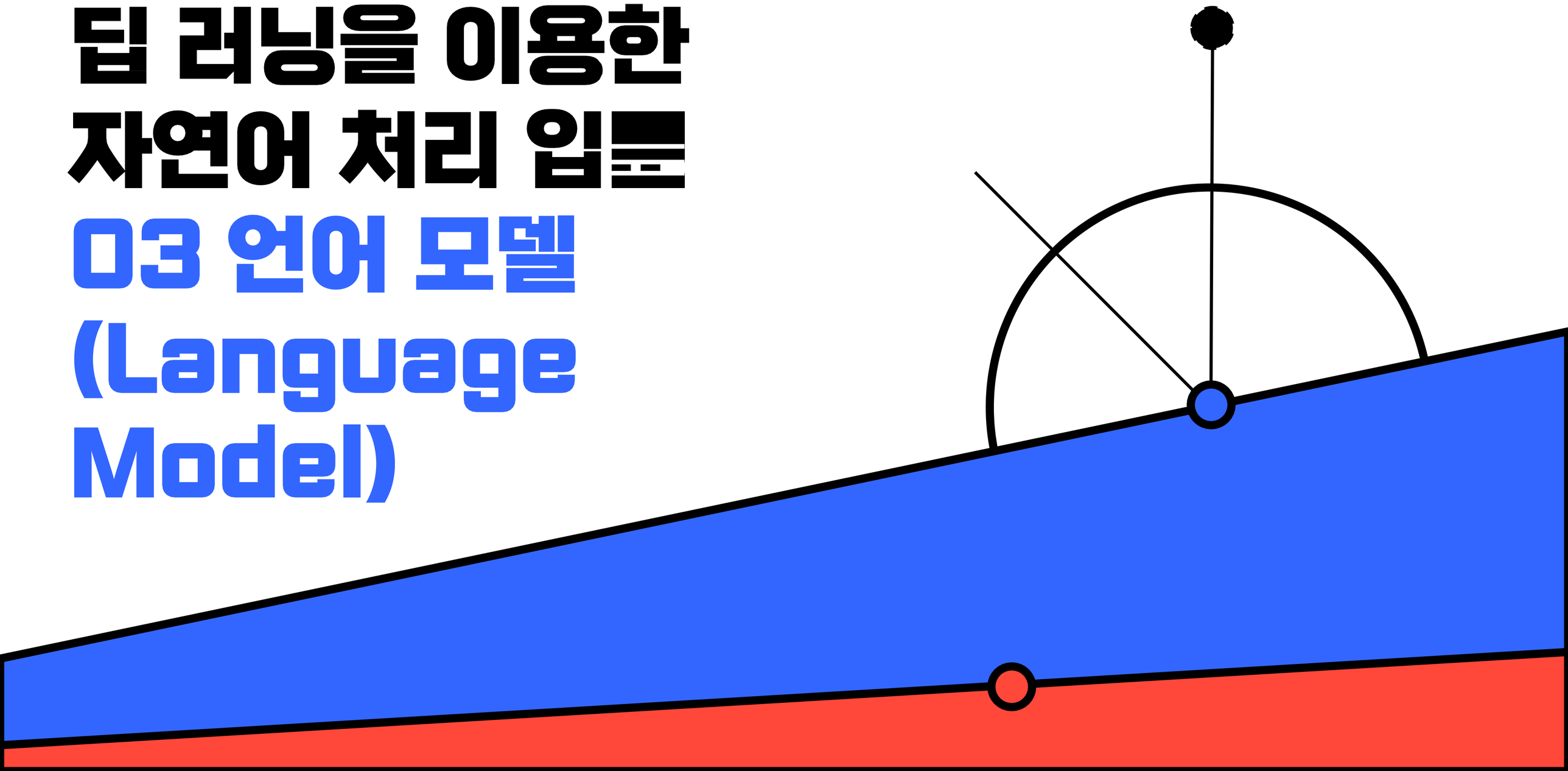
딥 러닝을 이용한 자연어 처리 입문

03 언어 모델 (Language Model)



목차

03-01 언어 모델(Language Model)이란?

