# 9장 언어 모델링

2022-11-24

# 목차



2022-11-24

# 9.1 들어가며

## 9.1.1 언어 모델의 소개

- 문장의 확률을 나타내는 모델이다. 언어 모델을 통해 문장 자체의 출현 확률을 예측하거나, 이전 단어들이 주어졌을 때, 다음 단어를 예측할 수 있다. 궁극적인 목표는 일상 생활에서 사용하는 실제 언어의 분포를 정확하게 근사하는 것이다.

| 번호 | 버스 정류장에서 방금 버스를 _____. |
|---|---|
| 1 | 사랑해 |
| 2 | 고양이 |
| 3 | 놓쳤다 |
| 4 | 사고남 |

▶ 우리는 정답을 쉽게 맞출 수 있습니다.

| 번호 | 문장 |
|---|---|
| 1 | 저는 어제 점심을 먹었습니다. |
| 2 | 저는 2015년 3월 18일 점심을 먹었습니다. |

▶ 어떤 문장을 접하기 쉬울까요?

# 9.1.1 언어 모델 소개



- 언어 모델
  - 사람은 살아오면서 수많은 문장을 접했고, 머릿속에 단어와 단어 사이의 확률이 학습
  - 대화도중 일부 단어를 못 알아들어도 대화 자체에는 큰 문제가 없음
  - 음성인식 (speech recognition), 광학문자인식 (optical character recognition: OCR) 에서도 활용
- 언어 모델 학습
  - 인터넷이나 책에서 많은 문장을 수집
  - 단어와 단어 사이의 출현 빈도를 카운트하여 확률 계산
  - 일상 생활에서 사용하는 실제 언어의 분포를 근사
  - 특정 분야의 언어 모델을 만들기 위해서는 해당 분야의 코퍼스가 필요

# 9.1 들어가며

## 9.1.2 다시, 지옥불 난도의 한국어

- 한국어는 교착어이다. 교착어는 단어의 의미가 어순보다는 단어에 부착되는 어미와 같은 접사 또는 조사에 의해 결정된다. 따라서 같은 의미의 단어라도 붙는 접사나 조사에 따라서 단어의 형태가 달라지거나 단어의 수가 늘어난다.
- 어순이 중요하지 않기 때문에 단어와 단어 사이의 확률 계산에 불리 ( 이에 반해 영어나 기타 라틴어 기반 언어들은 어순이 더 규칙적이므로 유리하다. )

Ex ) '버스+가', '버스+를', '버스+에', '버스+로' / '과학+도', '과학+자', '과학+관', '과학+실'

'학교+에', '학교+로', '학교+를', '학교+가'

어미와 조사 분리를 안해주면, 희소성 문제 해결의 어려움이 생김.

| 번호 | 문장 |
|---|---|
| 1 | 나는 학교에 갑니다 버스를 타고 . |
| 2 | 나는 버스를 타고 학교에 갑니다 . |
| 3 | 버스를 타고 나는 학교에 갑니다 . |
| 4 | (나는) 버스를 타고 학교에 갑니다 . |

▶ 어순이 다르지만 모두 같은 의미의 문장

# 9.1 들어가며

## 9.1.3 문장의 확률 표현 $P(w_1, w_2)$

✓ 조건부 확률(베이즈 정리)

$$P(w_1, w_2) = P(w_1)P(w_2 \mid w_1),$$

$$\text{because } P(w_2 \mid w_1) = \frac{P(w_1, w_2)}{P(w_1)}$$

✓ 연쇄법칙(chain rule)을

$$P(w_1, w_2, \cdots, w_n) = P(w_1)P(w_2 \mid w_1)P(w_3 \mid w_1, w_2)\cdots P(w_n \mid w_1, w_2, \cdots, w_{n-1})$$

✓ 조건부 확률을 사용하여 결합 확률을 계산하는 방법으로 다음과 같이 유도할 수 있다.

$$\begin{aligned} P(A,B,C,D) &= P(D \mid A,B,C)P(A,B,C) \\ &= P(D \mid A,B,C)P(C \mid A,B)P(A,B) \\ &= P(D \mid A,B,C)P(C \mid A,B)P(B \mid A)P(A) \end{aligned}$$

# 9.1 들어가며

✓ n개의 단어가 주어졌을 때 문장의 확률 $P(w_1, w_2, \cdots, w_n) = \prod_{i=1}^{n} P(w_i \mid w_{<i})$ $\log P(w_1, w_2, \cdots, w_n) = \sum_{i=1}^{n} \log P(w_i \mid w_{<i})$

