Chapter 9

언어 모델

김지환

서강대학교 컴퓨터공학과

Table of contents

9.1 n-gram 언어 모델

9.2 카테고리 기반 언어 모델

9.3 DNN 기반 언어 모델

9. 언어 모델 - 개요

- 언어 모델
  - 특정 단어열이 주어졌을 때 다음에 나올 단어들의 확률을 추정하는 모델
    - 예) 내일 오후 3시에 (□) 가자
      - (□): 학교, 강남, 서점, 공원, …
  - T개의 단어로 구성된 문장 W $(w_0 \ \ldots \ w_T)$ 에 대해서 문장 생성 확률 $P(W) = P(w_0 \ \ldots \ w_T)$을 계산

9. 언어 모델 - 개요

- 단어를 구분하는 단위
  - **형태소(morpheme)**: 의미를 가지는 언어 단위 중 가장 작은 언어단위이다. 그러므로 형태소는 더 쪼개면 전혀 의미가 없어지거나 또는 이전의 의미와 관련되는 의미가 없어지는 문법 단위이다.
    - 예) 내일 오후 세시에 학교 가자
      - 내일/오후/세/시/에/학교/가/자
  - **어절**: 어절은 띄어쓰기로 나누어지는 언어 단위이다.
    - 예제:
      - 내일/오후/세시에/학교/가자
      - 어절은/한국어에서/문장을/…/단위이다.
      - 길동이가/공부를/한다.
    - 영어의 경우에는 단어의 단위가 어절임
      - Let’s/go/to/school/tomorrow/at/three/pm

http://100.daum.net/encyclopedia/view/b25h1137a
https://ko.wikipedia.org/wiki/%EC%96%B4%EC%A0%88

9. 언어 모델 - 개요

- **음절(syllable):** 화자와 청자가 한 뭉치로 생각하는 발화의 단위. 음소보다 크고 낱말보다 작다. 음절은 자음과 모음 또는 단독 모음으로 구성된다.
  - 내/일/오/후/세/시/에/학/교/가/자
  - 동북아시아 언어(한중일)는 음절단위로 단어를 나눌 수 있는 언어이다.
    - 你/好/吗
    - は/じ/め/ま/し/て

9. 언어 모델 - 개요

- 한국어 문장 예제: (형태소 단위로 시퀀스를 구성)
  - 올 여름 평년보다 덥고 강수량 지역 차 크다
    - <s>/올/여름/평년/보다/덥/고/강수량/지역/차/크/다/</s>
    - w0/w1/w2/w3/w4/w5/w6/w7/w8/w9/w10/w11/</s>

- 영어 문장 예제: (단어 단위로 시퀀스를 구성)
  - It is hotter than usual this summer, and the regional difference of precipitation is big
    - <s>/It/is/hotter/than/usual/this/summer,/and/the/regional/difference/of/precipitation/is/big/</s>
    - w0/w1/w2/w3/w4/w5/w6/w7/w8/w9/w10/w11/w12/w13/w14/w15/w16

9. 언어 모델 - 개요

- 언어 모델 (Cont.)
  - 단어 별로 decomposition을 한 후, history(w_{k-1} w_{k-2} ... w_0)로 부터 다음 단어(w_k)를 예측함
  - T 개의 단어로 구성된 문장 W (w_0 ... w_T) 에 대해서 문장 생성 확률은 아래와 같이 계산됨

$$P(W) = \prod_{k=1}^{T} P(w_k | w_{k-1} w_{k-2} \dots w_0)$$

  - 단어 별 생성확률은 음성인식 decoding network의 각 단어의 end state에서 적용됨

- <s>/내일/오후/세시에/학교/가자/</s>
  - P(<s>,내일,오후,세시에,학교,가자,</s>) = P(<s>)*P(내일|<s>)*P(오후| 내일 ,<s>)*P(세시에|오후,내일,<s>)*P(학교|세시에,오후,내일,<s>) *P(가자|학교,세시에,오후,내일,<s>)*P(</s>|가자,학교,세시에,오후,내일,<s>)

9. 언어 모델 - 개요

- 각 단어들은 컴퓨터 내부에서 어휘에 대한 index로 표현
  - 어휘(Vocabulary): 인식 가능한 단어들의 집합
  - 예: 어휘의 크기가 10만이고, 단어들은 어절단위로 구분 되었으며, 어휘 내 단어들을 가나다순으로 정렬했을 때, 단어들이 아래의 순서의 단어로 어휘 내 위치해 있다고 가정한다.
    - '가자': 12,844번째 단어
    - '내일': 24,882번째 단어
    - '세시에': 35,493번째 단어
    - '오후': 69,864번째 단어
    - '학교': 95,867번째 단어
  - '내일/오후/세시에/학교/가자'는 아래와 같이 index의 열로 표현된다.
    - 24,882/69,864/35,493/95,867/12,844

9. 언어 모델 - 개요

- Out-of-vocabulary(OOV) 단어는 최소화 되도록 해야 함
  - 자연언어에서 나타나는 단어의 수는 무제한임
  - 따라서, 문장내의 모든 단어들이 어휘내의 단어는 아님
- Lexical Coverage
  - 영어 비즈니스 신문 텍스트의 경우, 5k 단어가 일반적으로 9%의 OOV rate을 가짐. (20k는 2%, 65k는 0.6%)
- n-gram 언어모델을 사용하는 인식기에서, 일반적으로 하나의 OOV 단어가 1.6단어의 음성 인식 오류를 생성함
- 특정 개인 또는 주제에 알맞게 어휘를 구성하면, 한정된 어휘 크기로 넓은 lexical coverage 를 얻을 수 있음
  - 빈도수가 높은 단어 순으로 어휘를 구성

9.1 n-gram 언어 모델

계산 방법

- 문장이 길어짐에 따라 history 또한 길어짐
  - 확률 계산이 computationally intractable하게 됨
  - 계산이 가능하게 하기 위해서는 history의 길이를 줄여야 함
- 표준화된 방법은 n-gram임
  - 가정사항: 최근 n-1개의 단어로 구성할 수 있는 모든 history는 같은 history로 다룸

$$P(w_k|w_{k-1}w_{k-2}\ldots w_0) = P(w_k|w_{k-1}\ldots w_{k-N+1})$$

- n-gram 모델
  - Unigram : 현재 한 단어만 반영
  - Bigram : 바로 앞 단어까지 반영
  - Trigram : 바로 앞 두 단어까지 반영
    - 예) (오후, 3시에, 학교), (오후, 3시에, 강남), (오후, 3시에, 서점), …
  - 대용량의 학습 코퍼스로부터 통계적 자료 추출하여 생성

9.1 n-gram 언어 모델

- Estimating n-Grams From Counts
  - Expectation Maximization에 따라 $P(w_i|w_{i-2}, w_{i-1})$는 아래와 같이 계산 된다.

$$P(w_i|w_{i-2}, w_{i-1}) = \frac{f(w_{i-2},\ w_{i-1},\ w_i)}{\Sigma_i f(w_{i-2}, w_{i-1}, w_i)} = \frac{f(w_{i-2},\ w_{i-1},\ w_i)}{f(w_{i-2}, w_{i-1})}$$

  - N-gram 언어모델의 장점
    - 통계적 모델로써 계산의 간편함
    - 대용량 학습 자료를 이용하여 쉽게 모델 생성이 가능함
  - N-gram 언어모델의 단점
    - N의 제약으로 인하여 longer history에 대한 정보를 표현하지 못함

9.1 n-gram 언어 모델

- 예) 말뭉치에 아래 세 문장이 있음
  - <s> I am egg </s>
  - <s> Joe I am </s>
  - <s> I do like green and egg </s>

- Bigram count
  - C(<s> I) = 2 / C(I am) = 2 / C(am egg) = 1 /
  - C(do like) = 1 / C(like green) = 1 / C(and egg) = 1 / C(egg </s>) = 2

- Bigram 언어 모델 생성 확률
  - P(I am egg) = P(I | <s>)*P(am | I)*P(egg | am) = 2/3*2/3*1/2 = 2/9
    - P(I | <s>) = C(<s> I) / C(<s>) = 2 / 3
    - P(am | I) = C(I am) / C(I) = 2 / 3
    - P(egg | am) = C(am egg) / C(am) = 1 / 2

9.1 n-gram 언어 모델

- n-gram 언어 모델 ARPA format

\data\
ngram 1=*n1*
ngram 2=*n2*
...
ngram *N*=*nN*
\1-grams:
p w [*bow*]
...
\2-grams:
p w1 w2 [*bow*]
...
N-grams:
p w1 ... wN
...
\end\

The number of unigram
The number of bigram
The number of N-gram
Log probability with 10 base
Back-off weight

```
-0.7079 white lies and
-1.0089 white party is
-1.0089 white party you
-0.7079 white right and
-0.7079 who Chuck Klosterman
-0.7079 who are you
-0.7079 who build the
-1.0089 who cares honestly
-1.0089 who cares who
-0.7079 who comments negative
-0.7079 who died headed
-0.4436 who do you
-1.0457 who else have
-0.1434 who else is
-0.7079 who enjoyed ruining
-0.7079 who feels like
-0.7079 who get me
-0.7079 who gives you
-1.6110 who is commander
```

<an example of 3-gram ARPA format>

9.1 n-gram 언어 모델

- 메모리 문제
  - 어휘의 크기가 10만(10^5)인 경우, tri-gram tuple은 세 단어의 index로 표현됨
  - table 형태로 저장하기 위해서는 3차원 integer table이 있어야 함
  - 이 경우 tri-gram tuple수는 10^{5*3} 이며, 필요한 바이트 수는 $4 * 10^{5*3}B = 4 * 10^3 * 10^3 * 10^3 * 10^3 * 10^3B$ = 4,000TB (1TB 외장하드가 4천 개가 필요함)

9.1 n-gram 언어 모델

- 학습 자료의 부족
  - 예) 특정 방송국이 100년간 방송한 자료가 있다고 하자. 1년에 300일, 하루에 20시간을 방송하였다고 가정하면, 총 수집한 오디오의 양은 600,000시간 분량이다.
  - 영미권 뉴스의 경우 100시간당 100만단어가 발화된다고 알려져 있다. 같은 기준을 적용하여, 100년간의 방송자료에 대해 transcription을 만들면 60억 ($6*10^9$)단어 = (600,000시간 * (100만 단어 / 100시간))가 존재함.
  - Table의 셀 개수가 10^{15}개임을 감안하면, tri-gram tuple의 수가 현저히 부족하다.

9.1 n-gram 언어 모델

- 앞서 설명한 바와 같이 Trigram 언어모델 생성 확률은 아래와 같이 계산한다

$$P(w_i|w_{i-2}, w_{i-1}) = \frac{f(w_{i-2}, w_{i-1}, w_i)}{f(w_{i-2}, w_{i-1})}$$

- 코퍼스를 아무리 많이 모으더라도, 실제 발화에서 나타나는 $f(w_{i-2}, w_{i-1}, w_i)$, $f(w_{i-2}, w_{i-1})$를 적절히 추정하지 못하는 경우가 발생하며, 최악의 경우는 0이 되는 경우
  - $f(w_{i-2}, w_{i-1}, w_i)$가 0이 되는 경우:
    - 문장생성 확률 $P(W) = \prod_{i=1}^{T} P(w_i|w_{i-2}, w_{i-1})$ 중 특정 단어 w_i 생성확률 $P(w_i|w_{i-2}, w_{i-1})$가 0이 되고, 따라서 $P(W)$가 0이 되어버림.
  - $f(w_{i-2}, w_{i-1})$가 0이 되는 경우
    - 0으로 나누는 문제가 발생하여 계산이 불가능

