” Ph.D. Thesis, Carnegie Mellon University, 1994.
[18] Holger Schwenk. “Continuous Space Language Models,” Computer Speech and Language, 21, pp. 492–518, 2007.
[19] Mittul Singh and Dietrich Klakow. “Comparing RNNs and Loglinear Interpolation of Improved Skip-model on Four Babel Languages: Cantonese, Pashto, Tagalog, Turkish,” Proceedings of ICASSP, pp. 8416–8420, 2013.
[20] Martin Sundermeyer, Ralf Schluter, and Hermann Ney. “LSTM¨ Neural Networks for Language Modeling,” Proceedings of Interspeech, pp. 194–197, 2012.
[21] Puyang Xu, Asela Gunawardana, and Sanjeev Khudanpur. “Efficient Subsampling for Training Complex Language Models,” Proceedings of EMNLP, pp. 1128–1136, 2011.