

# DATA NOISING AS SMOOTHING IN NEURAL NETWORK LANGUAGE MODELS

Ziang Xie, Sida I.Wang, Jiwei Li, Daniel L'évy, Aiming Nie,  
Dan Jurafsky, Andrew Y. Ng

ICLR 2017

김웅희

---

## • Index

### 목차

- Abstract
- Introduction
- Related Work
- Method
- Experiments
- Conclusion

## • Abstract



Data noising은 신경망 모델을 정규화하는 효과적인 방법

컴퓨터 비전과는 다르게 자연어 특성상 Data noising은 쉽게 사용할 수 없음

n-gram 모델에 smoothing 하는 방법을 이용해 NNLM에 noising 하는 방법 제안

---

## • Introduction

언어 모델링에서 가장 큰 문제는 데이터 희소성 문제

언어모델의 고전적인 n-gram 모델은 smoothing을 이용해 데이터 희소성 문제에 대처  
하지만 기존 정규화 방식은 input data 대신 weight와 hidden unit을 다룸

# • Introduction

## Data sparsity problem과 Smoothing이란?

데이터가 희소할 경우 일반화가 어렵다는 것

- **Data sparsity problem** : n-gram 언어 모델은 일부의 단어를 보기 때문에 zero count 문제가 생김
- **Smoothing** : zero count를 제거하기 위해 작은 상수항을 더함

	I	want	to	eat	chinese	food	lunch	spend
count	2533	927	2417	746	158	1093	341	278
I	5	827	0	9	0	0	0	2
want	2	0	608	1	6	6	5	1
to	2	0	4	686	2	0	6	211
eat	0	0	2	0	16	2	42	0
chinese	1	0	0	0	0	82	1	0
food	15	0	15	0	1	4	0	0
lunch	2	0	0	0	0	1	0	0
spend	1	0	1	0	0	0	0	0

  
**Smoothing**

	I	want	to	eat	chinese	food	lunch	spend
count	2533	927	2417	746	158	1093	341	278
V	1446	1446	1446	1446	1446	1446	1446	1446
count+V	3979	2373	3863	2192	1604	2539	1787	1724
I	0.00151	0.20809	0.00025	0.00251	0.00025	0.00025	0.00025	0.00075
want	0.00126	0.00042	0.25664	0.00084	0.00295	0.00295	0.00253	0.00084
to	0.00078	0.00026	0.00129	0.17784	0.00078	0.00026	0.00181	0.05488
eat	0.00046	0.00046	0.00137	0.00046	0.00776	0.00137	0.01962	0.00046
chinese	0.00125	0.00062	0.00062	0.00062	0.00062	0.05175	0.00125	0.00062
food	0.00630	0.00039	0.00630	0.00039	0.00079	0.00197	0.00039	0.00039
lunch	0.00168	0.00056	0.00056	0.00056	0.00056	0.00112	0.00056	0.00056
spend	0.00116	0.00058	0.00116	0.00058	0.00058	0.00058	0.00058	0.00058

---

## • Related Work

언어 모델링을 위한 Data Augmentation은 존재하지 않음  
 $L_2$ 정규화와 dropout과 같은 고전적인 정규화 작업이 있지만 data를 다루지는 않음  
무작위로 zero-masking 하는 연구가 있지만 추론만이 있음  
그 외의 연구들도 개선된 이유는 조사하지 않음

---

# • Method – Smoothing and Noising

Noising for RNN models

- RNN 모델은 count 하는 개념이 아니기 때문에 n-gram에서 사용하는 smoothing 방법이 **불가**
- 1. **Unigram noising** : 어떤 단어  $x_i$  에 대해  $\gamma$ 의 확률로 유사어로 대체  
ex) brown fox -> brown dog
- 2. **Blank noising** : 어떤 단어  $x_i$  에 대해  $\gamma$ 의 확률로 \_로 대체  
ex) brown fox -> brown \_

---

## • Method – Borrowing Techniques

### Noising Probability

Smoothing 기법을 차용하여 일반적으로 적용될 수 있는 bi-gram에 노이즈가 덜 발생하도록 확률( $\gamma$ ) 계산

**and the**

**Humpy Dumpty**

and the는 영어에서 가장 흔하게 등장하는 bi-gram  
확률  $\gamma(x_{1:t})$ 를 정의하여 해당 경우에는 noising이 덜 들어가도록 해야함

Humpy Dumpty는 sticky pair로 noising이 들어가지 않도록 함



# • Method – Borrowing Techniques

## 1. Absolute discounting

Bigram count in training set	Bigram count in heldout set
0	0.0000270
1	0.448
2	1.25
3	2.24
4	3.23
5	4.21
6	5.23
7	6.21
8	7.21
9	8.26