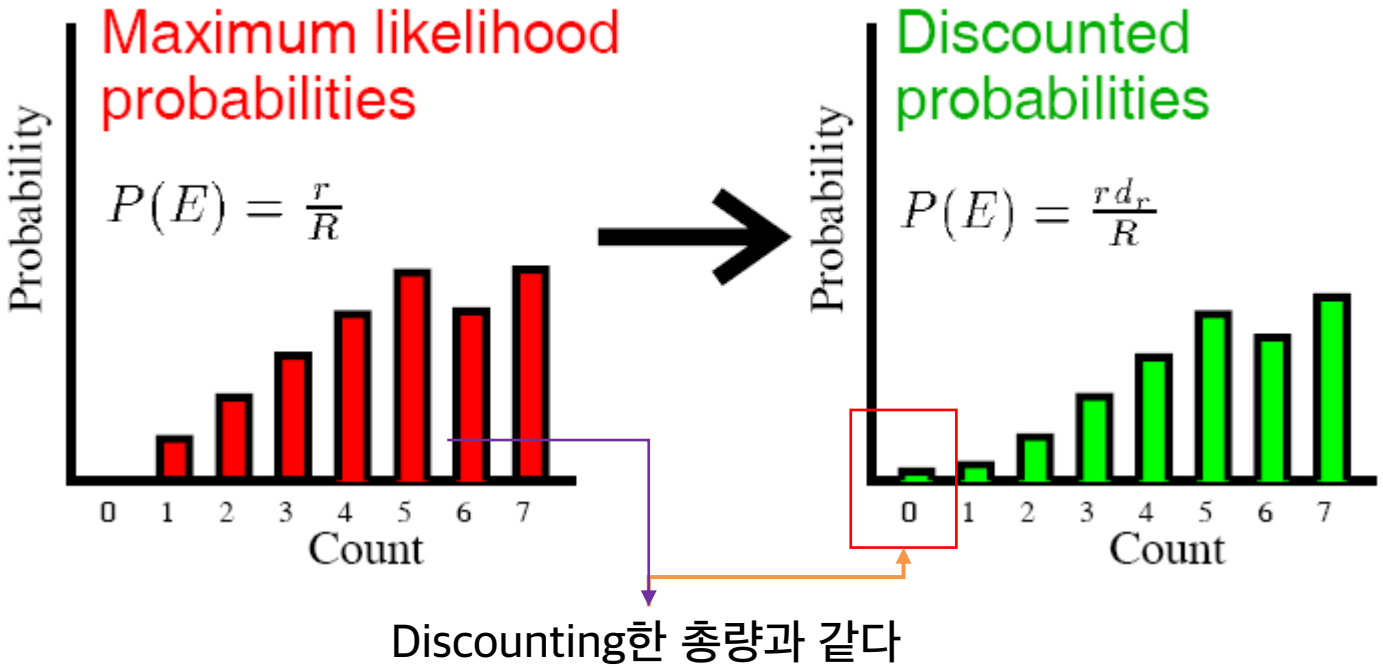
9.1 n-gram 언어 모델

- 위 문제를 아래의 두가지 방법으로 해결한다:

- Discounting & Smoothing
  - 0의 값을 가지는 $f(w_{i-2},\ \ w_{i-1},\ \ w_i)$에 대해 작은 값으로 flooring 시킴

- Backing-off:
  - trigram모델로 언어모델 생성확률 계산 시 $f(w_{i-2},\ w_{i-1}, w_i)$이 작아 적절한 확률 추정이 어려운 경우, 모델링 파워는 낮지만 적은 양의 코퍼스로부터 적절한 확률추정이 가능한 bigram, unigram 언어모델 확률로 대체하는 방법

9.1.1 Discounting & Smoothing



- X축: frequency (count)
- Y축: 해당 frequency를 가지는 tuple들에 대한 언어모델 생성 확률 값들의 총합

9.1.1 Discounting & Smoothing

- 0의 값을 가지는 $f(w_i, \quad w_j, \quad w_k)$에 대해 작은 값으로 flooring하는 과정에서 sum-to-one 제한 ($\sum_k \hat{P}(w_k|w_i, w_j) = 1$)을 만족하지 않게 됨

- 제한을 만족시키기 위해, 관측된 사건의 수를 discount 하여 비관측 사건에 할당. 따라서 언어모델 생성확률은 다음과 같이 수정됨.

$$\hat{P}(w_k|w_i, w_j) \;=\; d(f(w_i, w_j, w_k))\frac{f(w_i, w_j, w_k)}{f(w_i, w_j)}$$

여기서 d(r)을 discount coefficient라 한다

- d(r)의 추정방법에 따라 여러 방법론이 있다

9.1.1 Discounting & Smoothing

- Laplace smoothing (Add-one smoothing)
- Add-k smoothing
- Good Turing smoothing

9.1.1.1 Laplace Smoothing

- 가장 간단한 smoothing 기법으로 add one smoothing 이라고도 함

- 모든 비관측 n-gram과 관측 가능한 n-gram 발생 횟수에 1을 더함

$$P(w_i) = \frac{c(w_i)}{N} \quad \rightarrow \quad PLap(w_i) = \frac{c(w_i) + 1}{N + V} \; (unigram)$$

$$P(w_i|w_{i-1}) = \frac{c(w_{i-1},\ w_i)}{c(w_{i-1})} \quad \rightarrow \quad PLap(w_i|w_{i-1}) = \frac{c(w_{i-1}, w_i) + 1}{c(w_{i-1}) + V} \; (bigram)$$

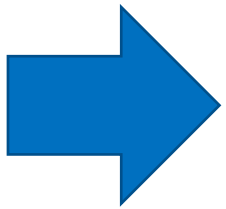
$$(N = Corpus\ size, \qquad V = number\ of\ word)$$

$$N = \sum_w c(w) = \sum_{w_i,w_j} c(w_i w_j) = \sum_{w_i w_j w_k} c(w_i w_j w_k)$$

9.1.1.1 Laplace Smoothing

```
1 buy the book
2 buy the book
3 buy the book
4 buy the book
5 sell the book
6 buy the house
7 buy the house
8 paint the house
```

예제 코퍼스



<s>	8
</s>	8
book	5
buy	6
house	3
paint	1
sell	1
the	8

Unigram 카운트

<s> buy	6
<s> sell	1
<s> paint	1
buy the	6
sell the	1
paint the	1
the book	5
the house	3
book </s>	5
house </s>	3

Bigram 카운트

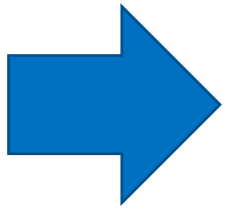
9.1.1.1 Laplace Smoothing

<s>	8
</s>	8
book	5
buy	6
house	3
paint	1
sell	1
the	8

Unigram 카운트

<s> buy	6
<s> sell	1
<s> paint	1
buy the	6
sell the	1
paint the	1
the book	5
the house	3
book </s>	5
house </s>	3

Bigram 카운트



Laplace smoothing

<s>	8
</s>	8
book	5
buy	6
house	3
paint	1
sell	1
the	8

Unigram 카운트

<s> buy	7	<s> <s>	1
<s> sell	2	<s> </s>	1
<s> paint	2	<s> book	1
buy the	7	<s> house	1
sell the	2	...	1
paint the	2		1
the book	6		1
the house	4		1
book </s>	6		1
house </s>	4	the the	1

Bigram 카운트

9.1.1.1 Laplace Smoothing

<s>	8
</s>	8
book	5
buy	6
house	3
paint	1
sell	1
the	8

Unigram 카운트

$$P(the|buy) = \frac{c(buy\ the)\ +\ 1}{c(buy) + V} = \frac{7}{6 + \boxed{7}} = \frac{7}{13}$$

Srilm에서는 단어의 개수를 셀 때 <s>는 제외

<s> buy	7
<s> sell	2
<s> paint	2
buy the	7
sell the	2
paint the	2
the book	6
the house	4
book </s>	6
house </s>	4

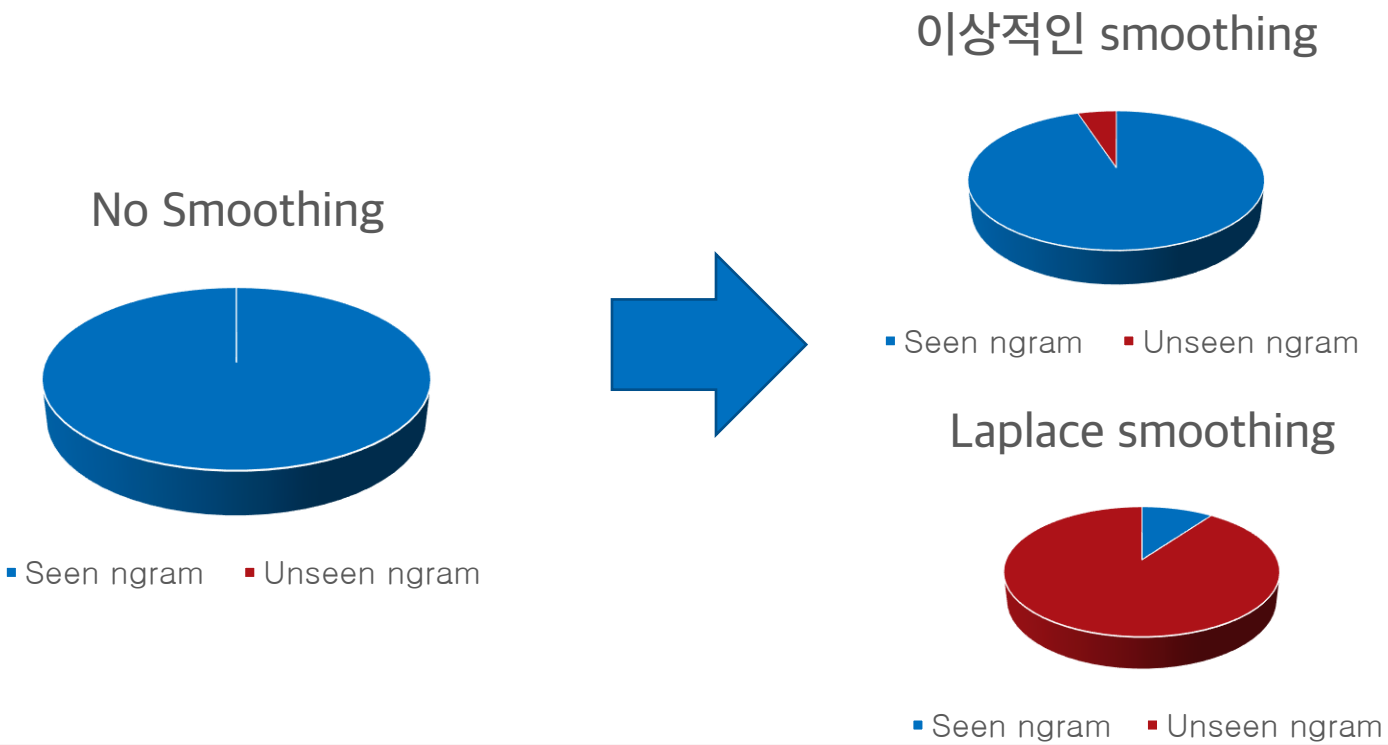
<s> <s>	1
<s> </s>	1
<s> book	1
<s> house	1
...	1
	1
	1
	1
	1
	1
the the	1

Bigram 카운트

$$P(the|the) = \frac{c(the\ the)\ +\ 1}{c(the) + V} = \frac{1}{8 + 7} = \frac{1}{15}$$

9.1.1.1 Laplace Smoothing

- Laplace Smoothing의 문제점
  - 일반적으로 비관측 이벤트가 관측 가능한 이벤트보다 많음
  - 비관측 이벤트에 대해 너무 많은 확률을 할당
  - 다른 smoothing 방법에 비해 성능이 매우 떨어짐



9.1.1.2 Add-k smoothing

- 1을 더하는 대신 1보다 작은 수 k를 더함

$$P(w_i) = \frac{c(w_i)}{N} \quad \rightarrow \quad P_{Add-k}(w_i) = \frac{c(w_i) + k}{N + kV} \; (unigram)$$

$$P(w_i|w_{i-1}) = \frac{c(w_{i-1},\ w_i)}{c(w_{i-1})} \quad \rightarrow \quad P_{Add-k}(w_i|w_{i-1}) = \frac{c(w_{i-1}, w_i) + k}{c(w_{i-1}) + kV} \; (bigram)$$

$$(N = Corpus\ size, \qquad V = number\ of\ word)$$

- Laplace smoothing 보다 성능은 좋지만 여전히 성능이 떨어짐

9.1.1.3 Good turing smoothing

- Good turing estimate
  - n번 관측된 이벤트에 대해 n+1번 관측된 이벤트의 횟수를 사용 (n=0, 1, …)

$$n_r = \left|\{w_i, w_j : C(w_i, w_j) = r\}\right| \text{ (r번 관측된 이벤트의 개수)}$$

$$r^* = C_{GT}(w_{i-1}, w_i) = (r+1)\frac{n_{r+1}}{n_r} \quad (smoothed\ count)$$

$$P_{GT}(w_{i-1}, w_i) = \frac{r^*}{N}, \qquad P_{GT}(w_i|w_{i-1}) = \frac{C_{GT}(w_{i-1}, w_i)}{C(w_i)}$$

9.1.1.3 Good turing smoothing

- Ex)

bigram tuple에 대한 출현 횟수와 카운트

r	n_r
1	138741
2	25413
3	10531
4	5997
5	…

V = 14585
Seen bigrams =
138741+25413+10531+… = 199252
N=615876

$$n_0 = 14585^2 - 199252 = 212522973$$

$$C_{GT(Unseen)} = (0+1) \times \frac{n_1}{n_0} = 0.00065$$

$$P_{GT(Unseen)} = \frac{C_{GT(Unseen)}}{N} = 1.06 \times 10^{-9}$$

C(person she) = 2 C_{GT}(person she) = (2+1)(10531/25413) = 1.243

C(person) = 223 P_{GT}(she | person) = C_{GT}(person she) / 223 = 0.0056

9.1.1.3 Good turing smoothing

- Good turing smoothing의 문제점
  - 모든 카운트들을 대체 할 수 없다 (ex. nr+1 = 0 인 경우 nr은 추정할 수 없다)