1. 언어 모델(Language Model)의 정의

단어들의 시퀀스나 문장에 대해 확률을 할당해서
주어진 단어들의 순서와 문맥을 이해하여
다음에 올 단어가 무엇일지 예측 (예측하는 과정을 언어 모델링)

ex. "나는 학교에 ___"
→ 만약 모델이 "갔다"라고 예측한다면,
완성된 문장은 "나는 학교에 갔다"

통계를 이용한 방법

인공 신경망을 이용한 방법

03-01 언어 모델(Language Model)이란?

2. 단어 시퀀스의 확률 할당

(대문자 P는 확률을 의미)

a. 기계 번역(Machine Translation)

P(나는 버스를 탔다) > P(나는 버스를 태운다)

: 언어모델이 좌측의 문장의 확률이 더 높다고 판단

b. 오타 교정(Spell Correction)

선생님이 교실로 부리나케 P(달려갔다) > P(잘려갔다)

: 언어모델이 좌측의 문장의 확률이 더 높다고 판단

c. 음성 인식(Speech Recognition)

P(나는 메롱을 먹는다) < P(나는 메론을 먹는다)

: 언어모델이 우측의 문장의 확률이 더 높다고 판단

>> 언어 모델은 확률을 통해 보다 적절한 문장을 판단

03-01 언어 모델(Language Model)이란?

3. 주어진 이전 단어들로부터 다음 단어 예측

A. 단어 시퀀스의 확률

하나의 단어를 w, 단어 시퀀스를 대문자 W
n개의 단어가 등장하는 단어 시퀀스 W의 확률

$$P(W) = P(w_1, w_2, w_3, w_4, w_5, \ldots, w_n)$$

B. 다음 단어 등장 확률

n-1개의 단어가 나열된 상태에서 n번째 단어의 확률
|의 기호는 조건부 확률(conditional probability)

$$P(w_n|w_1, \ldots, w_{n-1})$$

다섯번째 단어의 확률

$$P(w_5|w_1, w_2, w_3, w_4)$$

전체 단어 시퀀스 의 확률은
모든 단어가 예측되고 나서야 알 수 있으므로
단어 시퀀스의 확률

$$P(W) = P(w_1, w_2, w_3, w_4, w_5, \ldots w_n) = \prod_{i=1}^{n} P(w_i|w_1, \ldots, w_{i-1})$$

03-01 언어 모델(Language Model)이란?

4. 언어 모델의 간단한 직관

비행기를 타려고 공항에 갔는데 지각을 하는 바람에 비행기를 [?]

>> 앞에 어떤 단어들이 나왔는지 고려하여
후보가 될 수 있는 여러 단어들에 대해서 확률을 예측
가장 높은 확률을 가진 단어를 선택

'놓쳤다'

03-01 언어 모델(Language Model)이란?

5. 검색 엔진에서의 언어 모델의 예

딥 러닝을 이용한 |

딥 러닝을 이용한 부동산가격지수 예측
딥 러닝을 이용한 자연어 처리 입문
딥 러닝을 이용한 한국어 의존 구문 분석
딥 러닝을 이용한 개체명 인식
딥 러닝을 이용한 차량 번호판 검출
딥 러닝을 이용한 한국어 의미역 결정
딥 러닝을 이용한 한국어 형태소의 원형 복원 오류 수정
딥 러닝을 이용한
딥 러닝을 이용한 구문 분석

03-02 통계적 언어 모델 (Statistical Language Model, SLM)

1. 조건부 확률의 연쇄 법칙(chain rule)

조건부 확률의 관계

$$p(B|A) = P(A,B)/P(A)$$

$$P(A,B) = P(A)P(B|A)$$

4개의 확률이 조건부 확률의 관계를 가질 때

$$P(A,B,C,D) = P(A)P(B|A)P(C|A,B)P(D|A,B,C)$$

$$P(x_1, x_2, x_3 \ldots x_n) = P(x_1)P(x_2|x_1)P(x_3|x_1, x_2) \ldots P(x_n|x_1 \ldots x_{n-1})$$

: 조건부 확률의 연쇄 법칙(chain rule)

