세미나

Evaluating Language Models: Perplexity

2024211008

이현섭

개요

언어 모델을 평가할 때 더 좋은 모델은?

“다음 단어를 더 잘 예측하는 모델”

원시 확률 : 사전 확률, 관찰 전 확률, 단순 확률

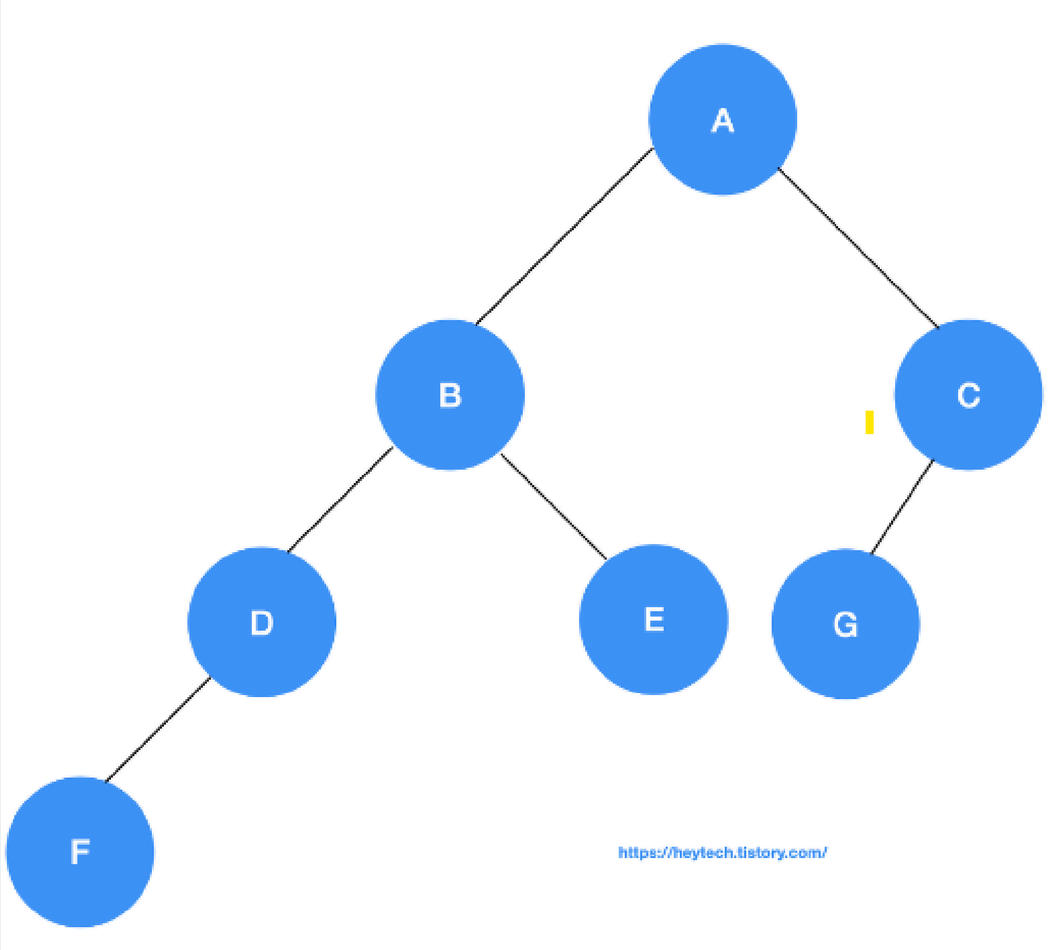
- 이들은 초기 상태의 확률, 간단한 추정의 확률
- 원시 확률을 사용하지 않는 이유?

문장 길이에 따른 편향
단어 순서의 중요성 무시
문맥 정보 반영 부족

PERPLEXITY

- 텍스트 생성(Text Generation) 언어 모델의 성능 평가지표 중 하나

- (무언가를 이해할 수 없어) 당혹스러운 정도, 헷갈리는 정도
- **perplexity 값이 낮을수록 언어 모델이 우수한 이유**



- **perplexity = 언어 모델의 분기계수(Branch Factor)**
- **자료구조에서 branch의 개수를 의미, 한 가지 경우를 골라야하는 Task에서 선택지의 개수를 뜻함.**

예시

- **체스에서 게임 털마다 말을 움직일 수 있는 경우가 평균 31~35가지**
- **바둑에서는 바둑돌을 둘 수 있는 곳이 평균 250가지**
- **모두 값이 크다. -> 경우의 수가 많다(perplexity 가 큰 복잡한 문제이다)**

언어 모델에서 PERPLEXITY의 의미

PERPLEXITY가 낮을수록 언어 모델의 성능이 우수하다.

단, 모델 평가 시 활용한 테스트 데이터셋이 충분히 신뢰도가 높을 때만!

데이터의 편향

- 훈련 데이터와의 불일치
- 도메인 불일치

데이터의 노이즈

- 오타, 오류
- 불필요한 정보

이 외, 데이터의 양과 부적절한 전처리도 PERPLEXITY 값이
모델의 실제 성능을 정확하게 반영하지 못할 수 있다.