- 따라서 Good turing smoothing은 다른 기법들과 함께 쓰임
- SRILM에서는 max 카운트를 설정하고 그 이상의 카운트에 대해서는 smoothing 하지 않음.
- 좀 더 좋은 smoothing을 위하여 back-off 모델이 필요

9.1.2 Back-Off 언어모델

- $f(w_{i-2},\ w_{i-1}, w_i)$이 작아 적절한 n-gram언어모델 생성확률 추정이 어려운 경우, 모델링 파워는 낮지만 적은 양의 코퍼스로부터 적절한 확률추정이 가능한 (n-1)-gram을 사용하여 추정함.
  - Bigram의 예:

Smoothing 된 확률을 사용

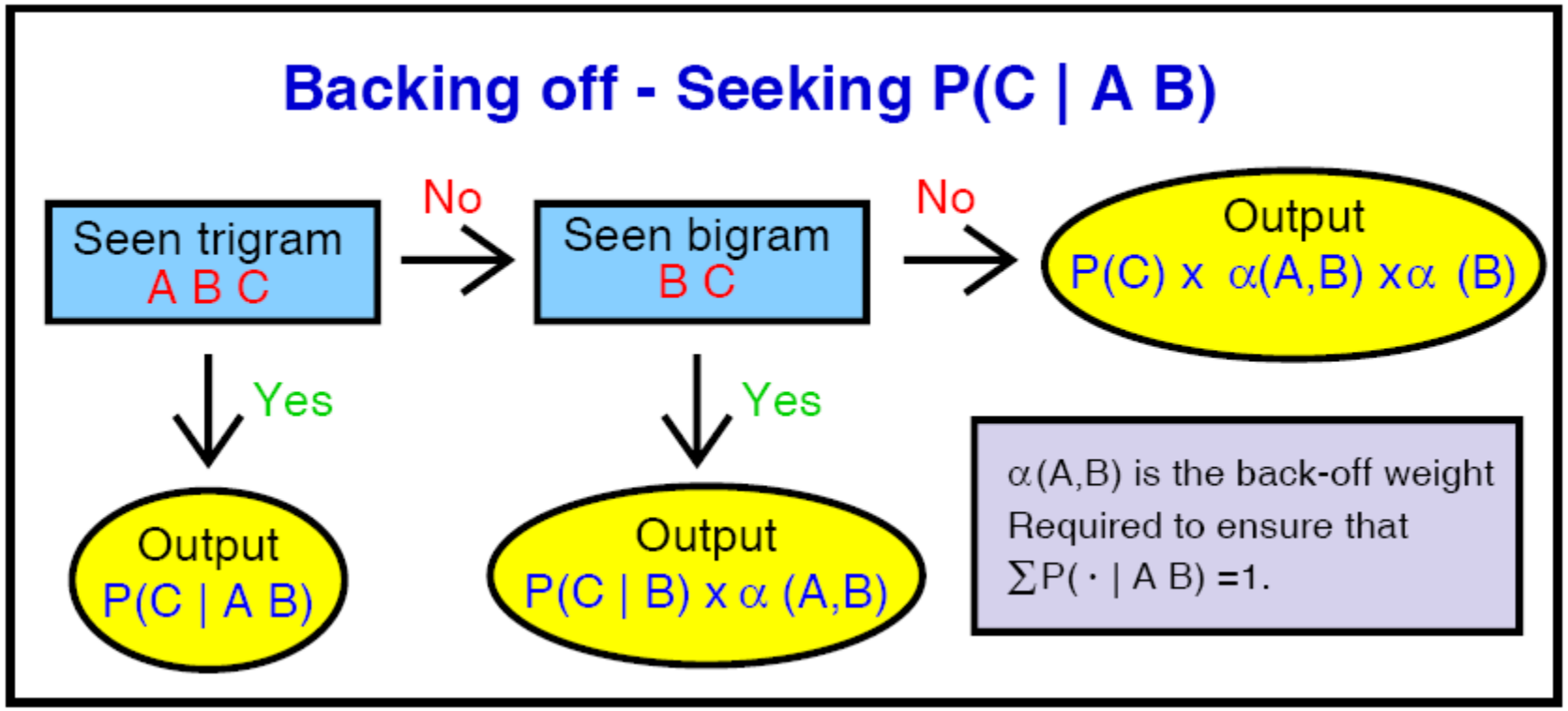
$$\Pr(w_j|w_i) = \begin{cases} d_{f(w_i,w_j)} \dfrac{f(w_i,w_j)}{f(w_i)} & f(w_i,w_j) > C \\ \alpha(w_i)\Pr(w_j) & \text{otherwise} \end{cases}$$

$\alpha(w_i)$는 오른쪽 식을 만족해야 함 $\sum_j \Pr(w_j|w_i) = 1$

C는 n-gram cut-off frequency

- (n-1)-gram언어모델 추정도 어려운 경우, (n-2)-gram, (n-3) -gram… 을 이용하여 추정한다.

9.1.2 Back-Off 언어모델



9.1.2.1 Katz back-off

- Katz back-off 모델은 주로 Good turing smoothing을 사용

$$P_{katz}(w_i|w_{i-1}) = \begin{cases} d_r \dfrac{C(w_{i-1}\, w_i)}{C(w_{i-1})} & (if\ r > 0) \\ \alpha(w_{i-1})\, P(w_i) & (if\ r = 0) \end{cases}$$

$$\alpha(w_{i-1}) = \frac{1 - \sum_{w \in A\ \{w|c(w_{i-1}w)>0\}} P_{katz}(w|w_{i-1})}{1 - \sum_{w \in A\ \{w|c(w_{i-1}w)>0\}} P_{katz}(w)}$$

$$d_r = \begin{cases} 1 \ (r > K) \\ \approx \dfrac{(r+1)n_{r+1}}{r n_r} \end{cases} \qquad (\text{Katz suggest K} = 5)$$

모든 k에 대해 Good-Turing estimate를 적용하는 것이 아니기 때문에 적절한 d_r을 찾아야함

9.1.2.1 Katz back-off

- 출현 횟수가 K 이하의 n-gram의 출현횟수의 합을 m이라 하면 discounting 하기 전은 다음과 같다.

$$\sum_{k=1}^{K} kn_k = m$$

  - Good-turing estimate을 적용하여 discounting을 하면 임의의 비관측 n-gram에 $\frac{n_1}{n_0}$의 카운트가 할당되고 이들의 합은 $\frac{n_1}{n_0} \times n_0 = n_1$이 된다.

- 출현 횟수가 K 이하의 관측 가능한 n-gram의 출현 횟수를 discounting 한 총합이 n1이 되어야 한다.

$$\sum_{k=1}^{K} d_k kn_k = m - n_1$$

$$\leftrightarrow \sum_{k=1}^{K} d_k kn_k = \sum_{k=1}^{K} kn_k - n_1$$

$$\leftrightarrow \sum_{k=1}^{K} (1 - d_k) kn_k = n_1$$

9.1.2.1 Katz back-off

- Good-Turing estimate을 적용하면

$$d_k = \frac{(k+1)n_{k+1}}{kn_k}$$

- 모든 k에 대해서 Good-Turing estimate를 하는 것이 아니기 때문에 적절한 상수를 곱해서 dk‘값을 찾음

$$\mu \sum_{k=1}^{K} (1 - d_k')k\, n_k = \mu \sum_{k=1}^{K} \left(1 - \frac{(k+1)n_{k+1}}{kn_k}\right) k\, n_k = n_1$$

$$\mu[n_1 - (K+1)n_{K+1}] = n_1$$

$$\mu = \frac{1}{1 - \frac{(K+1)n_{K+1}}{n_1}}$$

이를 대입하여 d_k’에 대해 정리하면

$$d_k' = \frac{\frac{(k+1)n_k}{kn_k} - \frac{(K+1)n_{K+1}}{n_1}}{1 - \frac{(K+1)n_{K+1}}{n_1}}$$

9.1.2.1 Katz back-off

- α는 Seen N-gram과 Unseen N-gram의 확률의 합이 1이 되는 값으로 선정

- Bigram의 경우) 집합 A와 B를 정의

$$A\ \{w|c(w_{i-1}w) > 0\}$$
$$\mathrm{B}\ \{w|c(w_{i-1}w) = 0\}$$

- 이때 Back-off 모델에서 Unseen N-gram의 확률을 다음과 같이 정의

$$P_{katz}(w_i|w_{i-1}) = \alpha(w_{i-1})\,P(w_i)$$

- 따라서 다음과 같이 유도됨

$$\sum_{w\in A} P_{katz}(w|w_{i-1}) + \sum_{w\in B} \alpha(w_{i-1})P(w) = 1$$
$$\alpha(w_{i-1}) = \frac{1 - \sum_{w\in A} P_{katz}(w|w_{i-1})}{1 - \sum_{w\in A} P_{katz}(w)}$$

9.1.2.1 Katz back-off

- 정리하면 다음과 같은 식이 완성된다

$$P_{katz}(w_i|w_{i-1}) = \begin{cases} \dfrac{C(w_{i-1}\, w_i)}{C(w_{i-1})} \ (if\ r > K) \\ d_r \dfrac{C(w_{i-1}\, w_i)}{C(w_{i-1})} \ (if\ K \geq r > 0) \\ \alpha(w_{i-1})\, P(w_i) \quad (if\ r = 0) \end{cases}$$

$$d_r = \frac{\dfrac{(r+1)n_{r+1}}{rn_r} - \dfrac{(K+1)n_{K+1}}{n_1}}{1 - \dfrac{(K+1)n_{K+1}}{n_1}}$$

$$\alpha(w_{i-1}) = \frac{1 - \sum_{w \in A\, \{w|c(w_{i-1}w)>0\}} P_{katz}(w|w_{i-1})}{1 - \sum_{w \in A\, \{w|c(w_{i-1}w)>0\}} P_{katz}(w)}$$

9.1.2.1 Katz back-off

```
K.Cay
K.Ache
Cay
```

예제 코퍼스

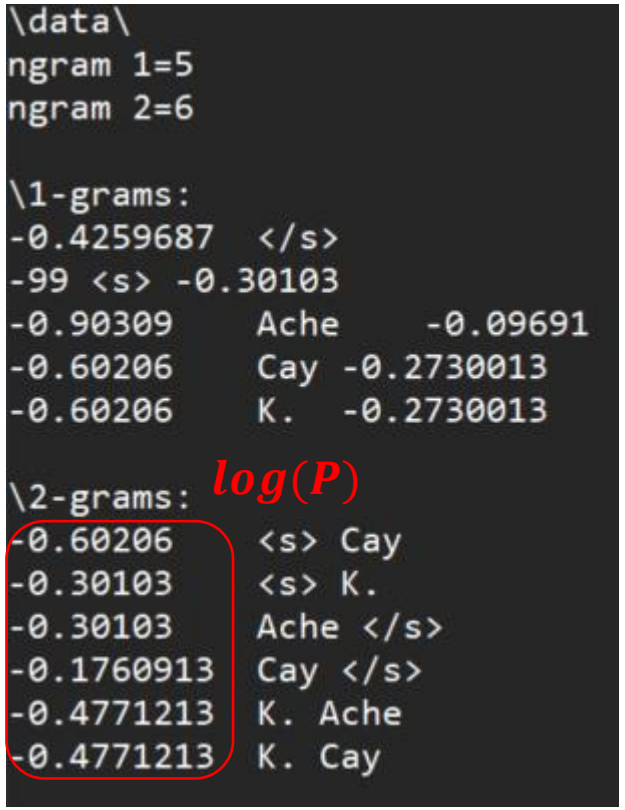
```
\data\
ngram 1=5
ngram 2=6

\1-grams:
-0.4259687   </s>
-99 <s> -0.30103
-0.90309     Ache     -0.09691
-0.60206     Cay -0.2730013
-0.60206     K.   -0.2730013

\2-grams:
-0.60206     <s> Cay
-0.30103     <s> K.
-0.30103     Ache </s>
-0.1760913   Cay </s>
-0.4771213   K. Ache
-0.4771213   K. Cay
```

ARPA format

9.1.2.1 Katz back-off



ARPA format

- SRILM에서는 다음과 같은 방식으로 바이그램의 확률을 조정
  1. n_r 값을 구한다
  2. Good-Turing estimate이 가능한 Max count 값을 찾는다
  3. 2에서 구한 값을 토대로 d_r 값을 구한다.
     (이때 구할 수 없는 경우는 1로 설정)
  4. d_r을 토대로 바이그램의 확률의 합이 1이 되거나 1보다 큰 경우 분모를 늘려 나가며 확률의 합이 1보다 작게 만든다.

