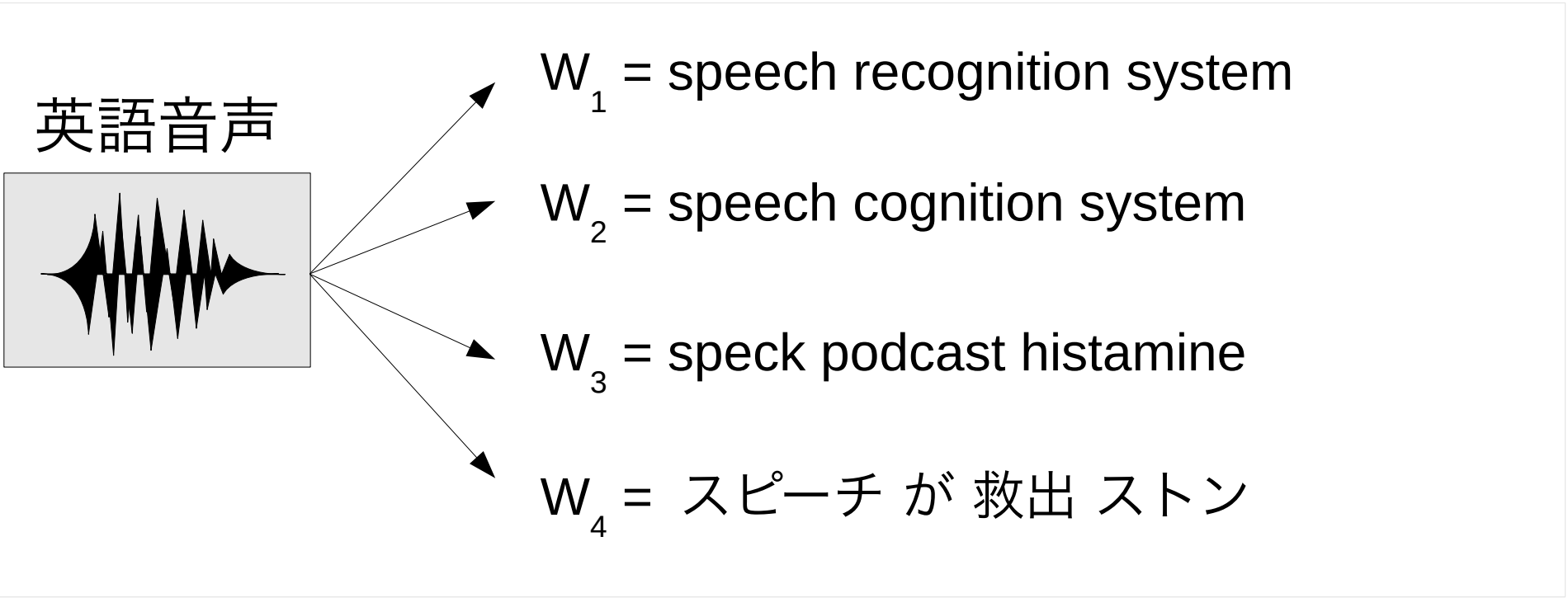
自然言語処理プログラミング勉強会 1 - 1-gram 言語モデル

Graham Neubig
奈良先端科学技術大学院大学 (NAIST)