03-02 통계적 언어 모델 (Statistical Language Model, SLM)

2. 문장에 대한 확률

'An adorable little boy is spreading smiles'

$$P(w_1, w_2, w_3, w_4, w_5, \ldots w_n) = \prod_{n=1}^{n} P(w_n | w_1, \ldots, w_{n-1})$$

$$P(\text{An adorable little boy is spreading smiles}) =$$
$$P(\text{An}) \times P(\text{adorable}|\text{An}) \times P(\text{little}|\text{An adorable}) \times P(\text{boy}|\text{An adorable little}) \times P(\text{is}|\text{An adorable little boy})$$
$$\times P(\text{spreading}|\text{An adorable little boy is}) \times P(\text{smiles}|\text{An adorable little boy is spreading})$$

>> 문장의 확률을 구하기 위해서 각 단어에 대한 예측 확률들을 곱함

03-02 통계적 언어 모델 (Statistical Language Model, SLM)

3. 카운트 기반의 접근

SLM은 이전 단어로부터 다음 단어에 대한 확률은 어떻게 구할까?
> 카운트에 기반하여 확률을 계산

An adorable little boy가 나왔을 때, is가 나올 확률

$$P(\text{is}|\text{An adorable little boy}) = \frac{\text{count}(\text{An adorable little boy is})}{\text{count}(\text{An adorable little boy })}$$

ex) An adorable little boy가 100번 등장했는데
그 다음에 is가 등장한 경우는 30번이라고 가정
> 이 경우 는 30%

03-02 통계적 언어 모델 (Statistical Language Model, SLM)

4. 카운트 기반 접근의 한계 - 희소 문제(Sparsity Problem)

자연어의 확률분포

EX. "An adorable little boy"라는 문장 뒤에
"is"라는 단어가 나올 확률은 상대적으로 높음

코퍼스(corpus)

언어 모델이 자연어의 확률 분포를 기계에게 가르치기 위한 많은 양의 데이터

하지만 카운트 기반의 방법으로 접근하려면 많은 양의 코퍼스 데이터가 필요

03-02 통계적 언어 모델 (Statistical Language Model, SLM)

4. 카운트 기반 접근의 한계 - 희소 문제(Sparsity Problem)

$$P(\text{is}|\text{An adorable little boy}) = \frac{\text{count}(\text{An adorable little boy is})}{\text{count}(\text{An adorable little boy })}$$

희소 문제(sparsity problem)
: 충분한 데이터를 관측하지 못하여 언어를 정확히 모델링하지 못하는 문제

코퍼스에 "An adorable little boy is"라는 단어 시퀀스가 없다면
>> 이 시퀀스에 대한 확률을 계산할 수 없거나 0

03-02 통계적 언어 모델 (Statistical Language Model, SLM)

4. 카운트 기반 접근의 한계 - 희소 문제(Sparsity Problem)

희소 문제를 완화하기 위해
n-gram 언어 모델이나 스무딩, 백오프와 같은 여러 **일반화 기법이 사용**

이러한 방법들은 **희소 문제를 완전히 해결하지 x**
따라서, 현대의 **언어 모델 트렌드**는 통계적 언어 모델보다는 **인공 신경망 언어 모델로 이동**

03-03 N-gram 언어 모델 (N-gram Language Model)

1. 코퍼스에서 카운트하지 못하는 경우의 감소

문장이 길어질수록

코퍼스에서 그 문장이 존재하지 않을 가능성이 높아지며, 카운트할 수 x 상황 발생

문장 더 짧게 만들면

확률을 계산하기 위해 필요한 카운트를 할 수 있는 가능성이 높아짐

$$P(\text{is}|\text{An adorable little boy}) \approx P(\text{is}|\text{boy})$$

$$P(\text{is}|\text{An adorable little boy}) \approx P(\text{is}|\text{little boy})$$

03-03 N-gram 언어 모델 (N-gram Language Model)

2. N-gram

SLM의 일종, 카운트에 기반한 통계적 접근을 사용
모든 단어를 고려 x, 일부 단어만 고려
일부 단어를 몇 개 보느냐 = n이 가지는 의미

unigrams : an, adorable, little, boy, is, spreading, smiles

bigrams : an adorable, adorable little, little boy, boy is, is spreading, spreading smiles

trigrams : an adorable little, adorable little boy, little boy is, boy is spreading, is spreading smiles

4-grams : an adorable little boy, adorable little boy is, little boy is spreading, boy is spreading smiles

~~An adorable little~~ boy is spreading ?