9.1.2.1 Katz back-off

```
\data\
ngram 1=5
ngram 2=6

\1-grams:
-0.4259687   </s>
-99 <s> -0.30103
-0.90309     Ache     -0.09691
-0.60206     Cay -0.2730013
-0.60206     K.   -0.2730013

\2-grams:
-0.60206     <s> Cay
-0.30103     <s> K.
-0.30103     Ache </s>
-0.1760913   Cay </s>
-0.4771213   K. Ache
-0.4771213   K. Cay
```

$log(P)$

ARPA format

1. n_r 값을 구한다

$$n_1 = 4 \;\; n_2 = 2$$

2. Good-Turing estimate이 가능한 Max count 값을 찾는다

n_2는 Good-Turing estimate 를 적용 할 수 없으니 (n_3값이 없기 때문) Max count는 1이 됨.

3. 2에서 구한 값을 토대로 d_r 값을 구한다.

(이때 구할 수 없는 경우나 $\frac{(r+1)n_{r+1}}{rn_r}$ >1 인 경우 1로 설정)

$$d_r = \frac{\frac{(r+1)n_{r+1}}{rn_r} - \frac{(K+1)n_{K+1}}{n_1}}{1 - \frac{(K+1)n_{K+1}}{n_1}}$$

$d_1 = 0$ 이 되므로 1로 설정

d_2는 구할 수 없으므로 1로 설정

9.1.2.1 Katz back-off

```
\data\
ngram 1=5
ngram 2=6

\1-grams:
-0.4259687   </s>
-99 <s> -0.30103
-0.90309     Ache     -0.09691
-0.60206     Cay -0.2730013
-0.60206     K.   -0.2730013

\2-grams:
-0.60206     <s> Cay
-0.30103     <s> K.
-0.30103     Ache </s>
-0.1760913   Cay </s>
-0.4771213   K. Ache
-0.4771213   K. Cay
```

$log(P)$

ARPA format

4. d_r을 토대로 바이그램의 확률의 합이 1이 되거나 1보다 큰 경우 분모를 늘려 나가며 확률의 합이 1보다 작게 만든다.

$$P(Cay|<s>) \times d_1 = \frac{1}{3} \times 1 = \frac{1}{3}$$

$$P(K.|<s>) \times d_2 = \frac{2}{3} \times 1 = \frac{2}{3}$$

바이그램의 확률의 합이 1이 되므로 분모를 1씩 증가 시킴

$$P(Cay|<s>) \times d_1 = \frac{1}{3+1} \times 1 = \frac{1}{4}$$

$$P(K.|<s>) \times d_2 = \frac{2}{3+1} \times 1 = \frac{1}{2}$$

따라서 $1 - \left(\frac{1}{4} + \frac{1}{2}\right) = \frac{1}{4}$의 확률이 Unseen 바이그램에 할당

9.1.2.1 Katz back-off

```
\data\
ngram 1=5
ngram 2=6

\1-grams:
-0.4259687   </s>
-99 <s> -0.30103
-0.90309     Ache     -0.09691
-0.60206     Cay -0.2730013
-0.60206     K.  -0.2730013

\2-grams:
-0.60206     <s> Cay
-0.30103     <s> K.
-0.30103     Ache </s>
-0.1760913   Cay </s>
-0.4771213   K. Ache
-0.4771213   K. Cay
```

$log(P)$ $log(\alpha(K.))$

ARPA format

$$\alpha(K.) = \frac{1 - \sum_{w \in A} P_{katz}(w|w_{i-1})}{1 - \sum_{w \in A} P_{katz}(w)}$$

A = {Ache, Cay}

$$\alpha(K.) = \frac{1 - (P(Ache|K.) + P(Cay|K.))}{1 - (P(Ache) + P(Cay)}$$

P(Ache| $K.$) = $\frac{1}{2+1} = {}^{1}/_{3}$

P(Cay| $K.$) = $\frac{1}{2+1} = {}^{1}/_{3}$

앞서 설명한 방식으로 discount 된 확률

P(Ache) = $10^{-0.90309} = {}^{1}/_{8}$

P(Cay) = $10^{-0.60206} = {}^{1}/_{4}$

$\alpha(K.) = {}^{8}/_{15}$ $log(\alpha(K.)) = -0.2730013$

9.1.2.1 Katz back-off

```
\data\
ngram 1=5
ngram 2=6

\1-grams:
-0.4259687   </s>
-99 <s> -0.30103
-0.90309     Ache     -0.09691
-0.60206     Cay -0.2730013
-0.60206     K.  -0.2730013

\2-grams:
-0.60206     <s> Cay
-0.30103     <s> K.
-0.30103     Ache </s>
-0.1760913   Cay </s>
-0.4771213   K. Ache
-0.4771213   K. Cay
```

$log(P)$ $log(\alpha(K.))$

ARPA format

확률의 합이 1이 되는지 알아보자

P(Ache| $\boldsymbol{K}.$) + P(Cay| $\boldsymbol{K}.$) + P(</s>| $\boldsymbol{K}.$) + P(K.| $\boldsymbol{K}.$) = 1 이 되어야함

P(Ache| $\boldsymbol{K}.$) = $\frac{1}{3}$

P(Cay| $\boldsymbol{K}.$) = $\frac{1}{3}$

P(</s>| $\boldsymbol{K}.$) =$\boldsymbol{\alpha}(\boldsymbol{K}.) \times \boldsymbol{P}(</\boldsymbol{s}>) = \frac{8}{15} \times \frac{3}{8} = \frac{1}{5}$

P(K.| $\boldsymbol{K}.$) =$\boldsymbol{\alpha}(\boldsymbol{K}.) \times \boldsymbol{P}(\boldsymbol{K}.) = \frac{8}{15} \times \frac{1}{4} = \frac{2}{15}$

P(Ache| $\boldsymbol{K}.$) + P(Cay| $\boldsymbol{K}.$) + P(</s>| $\boldsymbol{K}.$) + P(K.| $\boldsymbol{K}.$)

$= \frac{1}{3} + \frac{1}{3} + \frac{1}{5} + \frac{2}{15} = 1$

9.1.2.1 Katz back-off

```
buy the book
buy the book
buy the book
buy the book
sell the book
buy the house
buy the house
paint the house
```

예제 코퍼스

```
\1-grams:
-0.60206    </s>
-99 <s> -0.8750612
-0.80618    book    -0.6532125
-0.7269987  buy -0.7201593
-1.028029   house   -0.8750613
-1.50515    paint   -0.4771213
-1.50515    sell    -0.4771213
-0.60206    the -1.176091

\2-grams:
-0.2218488  <s> buy
-0.8239087  <s> paint
-0.8239087  <s> sell
-0.07918125 book </s>
-0.06694679 buy the
-0.04575749 house </s>
-0.1249387  paint the
-0.1249387  sell the
-0.30103    the book
-0.3467875  the house
```

ARPA format

9.1.2.1 Katz back-off

```
\1-grams:
-0.60206    </s>
-99 <s> -0.8750612
-0.80618    book    -0.6532125
-0.7269987  buy -0.7201593
-1.028029   house   -0.8750613
-1.50515    paint   -0.4771213
-1.50515    sell    -0.4771213
-0.60206    the -1.176091

\2-grams:
-0.2218488  <s> buy
-0.8239087  <s> paint
-0.8239087  <s> sell
-0.07918125 book </s>
-0.06694679 buy the
-0.04575749 house </s>
-0.1249387  paint the
-0.1249387  sell the
-0.30103    the book
-0.3467875  the house
```

ARPA format

1. n_r 값을 구한다

$$n_1 = 4$$

$$n_2 = 0$$

$$n_3 = 2$$

$$n_4 = 0$$

$$n_5 = 2$$

$$n_6 = 2$$

9.1.2.1 Katz back-off

```
\1-grams:
-0.60206    </s>
-99 <s> -0.8750612
-0.80618    book    -0.6532125
-0.7269987  buy -0.7201593
-1.028029   house   -0.8750613
-1.50515    paint   -0.4771213
-1.50515    sell    -0.4771213
-0.60206    the -1.176091

\2-grams:
-0.2218488  <s> buy
-0.8239087  <s> paint
-0.8239087  <s> sell
-0.07918125 book </s>
-0.06694679 buy the
-0.04575749 house </s>
-0.1249387  paint the
-0.1249387  sell the
-0.30103    the book
-0.3467875  the house
```

ARPA format

2. Good-Turing estimate이 가능한 Max count 값을 찾는다

n_6는 Good-Turing estimate 를 적용 할 수 없으니 (n_7값이 없기 때문) Max count는 5가 됨.

3. 2에서 구한 값을 토대로 d_r 값을 구한다.
(이때 구할 수 없는 경우는 1로 설정)

$$d_1 = 1.5 \quad d_2 = 1 \quad d_3 = 1.5 \quad d_4 = 1 \quad \boxed{d_5 = 1} \quad d_6 = 1$$

$\frac{(r+1)n_{r+1}}{rn_r}$ > 1 이므로 1로 설정

9.1.2.1 Katz back-off

```
\1-grams:
-0.60206    </s>
-99 <s> -0.8750612
-0.80618    book    -0.6532125
-0.7269987  buy -0.7201593
-1.028029   house   -0.8750613
-1.50515    paint   -0.4771213
-1.50515    sell    -0.4771213
-0.60206    the -1.176091

\2-grams:
-0.2218488  <s> buy
-0.8239087  <s> paint
-0.8239087  <s> sell
-0.07918125 book </s>
-0.06694679 buy the
-0.04575749 house </s>
-0.1249387  paint the
-0.1249387  sell the
-0.30103    the book
-0.3467875  the house
```

ARPA format

4. d_r을 토대로 바이그램의 확률의 합이 1이 되거나 1보다 큰 경우 분모를 늘려 나가며 확률의 합이 1보다 작게 만든다.

$$P(buy|<s>) \times d_6 = \frac{6}{8} \times 1 = \frac{6}{8}$$

$$P(paint|<s>) \times d_1 = \frac{1}{8} \times 1.5 = \frac{1.5}{8}$$

$$P(sell|<s>) \times d_1 = \frac{1}{8} \times 1.5 = \frac{1.5}{8}$$

바이그램의 확률의 합이 1보다 크기때문에 분모를 1씩 증가 시킴

$$P(buy|<s>) \times d_6 = \frac{6}{8+1} \times 1 = \frac{6}{9}$$

$$P(paint|<s>) \times d_1 = \frac{1}{8+1} \times 1.5 = \frac{1.5}{9}$$

$$P(sell|<s>) \times d_1 = \frac{1}{8+1} \times 1.5 = \frac{1.5}{9}$$

9.1.2.1 Katz back-off

```
\1-grams:
-0.60206    </s>
-99 <s> -0.8750612
-0.80618    book    -0.6532125
-0.7269987  buy -0.7201593
-1.028029   house   -0.8750613
-1.50515    paint   -0.4771213
-1.50515    sell    -0.4771213
-0.60206    the -1.176091

\2-grams:
-0.2218488  <s> buy
-0.8239087  <s> paint
-0.8239087  <s> sell
-0.07918125 book </s>
-0.06694679 buy the
-0.04575749 house </s>
-0.1249387  paint the
-0.1249387  sell the
-0.30103    the book
-0.3467875  the house
```

ARPA format

바이그램의 확률의 합이 1이기 때문에 분모를 1씩 증가 시킴

$$P(\boldsymbol{buy}|<\boldsymbol{s}>) \times d_6 = \frac{6}{8+2} \times 1 = \frac{6}{10}$$

$$P(\boldsymbol{paint}|<\boldsymbol{s}>) \times d_1 = \frac{1}{8+2} \times 1.5 = \frac{1.5}{10}$$

$$P(\boldsymbol{sell}|<\boldsymbol{s}>) \times d_1 = \frac{1}{8+2} \times 1.5 = \frac{1.5}{10}$$

$$\log\left(P(\boldsymbol{buy}|<\boldsymbol{s}>)\right) = -0.2218488$$

$$\log\left(P(\boldsymbol{paint}|<\boldsymbol{s}>)\right) = -0.8239087$$

$$\log\left(P(\boldsymbol{sell}|<\boldsymbol{s}>)\right) = -0.8239087$$

9.1.2.1 Katz back-off

```
\1-grams:
-0.60206    </s>
-99 <s> -0.8750612
-0.80618    book    -0.6532125
-0.7269987  buy -0.7201593
-1.028029   house   -0.8750613
-1.50515    paint   -0.4771213
-1.50515    sell    -0.4771213
-0.60206    the -1.176091

\2-grams:
-0.2218488  <s> buy
-0.8239087  <s> paint
-0.8239087  <s> sell
-0.07918125 book </s>
-0.06694679 buy the
-0.04575749 house </s>
-0.1249387  paint the
-0.1249387  sell the
-0.30103    the book
-0.3467875  the house
```

ARPA format

$$\alpha(< s >) = \frac{1 - (P(buy| < s >) + P(sell| < s >) + P(paint| < s >))}{1 - ((P(buy) + P(sell) + P(paint))}$$

$$\alpha(< s >) = \frac{1 - (\frac{6}{10} + \frac{1.5}{10} + \frac{1.5}{10})}{1 - (\frac{6}{32} + \frac{1}{32} + \frac{1}{32})} = \frac{2}{15}$$

$$\log(\alpha(< s >)) = -0.8750612$$

9.1.3 Language Model Scaling

$$\boldsymbol{arg_w max\, P(O|W)P(W)}$$

- 음향모델의 생성확률 $\boldsymbol{P(O|W)}$는 매 frame마다 $\boldsymbol{a_{ij}b_j(o_t)}$가 곱해진다. 반면, $\boldsymbol{P(W)}$는 매 단어마다 $P(w_i|w_{i-2}, w_{i-1})$가 곱해진다.
- 따라서 음향모델의 생성확률의 영향이 상대적으로 훨씬 크다.
- 이 문제점을 보완하기 위해, 언어 모델 Scaling factor alpha를 이용하여 $\boldsymbol{P(O|W)P(W)}$를 아래와 같이 계산한다.

$$\log p(\mathbf{A}|\mathbf{W}) + \alpha \log P(\mathbf{W})$$

- α 는 실험을 통해 계산된다. 영어 뉴스 인식의 경우 α 는 보통 2.0정도가 사용된다.