✓ i번째 문장 si= { BOS, 나는, 학교에, 갑니다, EOS } 에 대한 확률 표현 (연쇄법칙) $P(w_2 \mid w_1) = \frac{P(w_1, w_2)}{P(w_1)}$

$$P(\text{BOS}, \text{나는}, \text{학교에}, \text{갑니다}, \text{EOS})$$
$$= P(\text{BOS})P(\text{나는} \mid \text{BOS})P(\text{학교에} \mid \text{BOS}, \text{나는})P(\text{갑니다} \mid \text{BOS}, \text{나는}, \text{학교에})$$
$$P(\text{EOS} \mid \text{BOS}, \cdots, \text{갑니다})$$

$$P(\text{갑니다} \mid \text{BOS}, \text{나는}, \text{학교에}) \approx \frac{\text{Count}(\text{BOS}, \text{나는}, \text{학교에}, \text{갑니다})}{\text{Count}(\text{BOS}, \text{나는}, \text{학교에})}$$

다음의 사후 확률은 각 단어 조합의 출현 빈도를 세어 추정

# 9.2 n-gram



- n-gram
  - 확률 추정시, 전체 단어를 조합하는 대신 일부 단어 조합의 출현 빈도만을 계산

# 9.2.1 희소성 문제



- 언어 모델
  - 문장의 확률을 수식으로 표현
  - 수집한 코퍼스에서 각 단어 시퀀스의 출현 빈도를 계산
  - 출현 빈도에 기반하여 확률을 계산
- 희소성 문제
  - 애초에 출현 불가능한 단어의 조합의 경우의 수가 매우 큼
  - 단어들의 조합이 조금만 길어져도 출현 빈도를 구할 수 없음
  - 분자가 0 -> 확률이 0, 분모가 0 -> 정의 자체가 불가능

# 9.2 n-gram

## 9.2.2 마르코프 가정 (Markov assumption)

특정 시점의 상태 확률은 단지 그 직전 상태에만 의존한다는 논리

앞의 k개의 단어만 보고 다음 단어의 출현 확률을 계산 (보통 k는 0 에서 3)

$$P(x_i \mid x_1, x_2, \cdots, x_{i-1}) = P(x_i \mid x_{i-k}, \cdots, x_{i-1}) \qquad P(x_i \mid x_{i-2}, x_{i-1}) \quad \textit{(if } k = 2\textit{)}$$

✓ 연쇄법칙을 적용하여 문장에 대한 확률도 다음과 같이 표현할 수 있다.

$$P(x_1, x_2, \cdots, x_n) \approx \prod_{i=1}^{n} P(x_i \mid x_{i-k}, \cdots, x_{i-1}) \qquad \log P(x_1, x_2, \cdots, x_n) = \sum_{i=1}^{n} \log P(x_i \mid x_{i-k}, \cdots, x_{i-1})$$

# 9.2 n-gram

✓ 바로 앞의 일부 조합만 출현 빈도를 계산하여 확률을 추정 하는 방법을 n-gram이라고 한다. (n = k + 1)

✓ 훈련 코퍼스의 양이 적을수록 n의 크기도 작아져야 한다.
길고 복잡한 단어 시퀀스일수록 훈련 코퍼스에 등장하지 않을 가능성이 높기 때문이다.
3-gram 이 가장 많이 사용됨

| k | n-gram | 명칭 |
|---|---|---|
| 0 | 1-gram | uni-gram |
| 1 | 2-gram | bi-gram |
| 2 | 3-gram | tri-gram |

▶ n-gram별 명칭

3-gram: $$P(x_i \mid x_{i-2}, x_{i-1}) = \frac{\text{Count}(x_{i-2}, x_{i-1}, x_i)}{\text{Count}(x_{i-2}, x_{i-1})}$$

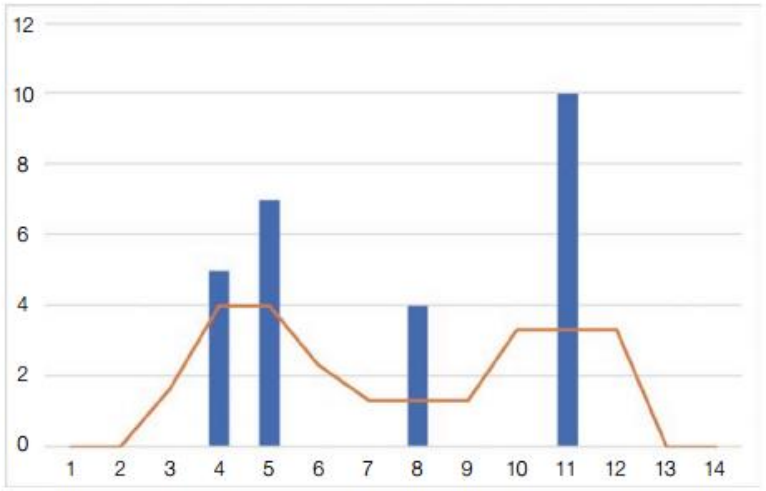
# 9.2 n-gram

### 9.2.3 일반화

- 훈련 데이터에서 보지 못한 샘플의 예측 능력 (일반화 능력) 이 중요

**| 스무딩과 디스카운팅 |** 모든 단어 시퀀스의 출현 빈도에 1을 더하는 것.

$$P(w_i \mid w_{<i}) \approx \frac{\text{Count}(w_{<i}, w_i)+1}{\text{Count}(w_{<i})+V}$$