Training set에서 0.75씩 빼주면 heldout set에 나타나는 bi-gram count와 비슷

$w_i$ 로 시작하는 고유한 bi-gram 조합의 수

$$P_{absolutediscounting}(w_i|w_{i-1}) = \frac{C(w_{i-1}w_i) - d}{\sum_v C(w_{i-1}v)} + \lambda(w_{i-1}P_{w_i})$$

$w_i$  bi-gram의 총개수

0.75로 지정하거나 count가 1일 땐 0.5로 지정

2,200만개 단어에서 4번 나타나는 bi-gram은 2,200만개 단어에서 3.23개 나타남

# • Method – Borrowing Techniques

## 1. Absolute discounting

$$\gamma_{AD}(x_1) = \boxed{\gamma_0} \frac{N_{1+}(x_1, \bullet)}{\sum_{x_2} c(x_1, x_2)}$$

fixed rate

$N_{1+}(x_1, \bullet) \stackrel{\text{def}}{=} |\{x_2 : c(x_1, x_2) > 0\}|$

Training set에서  $x_1$ 으로 시작하는 고유한 bi-gram 조합

$x_1$ 과  $x_2$ 의 bi-gram의 count 수

Absolute discounting으로 bi-gram에 대한 확률  $0 \leq \gamma_{AD} \leq 1$ 을 구함

# • Method – Borrowing Techniques

## 2. Kneser-Ney

Absolute discounting은 Sticky pair에 대한 대처 불가

I can't see without my reading \_\_\_\_\_ 이라는 문장에서 \_\_\_\_\_에 들어갈 말은 glasses  
하지만 unigram에서 Hong Kong이라는 단어가 많이 나와 Kong이 빈도수가 높다면 Kong이 나오게 됨  
unigram에서는 context에 대한 정보를 담을 수 없어 **continuation**이라는 개념 사용

$$P_{CONTINUATION}(w) = \frac{|\{v : C(vw) > 0\}|}{|\{(u', v') : C(u'w') > 0\}|}$$

전체 bigram count

1. Kong이 두 번째 단어로 나오는 bigram count를 더함
2. Bigram의 전체 count를 더함
3. 1번에서 구한 Kong의 Continuation count를 2번에서 구한 값으로 나누면 Kong의 Continuation 확률이 됨

구하고자 하는 단어가 bigram의  
2번째 단어로 사용된 count

# • Method – Borrowing Techniques

## 2. Kneser-Ney


Absolute discounting

$$P_{\text{absolutediscounting}}(w_i | w_{i-1}) = \frac{C(w_{i-1}w_i) - d}{\sum_v C(w_{i-1}v)} + \lambda(w_{i-1})P_{w_i}$$

Kneser-Ney

$$P_{KN}(w_i | w_{i-1}) = \frac{\max(c_{KN}(w_{i-1}w_i) - d, 0)}{C(w_{i-1}) + \lambda(w_{i-1})P_{\text{CONTINUATION}}(w_i)}$$

$$\lambda(w_{i-1}) = \frac{d}{\sum_v C(w_{i-1}v)} |\{w : C(w_{i-1}w) > 0\}|$$


$$P_{KN}(w_i | w_{n+1}^{i-1}) = \frac{\max(c_{KN}(w_{n+1}^i) - d, 0)}{\sum_v c_{KN}((w_{n+1}^{i-1}v))} + \lambda(w_{n+1}^{i-1})P_{KN}(w_i | w_{n+2}^i)$$

결국 Kneser-Ney는 기본적인 Absolute discounting에 Unigram 대신 continuation 개념을 적용한 것

# • Method – Borrowing Techniques

## 2. Kneser-Ney

$$\gamma(w_{i-n+1}^{i-1}) = \frac{D_1 N_1(w_{i-n+1}^{i-1} \bullet) + D_2 N_2(w_{i-n+1}^{i-1} \bullet) + D_{3+} N_{3+}(w_{i-n+1}^{i-1} \bullet)}{\sum_{w_i} c(w_{i-n+1}^i)}$$

Kneser-Ney로 bi-gram에 대한 확률  $0 \leq \gamma \leq 1$ 을 구함

$$\begin{aligned} Y &= \frac{n_1}{n_1 + 2n_2} \\ D_1 &= 1 - 2Y \frac{n_2}{n_1} \\ D_2 &= 2 - 3Y \frac{n_3}{n_2} \\ D_{3+} &= 3 - 4Y \frac{n_4}{n_3} \end{aligned}$$