9.1.4 Text Corpora Normalization

- 코퍼스는 사용하기 전에 '정규화' 되어야 함.
  - 사용할 수 없는 부분들은(예 : 표와 같은) 폐기되어야 하고, 데이터는 표준형식으로 저장 해야함.
  - 문장들은 개별적으로 태그를 지정하며, 숫자 / 날짜는 발음에 따라 처리해야함.

9.1.5 언어 모델 평가척도(Perplexity)

- 연속음성인식용 언어 모델의 품질은 음성인식율로 평가하는 것이 가장 좋다.
- 그러나 언어모델만을 연구하는 입장에서는, 음성인식율로 언어 모델 품질을 평가하는 경우에 음향모델과 디코딩 네트워크를 구현하고, 음성코퍼스를 마련해야 하는 등의 어려움이 있다.
- 따라서, 텍스트만을 이용한 언어모델 품질 척도가 필요하다.

9.1.5 언어 모델 평가척도(Perplexity)

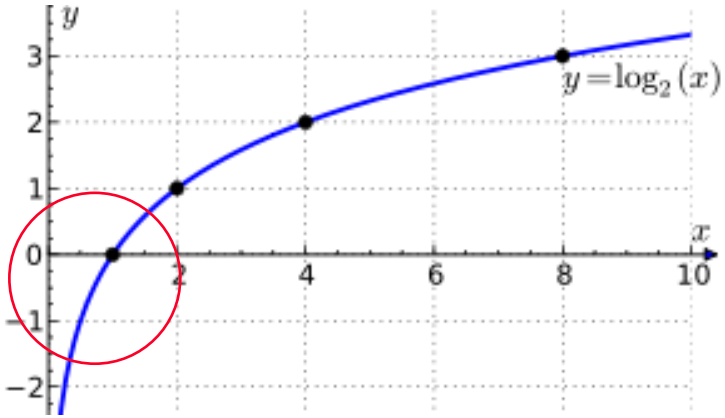
- LogProb(LP)는 각 단어 별 로그 n-gram생성확률의 산술 평균으로 정의한다.

$$\mathrm{LP} = \lim_{n\to\infty} -\frac{1}{N}\sum_{i=1}^{N} \log_2 P(w_i|w_{i-1}\ldots w_1)$$

  - 로그를 빼내면 단어 별 생성확률의 기하평균으로 이해할 수 있음
  - lim n->∞ 는 수집한 코퍼스의 양이 매우 크다는 것을 의미함.

9.1.5 언어 모델 평가척도(Perplexity)

- log(x) 함수는 x가 $0<x<1$ 인 경우 음수 값을 가진다. 따라서 각각의 $\log_2 P(w_i|w_{i-2},\ldots,w_1)$가 음수가 됨.
- 어떠한 평가 척도의 값이 음수가 되는 경우 직관적 의미 해석이 어려워 짐
  - 값이 커지는 것이 좋은 것인가? 절대값이 커지는 경우가 좋은 것인가?
- –1을 곱하여 LP의 값이 양수로 바꾸어 줌



9.1.5 언어 모델 평가척도(Perplexity)

- Perplexity(PP)는 LP를 이용하여 다음과 같이 정의함

$$\mathrm{PP} = 2^{\mathrm{LP}}$$

- ($a\log_b c = c\log_b a$)를 이용하면 PP는 아래와 같이 정리됨

$$\mathrm{PP} = \lim_{N\to\infty} \left\{ P(w_1 w_2 \ldots w_N)^{-\frac{1}{N}} \right\}$$

- PP의 의미는 단어 당 n-gram 생성확률의 기하 평균의 역수

9.1.5 언어 모델 평가척도(Perplexity)

- n-gram 생성 확률의 기하평균
  - 어휘의 크기가 10만인 경우 최악의 언어 모델은 어떤 언어 모델인가?
  - 비슷한 예로, 내일 일기예보를 하는데 최악의 일기예보는 무엇인가?
    - 맑을 확률 1/3, 흐릴 확률 1/3, 비옥 확률 1/3
    - 일기예보가 절대 틀리지 않지만, 아무런 정보를 주지 않음
    - N개의 카테고리가 가지는 확률이 각각 1/n인 경우 엔트로피가 가장 높음
  - 따라서 어휘의 크기가 10만의 경우 최악의 언어모델은 현재 단어로써 어휘내의 첫번째 단어에 대해 생성확률 1/10만, 어휘내의 두번째 단어에 대해서 1/10만, …, 어휘내의 10만 번째 단어에 대해서 1/10만의 확률을 생성하는 언어모델임

9.1.5 언어 모델 평가척도(Perplexity)

- 앞 페이지의 최악의 언어모델의 경우, 평균 단어 별 n-gram생성 확률의 기하평균은 1/10만이고 PP=10만이 됨

- 따라서 PP는 언어모델에 의해 추정되는 다음단어의 평균 수(average branching factor)로 이해할 수 있음

- 동일한 어휘 크기와 도메인에 대해서, PP가 작은 언어모델이 일반적으로 좋은 언어모델임
  - PP의 이론상 최대값은? : 어휘의 크기
  - PP의 이론상 최소값은? : 1
    - 음성을 듣지않고 지금까지 나온 단어만을 가지고 다음 단어가 정확히 예측된다는 의미임. 그러나 이는 불가능함

9.1.6 Relationship between PP and WER

- 일반적으로 perplexity가 감소함에 따라 word error rate (WER)도 감소함

- 많은 실험을 통해 통용되는 PP와 WER의 관계는 다음과 같음

$$WER = k\sqrt{(PP)}$$

where k : task dependent constant

9.1.7 언어모델 Interpolation

- 두개 이상의 언어모델을 이용하여 언어 모델 생성확률값을 계산하는 방법.

$$\hat{P}(W) = \sum_i^N \lambda_i P_i(W)$$

$$\text{where, } \sum_i^N \lambda_i = 1$$

- 예를 들어, SMS dictation용 언어모델을 만드는 경우, 실제 사람들이 송수신한 많은 양의 SMS를 수집하여 언어모델을 학습하는 것이 가장 좋다
- 그러나, SMS문장을 수집하는 것은 비용 및 시간의 문제로 많은 양을 수집하는 것에 한계가 있다
- 이 경우, 웹 크롤링 등을 통해 대용량으로 수집한 말뭉치를 이용하여 학습한 언어모델 P_1과, 타겟 도메인에서 수집한 소용량 말뭉치를 이용해 학습한 언어모델 P_2를 결합해서 언어모델 생성확률을 계산한다.

9.1.7 언어모델 Interpolation

- 타겟 도메인에서 수집한 Evaluation데이터에 대해 생성확률이 최대가 되도록 λ값을 설정한다
- 그러나, 타겟 도메인에서 수집한 자료가 적은 상황에서 interpolation을 사용하기 때문에, 별도의 evaluation데이터를 모으는 것은 힘들다
- 이 경우 Cross Validation 방법을 사용함
  - 예) 10 cross validation
    - 전체의 자료를 10등분 후 9/10를 이용하여 언어 모델을 학습
    - 나머지 1/10을 이용하여 λ를 추정
    - 같은 방법을 반복하면, 10개의 λ를 얻음
    - 10개의 λ를 기하 평균하여 사용함

9.2 카테고리 기반 언어모델

- 언어모델 구축의 주요문제점인 sparseness를 해결하기 위해 카테고리 모델을 사용
- 카테고리 히스토리가 주어졌을 때, 현재단어의 카테고리를 생성하고, 현재단어의 카테고리가 주어졌을 때 현재단어를 생성하는 방법을 많이 사용함

$$P(w_i|w_{i-2}, w_{i-1}) = P(c_i|c_{i-2}, c_{i-1})P(w_i|c_i)$$

c_{i-2}, c_{i-1}, c_i는 단어열 w_{i-2}, w_{i-1}, w_i 에 대응하는 카테고리열

9.2 카테고리 기반 언어모델

- 장점: 추정해야 하는 파라미터 개수를 대폭 줄일 수 있음
  - 예: 어휘크기 10만(10^5), 카테고리 개수 150인 경우
    - 단어 tri-gram 언어모델: $P(w_i|w_{i-2}, w_{i-1}) = \frac{f(w_{i-2},\ w_{i-1},\ w_i)}{f(w_{i-2},\ w_{i-1})}$에 따라 추정해야 하는 파라미터의 수는 아래와 같음
      - $(10^5 * 10^5 * 10^5) + (10^5 * 10^5) = 1.00001 * 10^{15}$개
    - 카테고리 tri-gram 언어모델: $P(c_i|c_{i-2}, c_{i-1})P(w_i|c_i)$에서 추정해야하는 파라미터의 수는 아래와 같음
      - $(150^3+150^2) + (150 + 150 * 10^5) = 1.839765 * 10^7$개
- 카테고리 기반 언어모델의 주요 문제:
  - 카테고리 선택 방법 (예: 품사)
  - 단어가 하나의 카테고리에만 속하는가? (예: 명사 can은 '깡통', 조동사 can은 '할 수 있다')
  - 모델 성능

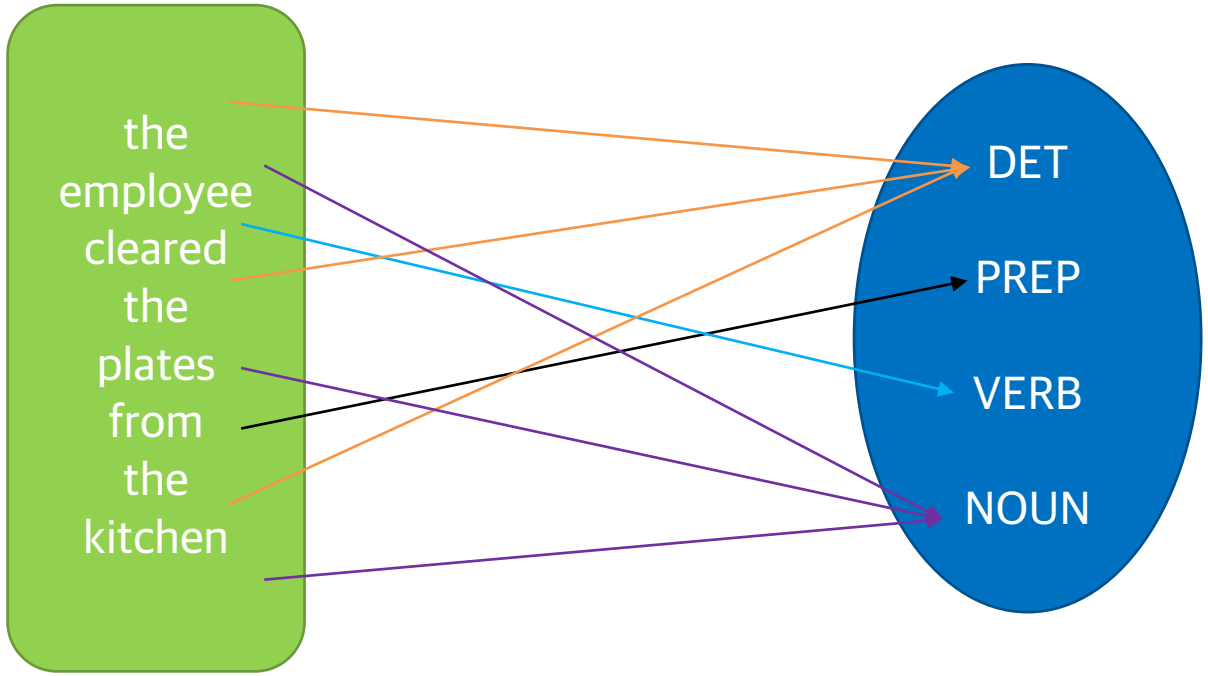
9.2.1 품사기반 카테고리 언어모델

- 단어 카테고리를 품사로 정의함
- 품사: 단어를 기능, 형태, 의미에 따라 나눈 갈래 [네이버 사전]
- 영어에서 많이 쓰이는 예
  - Commonly listed English parts of speech are noun, verb, adjective, adverb, pronoun, preposition, conjunction, interjection, and sometimes numeral, article, or determiner.