▶ 스무딩을 통해 분포의 모양을 좀 더 평탄하게 만들 수 있습니다.

$$P(w_i \mid w_{<i}) \approx \frac{\text{Count}(w_{<i}, w_i)+k}{\text{Count}(w_{<i})+kV}$$

$$\approx \frac{\text{Count}(w_{<i}, w_i)+(m/V)}{\text{Count}(w_{<i})+m}$$

$$P(w_i \mid w_{<i}) \approx \frac{\text{Count}(w_{<i}, w_i)+mP(w_i)}{\text{Count}(w_{<i})+m}$$

# 9.2 n-gram

## 9.2.3 일반화

| Kneser-Ney 디스카운팅 | 단어 w가 다른 단어 v의 뒤에 출현 할 때, 얼마나 다양한 단어 뒤에서 출현하는지(즉, v가 얼마나 다양한지)를 알아내는 것이다.

다양한 단어 뒤에 나타나는 단어일수록 훈련 코퍼스에서 보지 못한 단어 시퀀스로 나타날 가능성이 높음

- 'learning' 앞에 출현한 단어 : machine, deep, supervised, unsupervised, generative, …
- 'laptop' 앞에 출현한 단어 : slim, favorite, fancy, expensive, cheap, …

$$Score_{\text{continuation}}(w) \propto |\{v : \text{Count}(v,w) > 0\}|$$

( w와 함께 나타난 것이라고 가정) v들의 집합 {v : Count(v, w)>0}의 크기가 클수록 Score $_{\text{Continuation}}$은 클 것이라고 가정

$$Score_{\text{continuation}}(w) = \frac{|\{v : \text{Count}(v,w) > 0\}|}{\sum_{w'} |\{v : \text{Count}(v,w') > 0\}|}$$

w와 함께 나타난 v들의 집합 {v : Count(v, w)>0} 의 크기를, 전체 단어 집합으로 부터 샘플링한 w′ ∈ W 일때,
V, w′가 함께 나타난 집합 {v : Count (v, w′)>0}의 크기의 합으로 나눈다.

# 9.2 n-gram

$$P(w_i \mid w_{<i}) \approx \frac{\text{Count}(w_{<i}, w_i) + mP(w_i)}{\text{Count}(w_{<i}) + m}$$

$$Score_{\text{continuation}}(w) = \frac{|\{v : \text{Count}(v, w) > 0\}|}{\sum_{w'} |\{v : \text{Count}(v, w') > 0\}|}$$

## 9.2.3 일반화

| Kneser-Ney 디스카운팅 |

$$P_{\text{KN}}(w_i \mid w_{i-1}) = \frac{\max(\text{Count}(w_{i-1}, w_i) - d, 0)}{\text{Count}(w_{i-1})} + \lambda(w_{i-1}) \times Score_{\text{continuation}}(w_i),$$

$d = 0.75$

$$\text{where } \lambda(w_{i-1}) = \frac{d}{\sum_v \text{Count}(w_{i-1})} \times |\{w : c(w_{i-1}, v) > 0\}|$$

$|\{w_{i-1}: C(w_{i-1}, w_i) > 0\}|$ ??

# 9.2 n-gram

## 9.2.3 일반화

| 인터폴레이션 | 다수의 언어 모델 사이의 선형결합으로, 언어 모델을 일반화 하는 방법, 두 개의 다른 언어 모델을 선형적으로 일정 비율(λ)로 섞어주는 것이다.

$$\tilde{P}(w_n \mid w_{n-k}, \cdots, w_{n-1}) = \lambda P_1(w_n \mid w_{n-k}, \cdots, w_{n-1}) + (1-\lambda) P_2(w_n \mid w_{n-k}, \cdots, w_{n-1})$$
$$\text{where } 0 < \lambda < 1$$

✓ 특정 영역에 특화된 모델 구축에 유용함.