무시됨!

n-1개의 단어

다음에 나올 단어의 예측은 오직 n-1개의 단어에만 의존

03-03 N-gram 언어 모델 (N-gram Language Model)

2. N-gram

$$P(w|\text{boy is spreading}) = \frac{\text{count(boy is spreading } w)}{\text{count(boy is spreading)}}$$

$$P(\text{insults}|\text{boy is spreading}) = 0.500$$

$$P(\text{smiles}|\text{boy is spreading}) = 0.200$$

코퍼스에서 boy is spreading가 1,000번 등장

boy is spreading insults가 500번 등장

boy is spreading smiles가 200번 등장

> insults가 등장할 확률은 50%이며, smiles가 등장할 확률은 20%

03-03 N-gram 언어 모델 (N-gram Language Model)

3. N-gram Language Model의 한계

~~An adorable little~~ boy is spreading ?

무시됨!

n-1개의 단어

4-gram 언어 모델은 주어진 문장에서 앞에 있던 단어인 '
작고 사랑스러운(an adorable little)'이라는 수식어 반영하지 않음

작고 사랑스러운 소년이 모욕을 퍼트렸다? 웃음을 지었다?

(1) 희소 문제(Sparsity Problem)

일부 단어만을 보는 것으로
현실적으로 코퍼스에서
카운트 할 수 있는 확률을 높일 수 있지만
여전히 n-gram에 대한 희소 문제가 존재

03-03 N-gram 언어 모델 (N-gram Language Model)

3. N-gram Language Model의 한계

(2) n을 선택하는 것은 trade-off 문제

n을 크게 선택

- 실제 훈련 코퍼스에서 해당 n-gram을 카운트할 수 있는 확률 감소 > 희소 문제 심각
- 모델의 사이즈도 커지는 문제
- 모든 n-gram에 대해 카운트를 해야하기 때문에 메모리와 계산량이 증가

n을 작게 선택

- 훈련 코퍼스에서는 카운트가 잘 이루어질 수 있지만,
근사의 정확도는 현실의 확률 분포와 멀어질 수 있음

> trade-off 문제 때문에 정확도를 높이려면 n은 최대 5를 넘게 설정하지 않는 것이 권장

03-03 N-gram 언어 모델 (N-gram Language Model)

4. 적용 분야(Domain)에 맞는 코퍼스의 수집

마케팅 분야에서는 마케팅 단어가 빈번하게 등장할 것이고,
의료 분야에서는 의료 관련 단어가 당연히 빈번하게 등장

해당 도메인의 코퍼스를 사용한다면 당연히 언어 모델이 제대로 된 언어 생성

03-03 N-gram 언어 모델 (N-gram Language Model)

5. 인공 신경망을 이용한 언어 모델(Neural Network Based Language Model)

N-gram Language Model의 한계점을 극복하기위해
여러 일반화(generalization) 방법들이 존재

n-gram 언어 모델에 대한 취약점을
완전히 해결하지는 못하였고,
인공 신경망을 이용한 언어 모델이 많이 사용

03-04 한국어에서의 언어 모델 (Language Model for Korean Sentences)

1. 한국어는 어순이 중요하지 않다.

   > 다음 단어로 어떤 단어든 등장할 수 있음 > 예측 어려움

1. 나는 운동을 합니다 체육관에서.
2. 나는 체육관에서 운동을 합니다.
3. 체육관에서 운동을 합니다.
4. 나는 운동을 체육관에서 합니다.

03-04 한국어에서의 언어 모델 (Language Model for Korean Sentences)

2. 한국어는 교착어이다.