$D_n$ 은  $n$ 개의  $n$ -gram  
 $N_n$ 은  $n$ -gram의 선행 단어 수

# • Experiments

Noising scheme	Validation	Test
Medium models (512 hidden size)		
none (dropout only)	84.3	80.4
blank	82.7	78.8
unigram	83.1	80.1
bigram Kneser-Ney	<b>79.9</b>	<b>76.9</b>
Large models (1500 hidden size)		
none (dropout only)	81.6	77.5
blank	79.4	75.5
unigram	79.4	76.1
bigram Kneser-Ney	<b>76.2</b>	<b>73.4</b>
Zaremba et al. (2014)	82.2	78.4
Gal (2015) variational dropout (tied weights)	77.3	75.0
Gal (2015) (untied weights, Monte Carlo)	—	<b>73.4</b>

Table 2: Single-model perplexity on Penn Treebank with different noising schemes. We also compare to the variational method of Gal (2015), who also train LSTM models with the same hidden dimension. Note that performing Monte Carlo dropout at test time is significantly more expensive than our approach, where test time is unchanged.

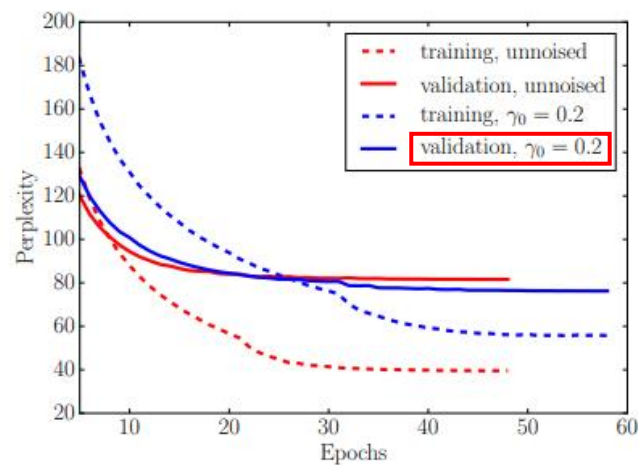
Noising scheme	Validation	Test
none	94.3	123.6
blank	85.0	110.7
unigram	85.2	111.3
bigram Kneser-Ney	84.5	110.6

Table 3: Perplexity on Text8 with different noising schemes.

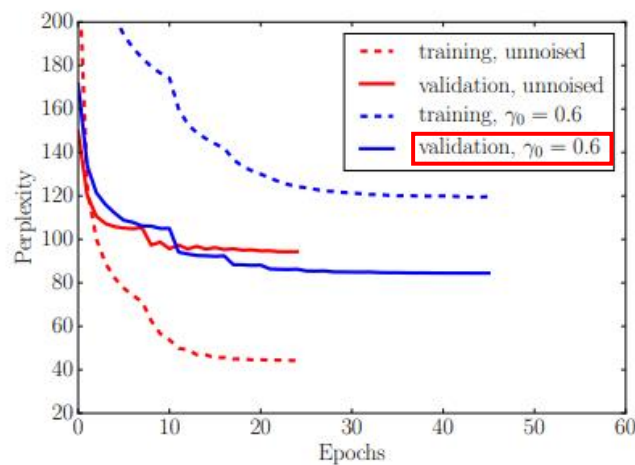
모든 지표에 대해 perplexity 성능 향상

※ perplexity : 언어모델의 내부 평가 지표로 수치가 낮을수록 성능이 좋음

# • Experiments



(a) Penn Treebank corpus.



(b) Text8 corpus.

Figure 1: Example training and validation curves for an unnoised model and model regularized using the bigram Kneser-Ney noising scheme.

Bi-gram Kneser-Ney noising으로 **validation perplexity**에 대해 좋은 성능을 보임

## • Experiments

Scheme	Perplexity	BLEU
dropout, no noising	8.84	24.6
blank noising	8.28	25.3 (+0.7)
unigram noising	8.15	25.5 (+0.9)
bigram Kneser-Ney	<b>7.92</b>	<b>26.0 (+1.4)</b>
source only	8.74	24.8 (+0.2)
target only	8.14	25.6 (+1.0)

Table 4: Perplexities and BLEU scores for machine translation task. Results for bigram KN noising on only the source sequence and only the target sequence are given as well.

언어 모델 이외에 기계 번역에서도 drop-out만 사용한 것보다 좋은 성능을 보임



---

## • Conclusion

Data noising은 신경망 기반 시퀀스 모델을 정규화하는데 효과적  
n-gram 모델의 smoothing 방법을 NNLM에 적용할 수 있음