- **일반 영역**
  - P(진정제준비.된)=0.00001
  - P(사나이준비.된)=0.01
- **특화 영역**
  - P(진정제준비.된)=0.09
  - P(약준비.된)=0.04
- **인터폴레이션 결과**
  - P(진정제준비.된)=0.5*0.09+(1-0.5)*0.00001=0.045005

# 9.2 n-gram

### 9.2.3 일반화

| 백오프 | 특정 n-gram의 확률을 n보다 더 작은 시퀀스에 대해 확률을 구하여 인터폴레이션한다.

$$\begin{aligned}\tilde{P}(w_n \mid w_{n-k}, \cdots, w_{n-1}) = \lambda_1 P(w_n \mid w_{n-k}, \cdots, w_{n-1}) \\ + \lambda_2 P(w_n \mid w_{n-k+1}, \cdots, w_{n-1}) \\ + \cdots \\ + \lambda_k P(w_n), \\ \text{where } \sum_i \lambda_i = 1\end{aligned}$$

### 9.2.4 결론

n-gram 방식은 출현 빈도를 통해 확률을 근사하므로 매우 쉽고 간편하지만, 대신 훈련 코퍼스에 등장하지 않은 단어 시퀀스는 확률을 정확하게 알 수 없는 단점도 명확하다. 따라서 마르코프 가정을 통해 단어 조합에 필요한 조건을 간소화하고, 나아가 스무딩과 백오프 방식을 통해서 남은 단점을 보완하고자 했다.

# 9.3 언어 모델의 평가 방법



- 언어 모델의 평가
  - 좋은 언어 모델 : 실제 우리가 쓰는 언어와 최대한 비슷하게 확률 분포를 근사
    - 많이 쓰이는 문장이나 표현은 높은 확률로 예측
    - 적게 쓰이는 문장이나 표현은 낮은 확률로 예측
    - 이미 사용되고 있는 문장에 대해 높은 확률값을 예측
    - 이미 사용되고 있는 문장의 앞부분이 주어졌을 때 그 다음 단어의 확률이 높을 때

# 9.3.1 퍼플렉서티



- 퍼블렉서티 (perplexity, PPL)
  - 언어 모델의 성능을 정량적으로 평가
  - 문장의 길이를 반영하여 확률값을 정규화
  - 언어 모델에서 테스트 문장들의 점수를 구하고, 언어 모델의 성능을 측정

$$\mathrm{PPL}(w_1, w_2, \cdots, w_n) = P(w_1, w_2, \cdots, w_n)^{-\frac{1}{n}} = \sqrt[n]{\frac{1}{P(w_1, w_2, \cdots, w_n)}} = \sqrt[n]{\frac{1}{\prod_{i=1}^{n} P(w_i \mid w_1, \cdots, w_{i-1})}} \approx \sqrt[n]{\frac{1}{\prod_{i=1}^{n} P(w_i \mid w_{i-n+1}, \cdots, w_{i-1})}}$$

  - 문장 길이 n으로 제곱근
    - 문장이 길어질수록 확률이 매우 작아짐
  - 문장의 확률값이 높을수록 PPL이 작아짐

# 9.5 NNLM

n-gram 기반 언어 모델의 약점을 보완하는 신경망 언어 모델인 NNLM (neural network language model)

## 9.5.1 희소성 해결하기

✓ n-gram 기반의 언어 모델은 단어 간의 유사도를 알지 못하기 때문에 훈련 데이터에서 보지 못한 단어의 조합에는 취약하다.

- 고양이는 좋은 반려동물입니다.

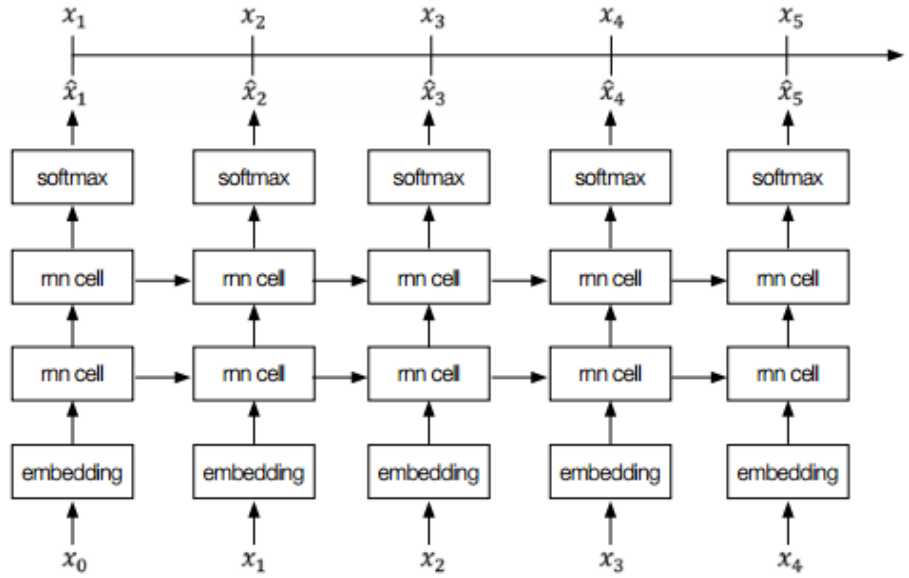
- $P$(반려동물|강아지는,좋은)
- $P$(반려동물|자동차는,좋은)

✓ 신경망 언어 모델, 즉 NNLM(neural network language model)은 단어 임베딩을 사용하여 단어를 차원 축소함 훈련 코퍼스에서 보지 못했던 단어의 조합을 만나더라도, 비슷한 훈련 데이터로부터 배운 것과 유사하게 대처할 수 있다. 희소성 해소를 통해 더 좋은 일반화 성능을 얻어낼 수 있습니다.