" 그녀 "

그녀가, 그녀를, 그녀의, 그녀와, 그녀로, 그녀께서, 그녀처럼

> 토큰화를 통해 접사나 조사 등을 분리해야

03-04 한국어에서의 언어 모델 (Language Model for Korean Sentences)

3. 한국어는 띄어쓰기가 제대로 지켜지지 않는다.

띄어쓰기를 제대로 하지 않아도 의미가 전달
띄어쓰기 규칙 또한 상대적으로 까다로운 언어

토큰이 제대로 분리 되지 않는채 훈련 데이터로 사용된다면
언어 모델은 제대로 동작하지 않음

03-05 펄플렉서티(Perplexity, PPL)

1. 언어 모델의 평가 방법(Evaluation metric) : PPL

'perplexed' = '헷갈리는'

$$PPL(W) = P(w_1, w_2, w_3, \ldots, w_N)^{-\frac{1}{N}} = \sqrt[N]{\frac{1}{P(w_1, w_2, w_3, \ldots, w_N)}}$$

PPL은 '낮을수록' 언어 모델의 성능이 좋다는 것을 의미
PPL은 문장의 길이로 정규화된 문장 확률의 역수
문장 W의 길이가 N

03-05 펄플렉서티(Perplexity, PPL)

1. 언어 모델의 평가 방법(Evaluation metric) : PPL

$$PPL(W) = P(w_1, w_2, w_3, \ldots, w_N)^{-\frac{1}{N}} = \sqrt[N]{\frac{1}{P(w_1, w_2, w_3, \ldots, w_N)}}$$

,

문장의 확률에 체인룰(chain rule)을 적용 $$PPL(W) = \sqrt[N]{\frac{1}{P(w_1, w_2, w_3, \ldots, w_N)}} = \sqrt[N]{\frac{1}{\prod_{i=1}^{N} P(w_i|w_1, w_2, \ldots, w_{i-1})}}$$

n-gram을 적용(bigram 언어 모델의 경우)

$$PPL(W) = \sqrt[N]{\frac{1}{\prod_{i=1}^{N} P(w_i|w_{i-1})}}$$

03-05 펄플렉서티(Perplexity, PPL)

2. 분기 계수(Branching factor) $$PPL(W) = P(w_1, w_2, w_3, \ldots, w_N)^{-\frac{1}{N}} = \left(\frac{1}{10}^{N}\right)^{-\frac{1}{N}} = \frac{1}{10}^{-1} = 10$$

PPL은 선택할 수 있는 가능한 경우의 수

Ex. PPL= 10
언어 모델은 테스트 데이터의 각 시점(단어 예측 시점)마다
평균적으로 10개의 단어를 고려하여
어떤 단어가 정답인지 판단

두 언어 모델의 PPL을 비교하는 경우,
동일한 테스트 데이터에 대해 PPL 값을 계산한 후 비교
PPL이 낮은 언어 모델이 더 좋은 성능을 가진 모델

PPL 값이 낮다 = 테스트 데이터 상에서 높은 정확도를 보인다
but 사람이 직접 평가했을 때 언어 모델이 좋다 보장하지 x

03-05 펄플렉서티(Perplexity, PPL)

3. 기존 언어 모델 Vs. 인공 신경망을 이용한 언어 모델

5-gram을 이용한 언어 모델이며 PPL이 67.6으로 측정
그 아래의 모델들은 인공 신경망을 이용한 언어 모델들
인공 신경망을 이용한 언어 모델들은
대부분 n-gram을 이용한 언어 모델보다 더 좋은 성능 평가

Model	Perplexity
Interpolated Kneser-Ney 5-gram (Chelba et al., 2013)	67.6
RNN-1024 + MaxEnt 9-gram (Chelba et al., 2013)	51.3
RNN-2048 + BlackOut sampling (Ji et al., 2015)	68.3
Sparse Non-negative Matrix factorization (Shazeer et al., 2015)	52.9
LSTM-2048 (Jozefowicz et al., 2016)	43.7
2-layer LSTM-8192 (Jozefowicz et al., 2016)	30
Ours small (LSTM-2048)	43.9
Ours large (2-layer LSTM-2048)	39.8