PERPLEXITY 는 대규모 언어모델, N-GRAM모델을 평가하는데 사용!

PERPLEXITY의 계산식

- 테스트 세트의 역확률을 단어의 수로 정규화한 것

- 역수 때문에 단어 시퀀스의 확률이 높을수록 perplexity가 낮아짐.
- 전체 문장의 확률을 직관적으로 나타내지만, 각 단어 간의 관계를 명시적으로 표현하지는 못함.

$$\begin{aligned}\text{perplexity}(W) &= P(w_1 w_2 \dots w_N)^{-\frac{1}{N}} \\ &= \sqrt[N]{\frac{1}{P(w_1 w_2 \dots w_N)}}\end{aligned}$$

- Chain Rule를 적용한 perplexity 계산식
- Chain Rule : 전체 문장의 확률을 각 단어의 조건부 확률의 곱으로 나타내는 규칙

- 언어 모델이 단어 사이의 관계를 어떻게 학습하고, 예측하는지 보여줌.

$$\text{perplexity}(W) = \sqrt[N]{\prod_{i=1}^{N} \frac{1}{P(w_i | w_1 \dots w_{i-1})}}$$

PERPLEXITY 모델 설명

(UNI-GRAM)유니그램 모델 VS (BIG-RAM)빅램 모델

- Unigram : 단어 하나하나를 독립적으로 보고, 각 단어가 등장할 확률만을 고려하는 모델
- Bigram : 두 개의 연속된 단어(bigram)를 함께 고려하여 앞 단어가 주어졌을 때 다음 단어가 나올 확률을 계산

- 단어 사이의 관계를 고려하지 않기 때문에 각 단어의 독립적인 확률만 사용

$$\text{perplexity}(W) = \sqrt[N]{\prod_{i=1}^{N} \frac{1}{P(w_i)}}$$

- 앞 단어의 정보를 활용하여 다음 단어를 예측하기 때문에 조건부 확률을 사용

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

문장을 통한 짧은 예시

- Ex) 저는 오늘 점심을 먹었습니다. (단어가 4개)
- 유니그램 모델 : 각 단어가 동일한 확률, 1/4, 4가 나옴
- 빅램 모델 : 만약에 확률이 매우 높다고 가정, 유니그램보다 낮게 나올 것.

월스트리트 저널 신문 예시

3800만 단어를 사용, UNI-GRAM, BIG-RAM, TRIGAM-RAM을 적용

	Unigram	Bigram	Trigram
Perplexity	962	170	109

n-gram이 단어 시퀀스에 대해 제공하는 정보가 많을수록
n-gram이 문자열에 할당하는 확률이 높아짐

다음에 올 단어가 무엇인지 더 잘 알고 있기 때문
-> 더 높은 확률을 할당. (낮은 퍼플렉서티는 언어 모델이 테스트 세트의 더 나은 예측자)

참고사항

퍼플렉서티를 계산할 때, 언어 모델은 테스트 세트에 대한 지식 없이 구성
두 언어 모델의 퍼플렉서티는 동일한 어휘를 사용하는 경우에만 비교
이 개선방법은 음성 인식이나 기계 번역과 같은 언어 처리 작업의 성능의 개선을 보장 X

퍼플렉서티는 모델의 언어 이해 능력을 측정하는 시험 점수와 같습니다.
시험 점수가 높다고 해서 실제 사회생활에서 모든 일을 잘 할 수 있는 것은 아니듯이, 퍼플렉서티가 낮다고 해서 모든 자연어 처리 작업에서 완벽한 성능을 보장하는 것은 아닙니다.

예시

모델에 따른 perplexity의 계산 과정

L = {red, blue, green}

주목해야 하는 점 : 모델 사용

- 몇 개의 단어(branch factor) = 3

TEST SET

S = {red, red, red, red, blue}

모델 1

모든 확률 : 1/3

$$= ((\frac{1}{3})^5)^{-\frac{1}{5}}$$

$$= (\frac{1}{3})^{-1}$$

perplexity : 3

모델 2

확률 : RED = 0.8 | BLUE = 0.1 | GREEN = 0.1

$$(0.8)^4 \times (0.1)^1$$

$$= 0.04096^{-\frac{1}{5}}$$

perplexity : 1.89

예시2

모델에 따른 perplexity의 계산 과정

L = {봄, 여름, 가을 ,겨울}

주목해야 하는 점 : 모델 사용

- 몇 개의 단어(branch factor) = 4

TEST SET

S = {가을, 봄 , 겨울, 겨을, 가을}

모델 1

모든 확률 : 1/4

$$((\frac{1}{4})^5)^{-\frac{1}{5}}$$

↓

$$(\frac{1}{4})^{-1}$$

perplexity : 4

모델 2

확률 : 가을 : 0.5 |겨울 : 0.3 | 봄 : 0.1 | 여름 : 0.1

$$(0.5)^2 \times (0.3)^2 \times (0.1)^1$$

↓

$$0.00225^{(-\frac{1}{5})}$$

perplexity : 3.385

세미나

THANK YOU