# 9.5 NNLM

## 9.5.2 RNNLM

RNN을 사용한 언어 모델인 RNNLM (RNN language model) 은 다음과 같은 구조를 지닙니다.




▶ RNNLM의 구조

$$P(w_1, w_2, \cdots, w_k) = \prod_{i=1}^{k} P(w_i \mid w_{<i})$$

$$\log P(w_1, w_2, \cdots, w_k) = \sum_{i=1}^{k} \log P(w_i \mid w_{<i})$$

문장의 첫 단어부터 해당 단어 직전의 단어까지
모두 조건부에 넣어 확률을 근사함. (로그 합으로 표현)

# 9.5.3 구현



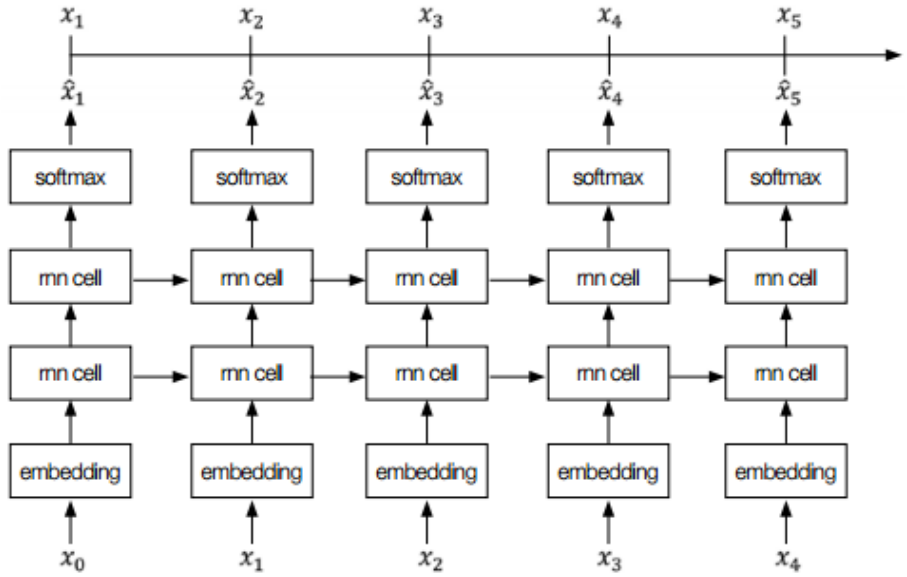
- 모델의 수식화

$$x_{1n} = \{x_0, x_1, \cdots, x_n, x_{n+1}\}$$
$$\text{where } x_0 = \text{BOS and } x_{n+1} = \text{EOS}$$

$$\hat{x}_{i+1} = \text{softmax}(\text{linear}_{\text{hidden\_size}\to|V|}(\text{RNN}(\text{emb}(x_i))))$$
$$\hat{x}_{1n}[1:] = \text{softmax}(\text{linear}_{\text{hidden\_size}\to|V|}(\text{RNN}(\text{emb}(x_{1n}[:-1])))),$$

$$\text{linear}_{d_1\to d_2}(x) = Wx + b \text{ where } W \in \mathbb{R}^{d_1\times d_2} \text{ and } b \in \mathbb{R}^{d_2},$$

and hidden_size is dimension of hidden state and |V| is size of vocabulary.




▶ RNNLM의 구조

# 9.5.3 구현

- 임베딩 과정

$$x_{1:n}[:-1] = \{x_0, x_1, \cdots, x_n\}$$
$$x_{emb} = \mathrm{emb}(x_{1:n}[:-1])$$

$$\text{where } |x_{1:n}[:-1]| = (\text{batch\_size}, \text{n+1})$$
$$\text{and } |x_{emb}| = (\text{batch\_size}, \text{n+1}, \text{word\_vec\_dim})$$