Part of Speech	Explanation	Examples
Nouns	A word that names a person, a place or a thing	Boy, Sam, cat, Paris
Pronouns	A word that is used instead of a noun	He, my, yourself
Adjectives	A word that describes a person or thing	pretty, easy, fat
Verbs	A word or group of words that express an action or a state	go, jump, be, think
Adverbs	A word that describes or gives more information about a verb, an adjective, another adverb, or even the entire sentence	quickly, tomorrow, outside
Prepositions	A word that is used before a noun or a pronoun to connect it to another word in the sentence. It is usually used to show location, direction, time, and so forth.	on, in, to, from, of
Conjunctions	A word that joins parts of a sentence together	and, or, but
Interjections	A short sound, word or phrase used to express the speaker's emotion.	Wow, hmm, well, oh dear

9.2.1 품사기반 카테고리 언어모델

- 품사 태깅: 단어들의 의미와 문맥에 기반하여 대응되는 품사를 표시하는 작업



9.2.1 품사기반 카테고리 언어모델

- The Lancaster-Oslo/Bergen (LOB) Corpus Tag-set

IDX	Tag	Description	Examples
1	&FO	formula	10*:-1**: dE *:238**:U a*;n**; T*:-3/2**: E*;p**;(P) R*?8r(cdE.cde) ... [See note 1]
2	&FW	foreign word	de Welt von Retour Flamme route Musique Ancienne Pro unheimliche Opus baraka Biennale Internationale Nov um sine die cantabile letzt bru"cke ...
3	!	exclamation mark	!
4	(	opening parenthesis	(
...			
82	NNU"	noun, abbreviated unit of measure ment, ditto	\0cent cent \0yd [See note 4]
83	NNUS	noun, abbreviated unit of measure ment, plural	\0pts \0yds \0gns *+s \0pp \0mins \0hrs \0revs \0galls \0lbs \0ins [See note 4]
84	NP	noun, singular, proper	Trevor Williams Michael Manchester Foot-Griffiths Bell Karen Roy Dennis Welensky Rhodesia Nkumbula Macleod Julius Accra Ellender Adenauer George Enoch France Corell-Barnes Selwyn ...
85	NP$	noun, singular, proper, genitive	Cheung's Griffith's Oxford's England's Guy's Swansea's Conroy's Zealand's Kent's London's Reid's Margaret's Windsor's Chatterley's Nancy's Sibelius's Shakespeare's Khruschev's ...
...			
150	WPR	WH-pronoun, relative, nominative or accusative	who that
151	WRB	WH-adverb	when wherever where how why however whenever wherein whereby whence whereof whereunto whereon
152	XNOT	negator	not n't na
153	ZZ	letter of the alphabet	G-91 F B zh2014 A T-34 bf alp P A20 X D pi O F11309 a b Q M R4 x z q y H M1 J1 M2 J2 n S U c e d f ...

http://www.comp.leeds.ac.uk/amalgam/tagsets/lob.html

9.2.1 품사기반 카테고리 언어모델

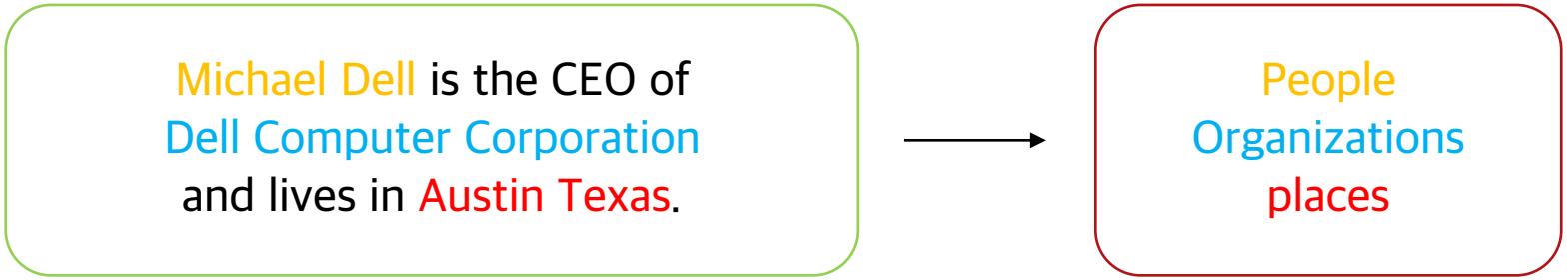
- American National Corpus (ANC)
  - Frequency Data, Written & Spoken, Sorted by count

Word	Lemma	POS	Count	Word	Lemma	POS	Count	Word	Lemma	POS	Count
the	the	DT	1204816	by	by	IN	98472	had	have	VBD	48052
of	of	IN	606545	he	he	PRP	98408	when	when	WRB	47301
and	and	CC	595372	this	this	DT	96575	can	can	MD	46908
to	to	TO	533653	not	not	RB	96492	know	know	VBP	46581
a	a	DT	490433	we	we	PRP	95024	who	who	WP	46273
in	in	IN	409406	or	or	CC	92829	which	which	WDT	44725
it	it	PRP	255012	from	from	IN	88433	their	their	PRP$	44191
is	be	VBZ	227908	have	have	VBP	75789	said	say	VBD	42294
for	for	IN	204432	an	an	DT	73833	have	have	VB	41838
i	i	PRP	188426	uh	uh	UH	71385	she	she	PRP	41716
you	you	PRP	179282	that	that	WDT	70885	been	be	VBN	41507
that	that	IN	168659	were	be	VBD	69385	well	well	RB	40404
was	be	VBD	158470	do	do	VBP	68868	no	no	DT	37856
i	i	NNP	157306	his	his	PRP$	59406	than	than	IN	37807
with	with	IN	150610	about	about	IN	58578	some	some	DT	37701
on	on	IN	141239	has	have	VBZ	57275	will	will	MD	37536
's	be	VBZ	127676	if	if	IN	57238	because	because	IN	37154
's	's	POS	111857	just	just	RB	55819	other	other	JJ	37040
but	but	CC	111337	what	what	WP	55752	did	do	VBD	36141
be	be	VB	108187	like	like	IN	54490	me	me	PRP	34866
as	as	IN	107588	my	my	PRP$	53275	out	out	IN	33928
they	they	PRP	106782	yeah	yeah	NN	52301	'm	be	VBP	33489
are	be	VBP	106108	would	would	MD	51778	them	them	PRP	33207
n't	n't	RB	105380	all	all	DT	50687	also	also	RB	32854
at	at	IN	100272	there	there	EX	50605	're	be	VBP	32626
that	that	DT	98949	so	so	RB	49919	people	people	NNS	32524

http://www.anc.org/data/anc-second-release/frequency-data/

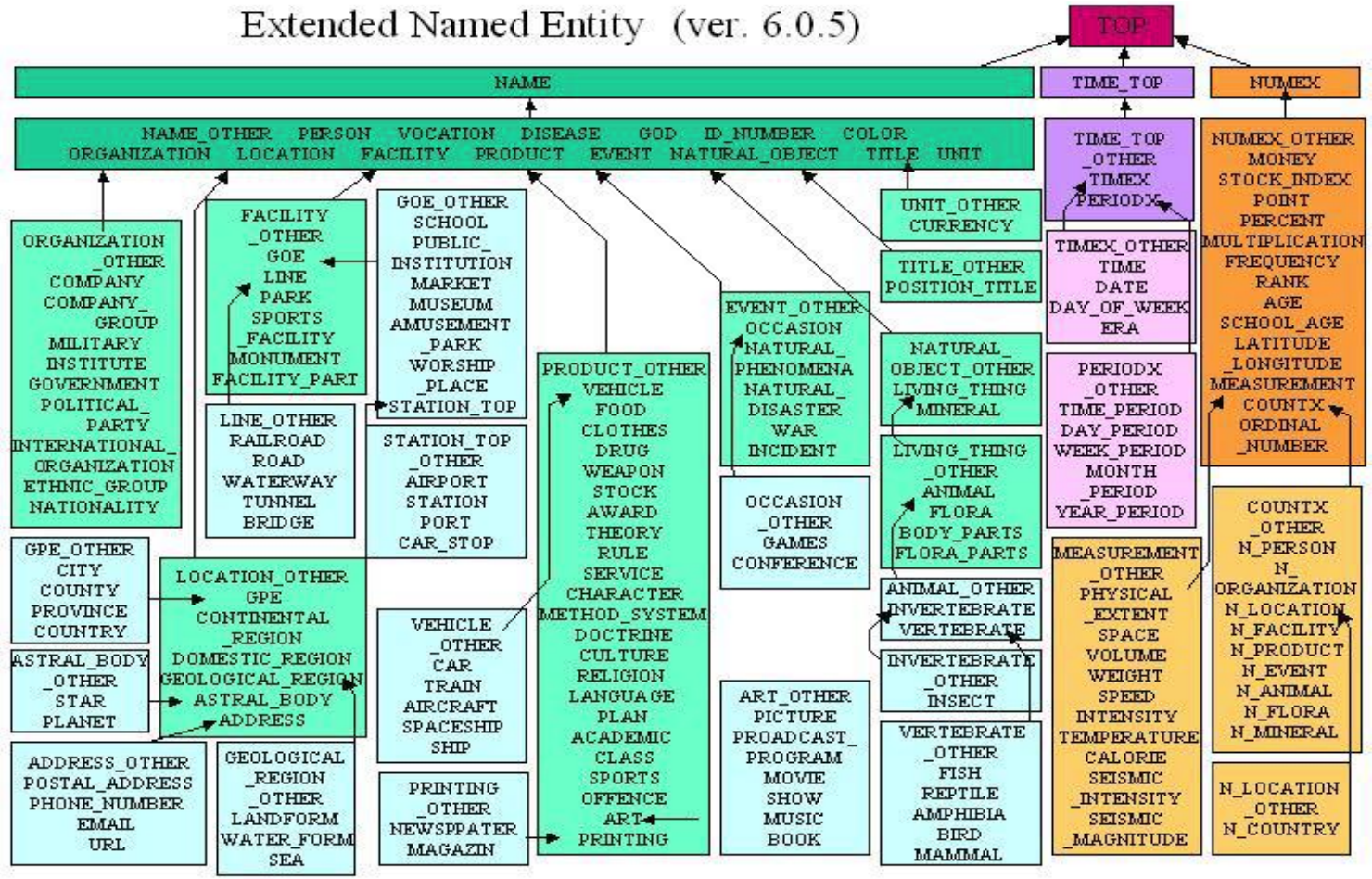
9.2.1 품사기반 카테고리 언어모델 - 개체명(Named Entity)

- 사람, 위치, 제품 등과 같이 적절한 이름으로 명명할 수 있는 실제 개체를 말함
  - 개체명은 개체의 인스턴스로 볼 수 있음
    - 예: 'New York City' 는 a 'city'의 인스턴스
  - 개체명인식(NER)에 활용
    - NER: 특정 유형의 개체 혹은 텍스트에서의 관계를 참조하는 구를 식별



9.2.1 품사기반 카테고리 언어모델 - 개체명(Named Entity)

- 개체명 세부 분류의 예제



(On-Demand Information Extraction and Linguistic Knowledge Acquisition, Satoshi Sekine, New York University)

9.2.1 품사기반 카테고리 언어모델

- 품사가 태깅된 학습 데이터가 필요함
- 단어가 여러 개의 카테고리를 가질 수 있기 때문에, 아래의 언어모델 확률을 계산할 때, $P(w_k|c_k)$는 현재 단어가 가질 수 있는 모든 카테고리에 대해서, $P(c_k|h_c)P(h_c|w_0 \dots w_{k-1})$는 가능한 모든 카테고리 히스토리에 대해서 합산을 해 주어야 함