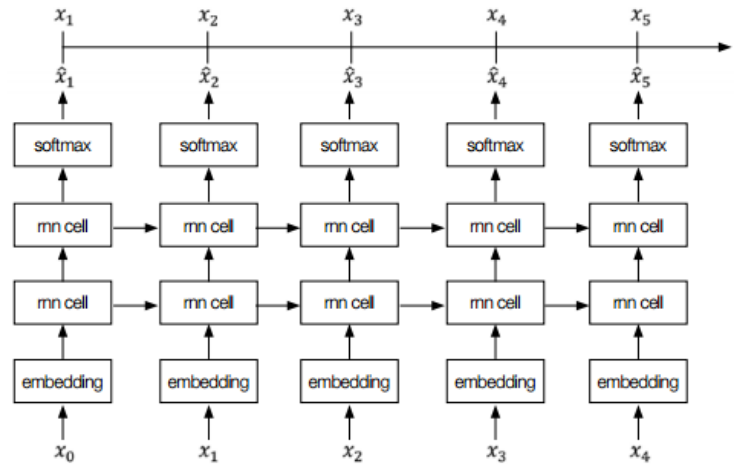
- RNN 과정

$$h_{0:n} = \mathrm{RNN}(x_{emb})$$
$$\text{where } |h_{0:n}| = (\text{batch\_size}, n+1, \text{hidden\_size})$$

- 출력 과정

$$\hat{x}_{1:n} = \mathrm{softmax}\left(\mathrm{linear}_{\text{hidden\_size}\to|V|}(h_{0:n})\right)$$

$$\text{where } |\hat{x}_{1:n}| = (\text{batch\_size}, n+1, |V|)$$
$$\text{and } x_{1:n}[1:] = \{x_1, x_2, \cdots, x_{n+1}\}$$



▸ RNNLM의 구조

손실 함수

$$\mathcal{L}(\hat{x}_{1:n}, x_{1:n}[1:]) = -\frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{n+1} x_j^i \log \hat{x}_j^i$$

where $x_j^i$ is one-hot vector

# 9.5 NNLM

### 9.5.5 결론

NNLM은 단어 임베딩 벡터를 사용하여 훈련 데이터셋에서 보지 못한 단어의 조합에 대처 가능해졌다. 하지만 테이블 검색 수준의 연산량이 있으면 되는 n-gram 방식에 비해 NNLM은 다수의 행렬 연산 등이 포함된 피드포워드 연산을 수행해야 하기 때문에 그만큼 연산량 도 많이 증가함.

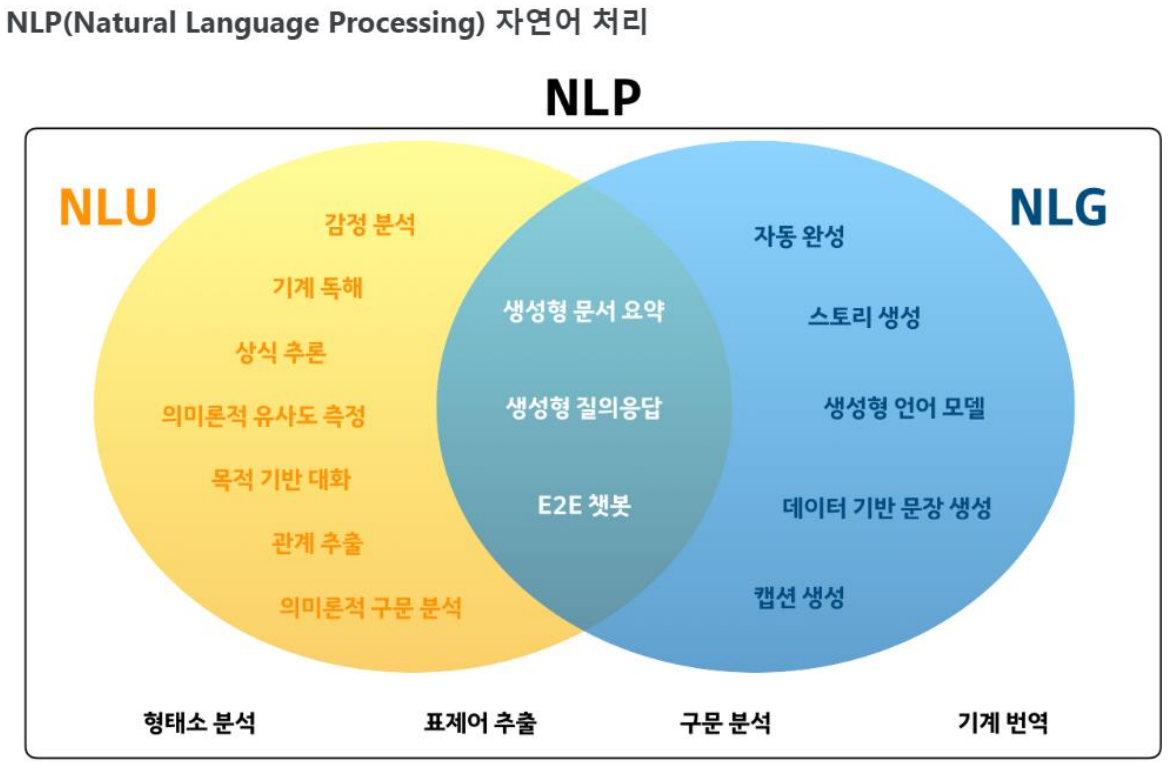
GPU 및 향상된 하드웨어 환경에서 NNLM의 중요성은 매우 커지고 있음

현재는 RNN 이 아닌 Transformer 아키텍쳐에 기반한 모델이 다수를 차지하고 있음

# 9.6 언어 모델의 활용



- 자연어 생성
  - 기계번역에서 챗봇까지
  - 언어 모델이 자연어 생성의 기본 초석



# 9.6 언어 모델의 활용

**9.6.1 음성 인식** 실제 사람의 경우에도 말을 들을 때 언어 모델이 매우 중요하게 작용한다.

✓ 음성 신호 X가 주어졌을 때 확률을 최대로 하는 문장 Ŷ를 구하는 것이 목표

$$\hat{Y} = \underset{Y \in \mathcal{Y}}{\operatorname{argmax}} P(Y \mid X) = \underset{Y \in \mathcal{Y}}{\operatorname{argmax}} \frac{P(X \mid Y)P(Y)}{P(X)},$$

where X is an audio signal and Y is a word sequence, $Y = \{y_1, y_2, \cdots, y_n\}$

그럼 베이즈 정리에 의해서 수식이 전개됩니다. 그리고 밑변 $P(X)$는 날려버릴 수 있습니다.

$$\underset{Y \in \mathcal{Y}}{\operatorname{argmax}} \frac{P(X \mid Y)P(Y)}{P(X)} = \underset{Y \in \mathcal{Y}}{\operatorname{argmax}} P(X \mid Y)P(Y)$$

✓ 여기서 P(X|Y)가 음향 모델을 의미하고 P(Y)는 언어 모델을 의미한다. 즉, P(Y)는 문장의 확률을 의미하며, P(X|Y)는 문장(단어 시퀀스 또는 음소 시퀀스)이 주어졌을 때 해당 음향 시그널이 나타날 확률을 나타낸다.

# 9.6 언어 모델의 활용

## 9.6.2 기계번역

번역 시스템을 구성할 때도 언어 모델은 중요한 역할을 한다. 기존의 통계 기반 기계번역(SMT) 시스템에서는 음성 인식 시스템과 유사하게 언어 모델이 번역모델과 결합하여 자연스러운 문장을 만들어내도록 동작했습니다.

$$\hat{Y} = \underset{Y \in \mathcal{Y}}{\operatorname{argmax}} P(Y \mid X) = \underset{Y \in \mathcal{Y}}{\operatorname{argmax}} \frac{P(X \mid Y)P(Y)}{P(X)},$$

where X is an audio signal and Y is a word sequence, $Y = \{y_1, y_2, \cdots, y_n\}$

그럼 베이즈 정리에 의해서 수식이 전개됩니다. 그리고 밑변 $P(X)$는 날려버릴 수 있습니다.

$$\underset{Y \in \mathcal{Y}}{\operatorname{argmax}} \frac{P(X \mid Y)P(Y)}{P(X)} = \underset{Y \in \mathcal{Y}}{\operatorname{argmax}} P(X \mid Y)P(Y)$$