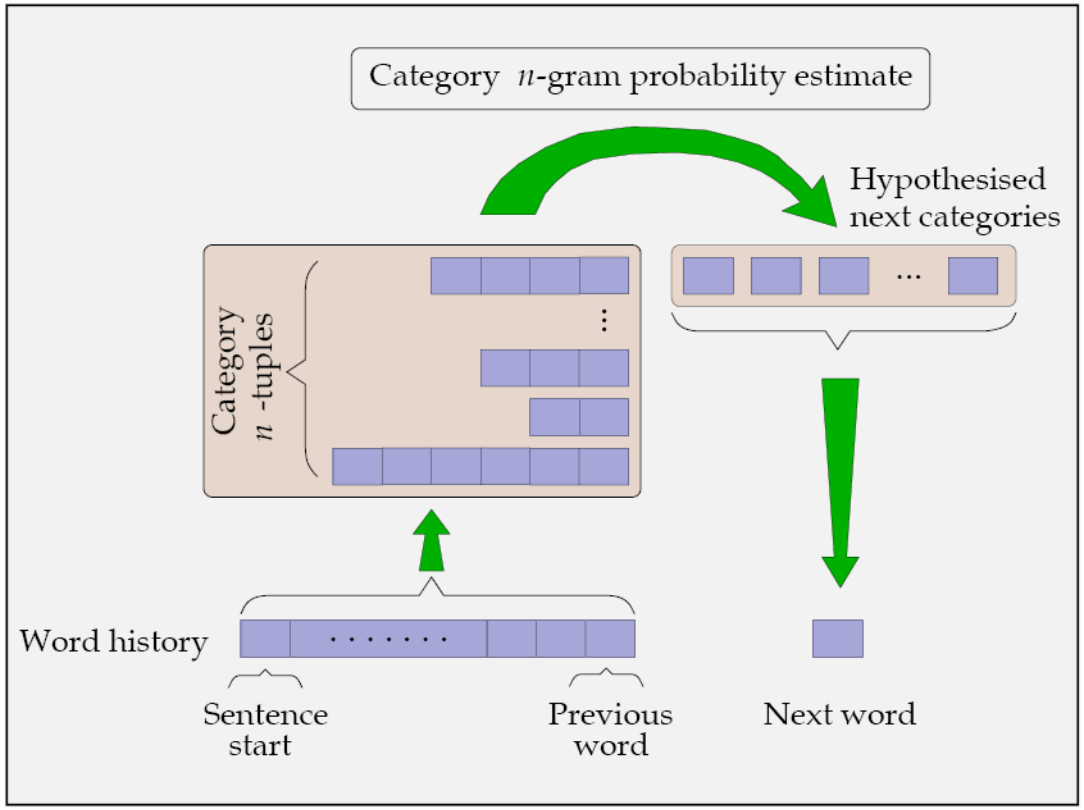
$$P(w_k|w_{k-2}, w_{k-1}) = \sum_{c_k \in C_k} P(w_k|c_k) \sum_{h_c \in H_{C_k}} P(c_k|h_c)P(h_c|w_0 \dots w_{k-1})$$

C_k: 현재단어가 가질 수 있는 카테고리의 집합

H_{C_k}: 가능한 모든 카테고리 히스토리의 집합

9.2.1 품사기반 카테고리 언어모델

- 카테고리기반 언어모델의 장점: 더 긴 히스토리를 볼 수 있음



Niesler, T. R., Whittaker, E. W., & Woodland, P. C. (1998, May). Comparison of part-of-speech and automatically derived category-based language models for speech recognition. In *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on* (Vol. 1, pp. 177-180). IEEE.

9.2.1 품사기반 카테고리 언어모델

- 하지만, 품사 등을 사람이 수동으로 태깅하고 카테고리를 지정하는 것은 노력이 매우 많이 필요함
  - 한 단어가 두개 이상의 품사를 가질 수 있음
  - 품사태깅 자체가 가지는 오류가 있음

- 이를 해결하기 위한 방법으로 자동화 알고리즘을 이용하여 단어를 그룹핑하는 방법이 있음
  - 단어당 하나의 카테고리만 매핑

9.2.2 단어 카테고리 자동생성

- 기본 알고리즘 (어휘 크기: |V|, 카테고리 수: K)
  1. Unigram 통계 생성 (단어 별 빈도수 측정)
  2. (초기화) 가장 빈도가 높은 단어를 카테고리1에 매핑. 그 다음 높은 빈도의 단어를 카테고리 2에 매핑. 같은 방법으로 K-1번째 카테고리 까지 매핑. (K-1개의 각 카테고리들은 1개의 단어만 매핑되어있음)
  3. (초기화) 나머지 단어들(총 |V|-(K-1))을 모두 K번째 카테고리에 매핑
  4. 현재의 단어-카테고리 매핑 상에서 모든 가능한 카테고리 이동에 대하여(총 |V|(K-1)개) 각 이동을 적용했을 때의 perplexity변화를 측정함
  5. 4의 과정에서 perplexity를 가장 낮추어 주는 이동을 선택하고 이를 적용한다
  6. Perplexity를 낮추는 이동이 없을 때 까지 4-5의 과정을 반복한다

9.2.2 단어 카테고리 자동생성

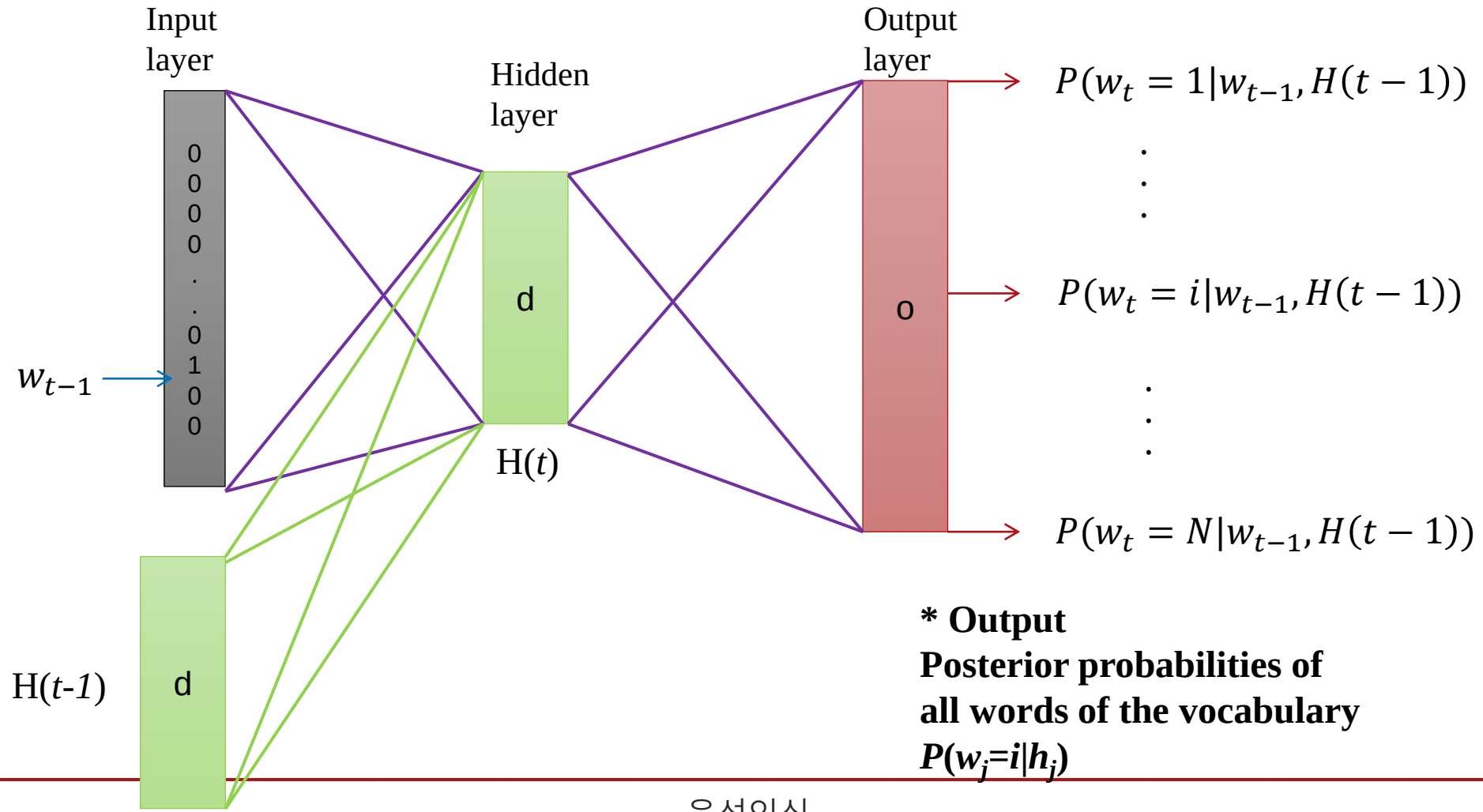
- 예: 20,000 단어에서 자동 생성한 100개의 카테고리 [Martin et. al.]
  - Class 2: THE, JAPAN'S, YESTERDAY'S, BRITAIN'S, TODAY'S, CANADA'S, CHINA'S, FRANCE'S, MEXICO'S …
  - Class 12: SAID, SAYS, ADDS, SUCCEEDS, CONTENDS, RECALLS, EXPLAINS, ASKS, PREDICTS, CONCEDES …
  - Class 22: BY, THEREBY
  - Class 32: PLANS, AGREED, EXPECTS, BEGAN, DID, MAKES, CAME, TOOK, GOT, DOES, CONTINUED, CALLS, HELPED …
  - Class 42: NEW, MAJOR, BIG, OLD, FULL, ADDITIONAL, SINGLE, NON, JOINT, LEADING, WIDE, DOUBLE, …
  - Class 52: U., JONES, BROTHERS, LYNCH, LEHMAN, STANLEY, HUTTON, SACHS, REYNOLDS, BACHE, PEABODY, …
  - Class 62: THAN, QUARTER, HALF, EIGHTHS, QUARTERS, EIGHTH, SIXTEENTHS, INTERSTATES
  - Class 72: BUSINESS, INTEREST, TAX, TRADE, DEBT, MONEY, CAPITAL, MANAGEMENT, WORK, CASH, GROWTH …
  - Class 82: INCORPORATED, CORPORATION, GROUP, UNIT, LIMITED, MAKER, INDUSTRIES, DIVISION, UNIVERSITY, …
  - Class 92: OFFICIALS, IT'S, ANALYSTS, TRADERS, EXECUTIVES, THAT'S, WE'RE, SOURCES, THERE'S, DEALERS …

9.3.1 DNN 기반 언어 모델

- DNN 기반으로 discrete 입력에 대해서, 대용량 자료에서 기존 n-gram보다 좋은 성능을 보이기 어려움
  - DNN 기반 언어모델의 구조(Feed-Forward 구조)
    - 가정 사항:
      - 어휘 사전: 약 65K
      - Trigram: (word history 2개 단어)
    - Input layer unit 수: 130K (2 * 65K) // tri-gram
    - Hidden layer: 2-5 hidden layers
    - Output layer unit 수: 65K
    - 1-of-N coding 형태의 discrete 입력
  - 음향 모델과의 비교 분석
    - 입력 차이 : 약 200배
    - 출력 차이 : 약 6배

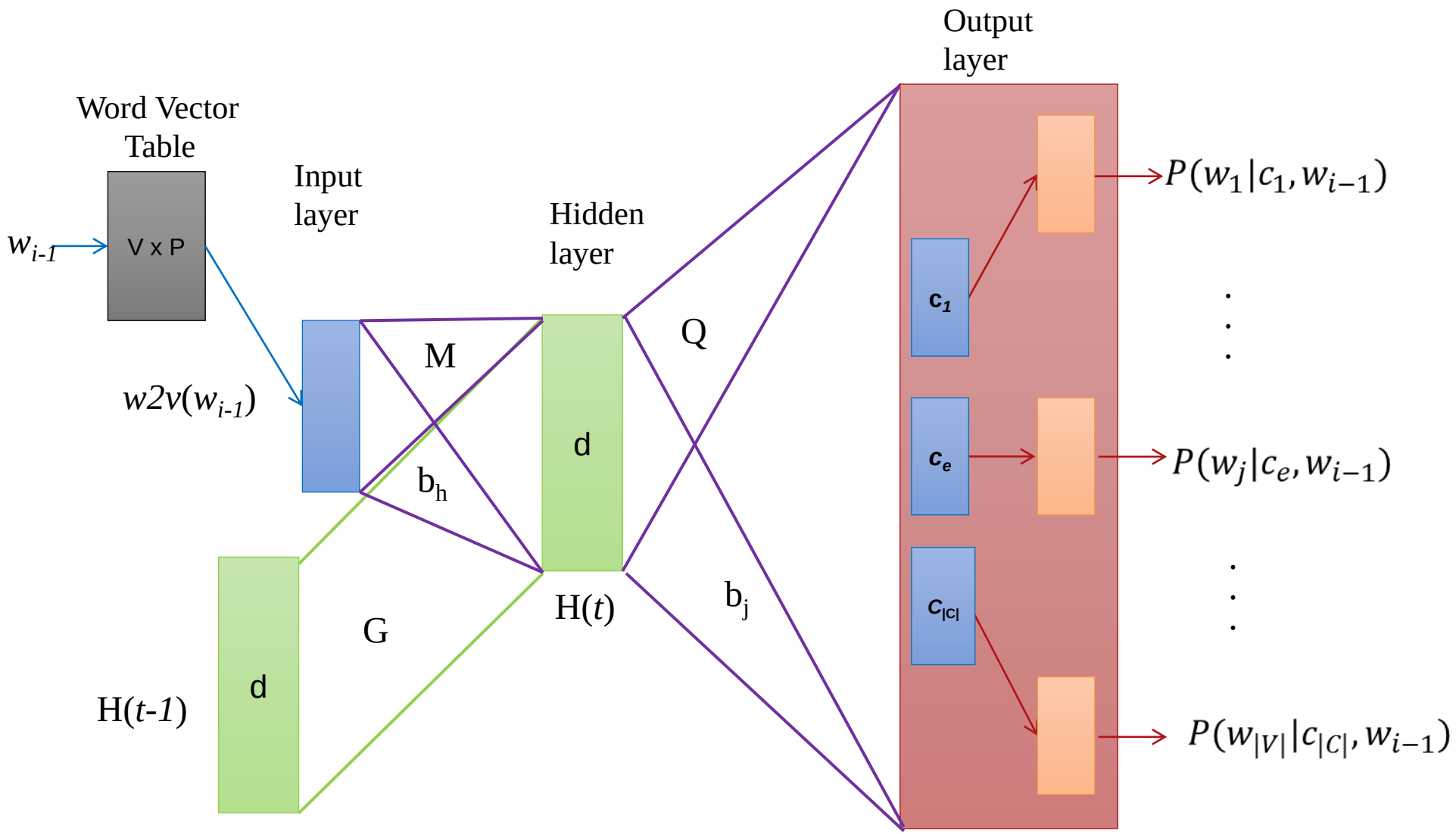
9.3.1 DNN 기반 언어 모델

- 대표적인 DNN 기반 언어모델 방법
  - Recurrent Neural Network (RNN) 기반 언어모델 [Mikolov 2013]



*** Output**
Posterior probabilities of all words of the vocabulary
***P*(w_j=*i*|h_j)**

9.3.1 DNN 기반 언어 모델



9.3.2 Word Vector를 이용한 입력 차원 감소

- Continuous word vector space
  - 사람들은 언어학적인 경험을 통해 단어간의 유사 관계를 앎
    - 다양한 문장을 미리 학습하여 생기는 경험적 단어 관계를 포함
  - 단어간에는 의미적 또는 문법적 유사 관계를 가지고 있으며, 이를 word간의 관계가 similarity를 통해 표현됨
  - Continuous word vector 모델 [P. Turney 2010]을 이용하고자 함

9.3.2 Word Vector를 이용한 입력 차원 감소

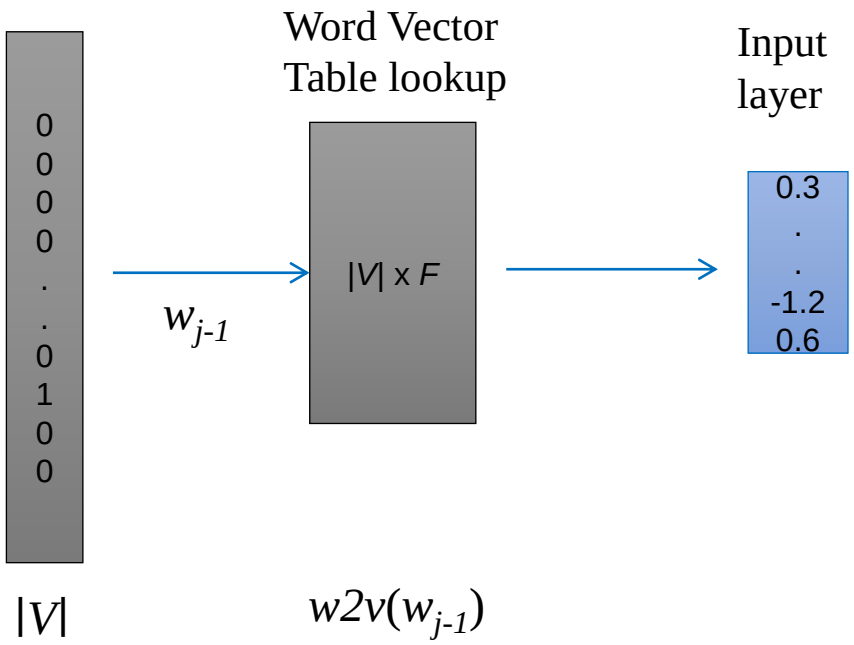
- Word Vector의 목표
  - Distributional Hypothesis
    - "Words which are similar in meaning occur in similar contexts." [H. Rubenstein 1965]
    - "Words with similar meanings will occur with similar neighbors if enough text material is available." [H. Schutze 1995]
    - 유사한 context에서의 단어들은 유사한 syntactic 또는 semantic 의미를 내포함
      - 유사한 context에서의 단어들은 interchangeable함
      - 예: The cat is walking in the bedroom.
        The dog was running in a room.
  - 즉, context가 유사한 단어들이 word vector space 상에서 서로 proximity가 높도록 표현해야 함

9.3.2 Word Vector를 이용한 입력 차원 감소

- Distributional Hypothesis를 만족하는 Word Vector의 장점
  - 앞 예제에서 ‘cat’과 ‘dog’는 유사한 역할을 담당
    - (‘cat’, ‘dog’), (‘is’, ‘was’), (‘walking’, ‘running’)
  - Semantic 또는 syntactic 관점에서 유사한 word들이 continuous vector space상에서 서로 가깝게 표현되면, 다양한 unseen word sequence를 포함하는 문장에 대해서도 generalization을 잘 표현하게 됨
    - 예:
      - The cat was walking in the bedroom. (is →was)
      - The cat is running in a room. (walking → running)

9.3.2 Word Vector를 이용한 입력 차원 감소

- 구글 Word2Vec을 이용하여 Distributional Hypothesis에 부합하는 word vector 생성
  - Word vector의 차원을 어휘 크기 (|V|) 에서 100 ~ 600 차원으로 표현



9.3.3 Hierarchical Softmax를 이용한 출력 차원감소

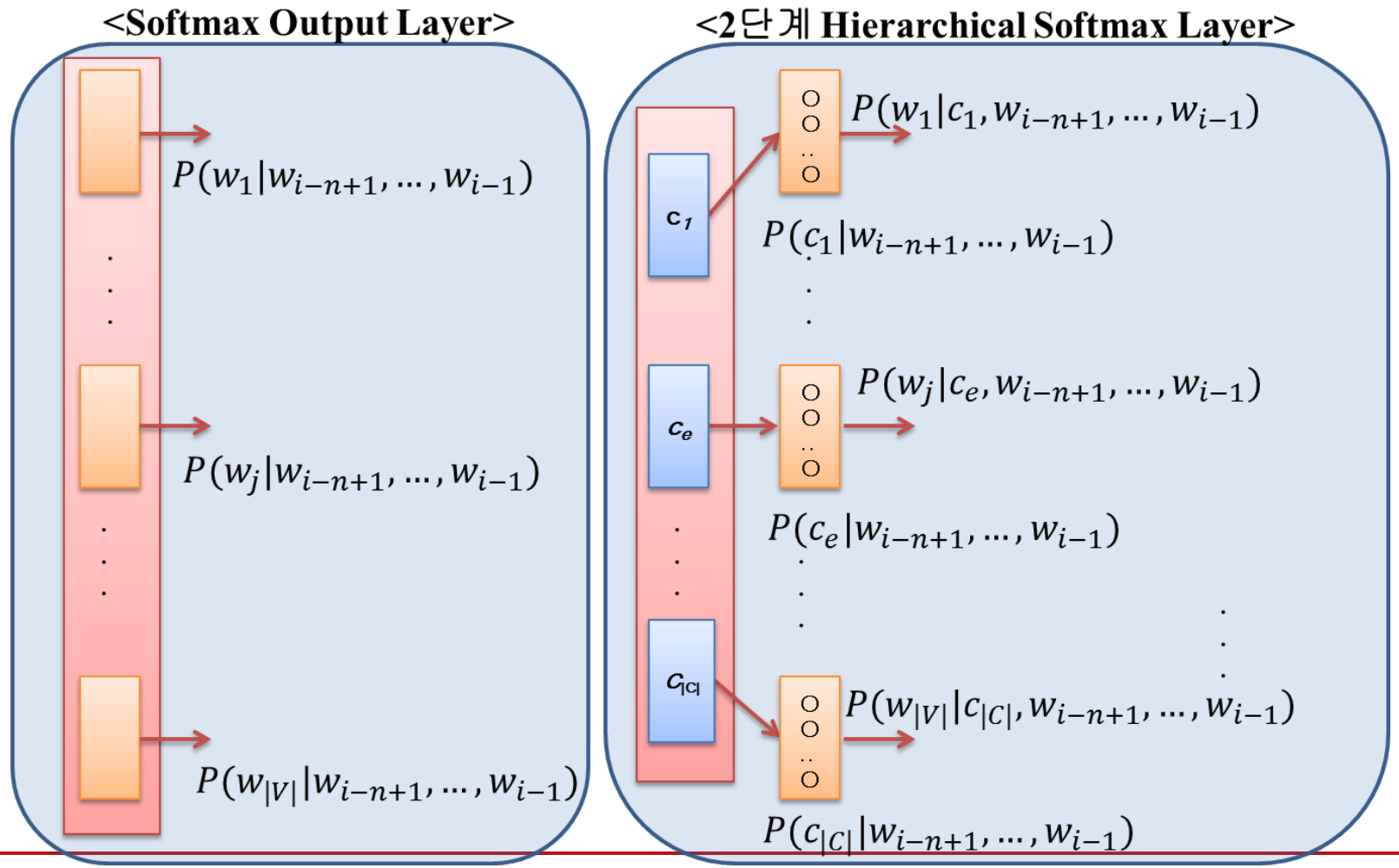
- Softmax 방법
  - Neural network 구조에서 output layer의 특정 output unit이 생성될 조건부 확률을 추정하는 경우
    - 언어모델링에서는 단어 히스토리가 주어진 경우에 현재 단어의 확률을 추정하는 조건부 확률을 계산
  - 모든 output unit의 값 대비 특정 output unit의 값의 비율로 측정하는 normalization 방법

$$p_j = \frac{\exp(o_j)}{\sum_{r=1}^{|V|} \exp(o_r)}$$

  - 문제점
    - 어휘 사전내의 모든 단어에 대해 조건부 확률을 구하는 방법으로,
      매 학습 자료에 대해, O(|V|) 연산량이 요구됨

9.3.3 Hierarchical Softmax를 이용한 출력 차원감소

- Hierarchical Softmax 방법



9.3.3 Hierarchical Softmax를 이용한 출력 차원감소

- 이슈 사항
  - 단어 카테고리의 개수는 어떻게 정하는가?
  - 단어 카테고리내에 포함되는 단어는 어떻게 결정되는가?

  - → Class-based 언어모델 연구에서 제안된 Auto-clustering 기법을 이용함

9.3.3 Hierarchical Softmax를 이용한 출력 차원감소

- 단어 카테고리내에 포함되는 단어 결정
  - 상향식 방식의 자동 클러스터링 방법인 Brown 단어 군집화 방법 [D. Jurafsky 2009][J. Turian 2010] 활용 (단, 군집화 개수는 미리 제공)
    - Brown 단어 군집화 알고리즘은 학습 말뭉치의 모든 단어로부터 생성한 단어 집합을 이진 트리 자료구조로 표현
    - 트리의 말단 노드는 단어를 나타내며, 동일 부모 노드를 가진 단어들은 하나의 단어 카테고리를 나타냄
    - 이진 트리 자료 구조에서 동일 부모 노드를 결정하는 목적식

$$loss = \frac{1}{N} log \prod_{i=1}^{N} P(w_i|c_i)P(c_i|c_{i-1})$$

      - 이 목적식을 최소화하는 단어 군집화 문제로 표현

    - WSJ 코퍼스를 이용하여 iteration 횟수에 따른 RNN 기반 언어모델 학습 결과, word class가 700일 때 가장 낮은 perplexity를 보임