# 9.6 언어 모델의 활용



### 9.6.3 광학 문자 인식

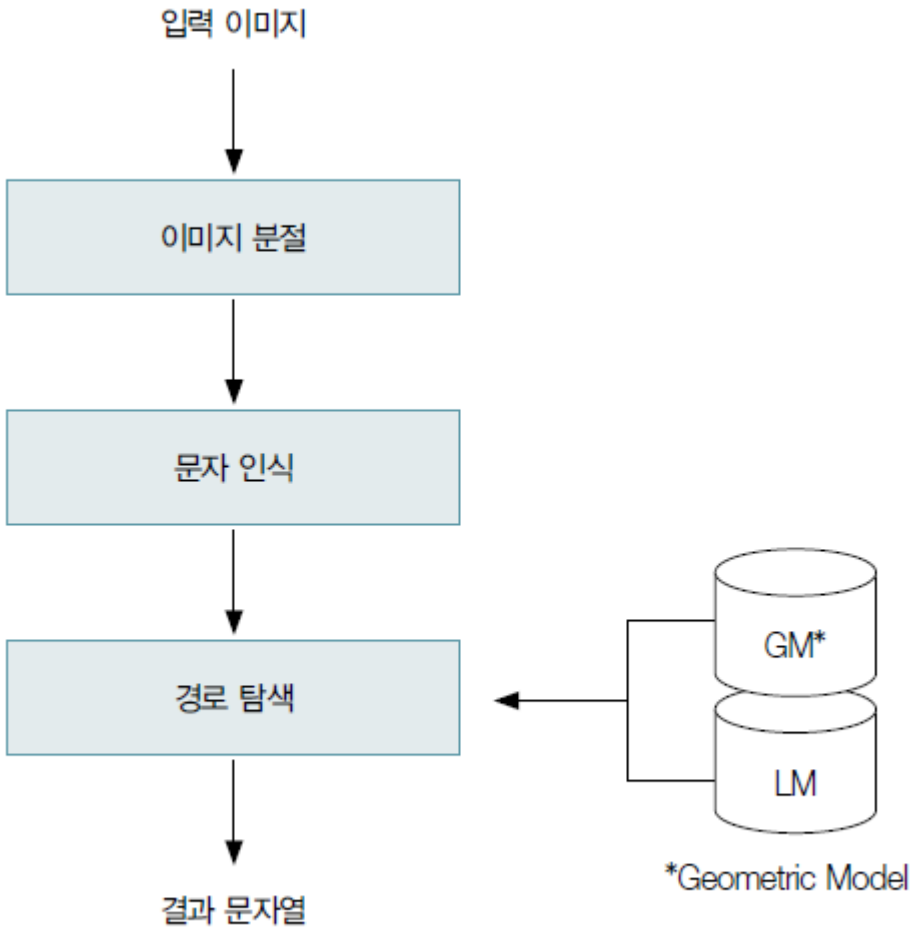
광학 문자 인식(OCR) 시스템을 만들 때도 언어 모델이 사용된다. 사진에서 추출하여 글자를 인식할 때 각 글자(character) 간의 확률을 정의하면 훨씬 더 높은 성능을 얻어낼 수 있다.

$$\hat{Y} = \underset{Y \in \mathcal{Y}}{\operatorname{argmax}} P(Y \mid X) = \underset{Y \in \mathcal{Y}}{\operatorname{argmax}} \frac{P(X \mid Y)P(Y)}{P(X)},$$

where X is an audio signal and Y is a word sequence, $Y = \{y_1, y_2, \cdots, y_n\}$

그럼 베이즈 정리에 의해서 수식이 전개됩니다. 그리고 밑변 $P(X)$는 날려버릴 수 있습니다.

$$\underset{Y \in \mathcal{Y}}{\operatorname{argmax}} \frac{P(X \mid Y)P(Y)}{P(X)} = \underset{Y \in \mathcal{Y}}{\operatorname{argmax}} P(X \mid Y)P(Y)$$




▶ 광학문자 인식 시스템의 구성 예

# 9.6 언어 모델의 활용



### 9.6.4 기타 자연어 생성 문제

앞서 나온 음성 인식이나 기계번역, 문자 인식 역시 주어진 정보를 바탕으로 문장을 생성해내는 자연어 생성에 속한다고 볼 수 있다. 주어진 정보를 바탕으로 뉴스 기사를 작성하거나 주어진 뉴스 기사를 요약하여 제목을 생성하고, 사용자의 질문에 따라 대답을 생성하는 챗봇도 생성 문제의 범주에 속한다.

### 9.6.5 기타

검색엔진에서 검색어 입력 도중 드롭다운으로 제시되는 검색어 완성 기능

$$\hat{Y} = \underset{Y \in \mathcal{Y}}{\operatorname{argmax}} P(Y \mid X) = \underset{Y \in \mathcal{Y}}{\operatorname{argmax}} \frac{P(X \mid Y)P(Y)}{P(X)},$$

where X is an audio signal and Y is a word sequence, $Y = \{y_1, y_2, \cdots, y_n\}$

그럼 베이즈 정리에 의해서 수식이 전개됩니다. 그리고 밑변 $P(X)$는 날려버릴 수 있습니다.

$$\underset{Y \in \mathcal{Y}}{\operatorname{argmax}} \frac{P(X \mid Y)P(Y)}{P(X)} = \underset{Y \in \mathcal{Y}}{\operatorname{argmax}} P(X \mid Y)P(Y)$$

# 9.7 마치며



- 언어 모델
  - 주어진 문장을 확률적으로 모델링
  - 문장 예측 능력의 필요성
  - N-gram
    - 단어를 불연속적이 존재로 취급
    - 희소성 문제
      - 마르코프 가정, 스무딩, 디스카운팅
- 신경망 기반 언어 모델
  - 일반화 문제 해결
  - 신경망은 희소한 단어들의 조합에 대해서도 효과적으로 차원을 축소
  - 유사도 비교 등을 수행
  - 처음 보는 시퀀스도 기존 지식에 기반하여 예측