言語モデルの基礎

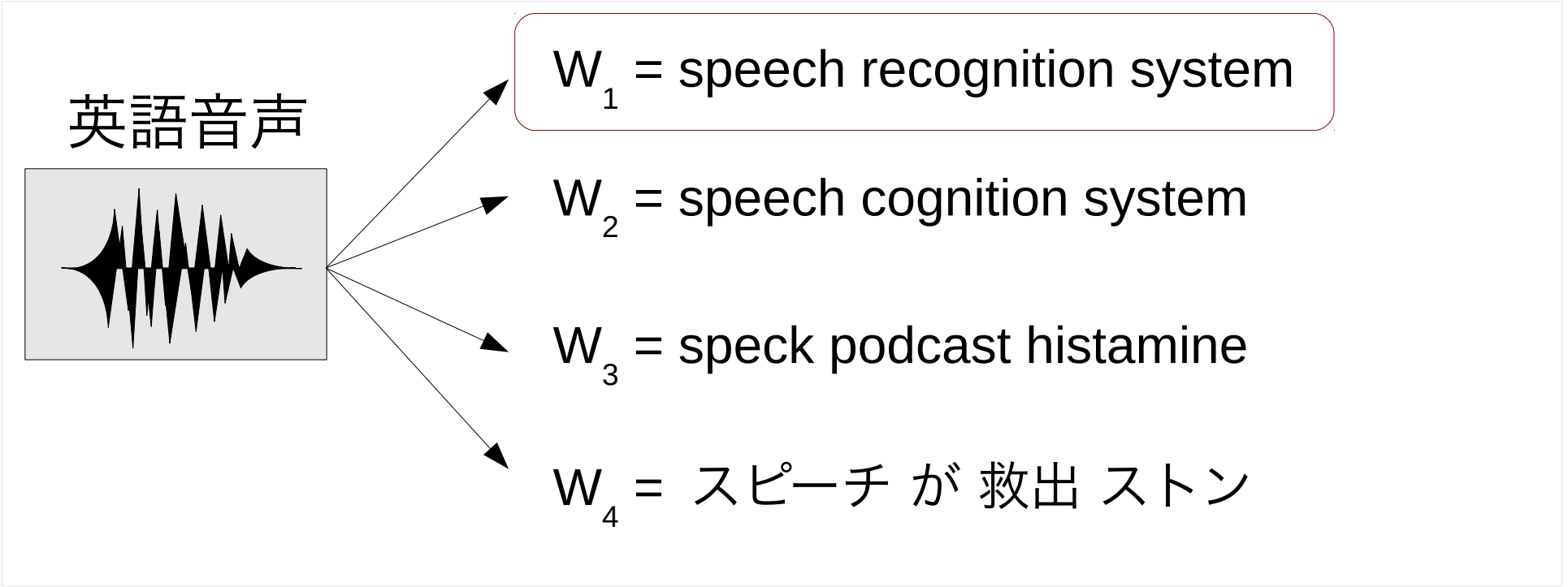
言語モデル？

- 英語の音声認識を行いたい時に、どれが正解？



言語モデル？

- 英語の音声認識を行いたい時に、どれが正解？



- 言語モデルは「もっともらしい」文を選んでくれる

確率的言語モデル

- 言語モデルが各文に確率を与える

W_1 = speech recognition system	$P(W_1) = 4.021 * 10^{-3}$
W_2 = speech cognition system	$P(W_2) = 8.932 * 10^{-4}$
W_3 = speck podcast histamine	$P(W_3) = 2.432 * 10^{-7}$
W_4 = スピーチ が 救出 ストン	$P(W_4) = 9.124 * 10^{-23}$

- $P(W_1) > P(W_2) > P(W_3) > P(W_4)$ が望ましい
  - (日本語の場合は $P(W_4) > P(W_1), P(W_2), P(W_3)$ ？)

文の確率計算

- 文の確率が欲しい

$$W = \text{speech recognition system}$$

- 変数で以下のように表す

$$P(|W| = 3, w_1 = \text{”speech”}, w_2 = \text{”recognition”}, w_3 = \text{”system”})$$

文の確率計算

- 文の確率が欲しい

W = speech recognition system

- 変数で以下のように表す (連鎖の法則を用いて):

$P(|W| = 3, w_1=\text{"speech"}, w_2=\text{"recognition"}, w_3=\text{"system"}) =$

$P(w_1=\text{"speech"} \mid w_0 = \text{"<s>"})$

$* P(w_2=\text{"recognition"} \mid w_0 = \text{"<s>"}, w_1=\text{"speech"})$

$* P(w_3=\text{"system"} \mid w_0 = \text{"<s>"}, w_1=\text{"speech"}, w_2=\text{"recognition"})$

$* P(w_4=\text{"</s>"} \mid w_0 = \text{"<s>"}, w_1=\text{"speech"}, w_2=\text{"recognition"}, w_3=\text{"system"})$

注：
文頭「 <s> 」と文末「 </s> 」記号

注：
$P(w_0 = \text{<s>}) = 1$

確率の漸次的な計算

- 前のスライドの積を以下のように一般化

$$P(W) = \prod_{i=1}^{|W|+1} P(w_i | w_0 \dots w_{i-1})$$

- 以下の条件付き確率の決め方は？

$$P(w_i | w_0 \dots w_{i-1})$$

最尤推定による確率計算

- コーパスの単語列を数え上げて割ることで計算

$$P(w_i|w_1 \ldots w_{i-1}) = \frac{c(w_1 \ldots w_i)}{c(w_1 \ldots w_{i-1})}$$

i live in osaka . </s>
i am a graduate student . </s>
my school is in nara . </s>

P(live | <s> i) = c(<s> i live)/c(<s> i) = 1 / 2 = 0.5

P(am | <s> i) = c(<s> i am)/c(<s> i) = 1 / 2 = 0.5

最尤推定の問題

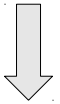
- 頻度の低い現象に弱い：

学習：

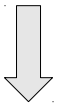
i live in osaka . </s>
i am a graduate student . </s>
my school is in nara . </s>

確率計算：

<s> i live in nara . </s>



P(nara|<s> i live in) = 0/1 = 0



P(W=<s> i live in nara . </s>) = 0

1-gram モデル

- 履歴を用いないことで低頻度の現象を減らす

$$P(w_i|w_1 \ldots w_{i-1}) \approx P(w_i) = \frac{c(w_i)}{\sum_{\tilde{w}} c(\tilde{w})}$$

i live in osaka . </s>
i am a graduate student . </s>
my school is in nara . </s>

P(nara) = 1/20 = 0.05
P(i) = 2/20 = 0.1
P(</s>) = 3/20 = 0.15

P(W=i live in nara . </s>) =
0.1 * 0.05 * 0.1 * 0.05 * 0.15 * 0.15 = $5.625 * 10^{-7}$

整数に注意！

- ２つの整数を割ると小数点以下が削られる

```
first_int = 1
second_int = 2

print(first_int/second_int)
```

```
$ ./my-program.py
0
```

- １つの整数を浮動小数点に変更すると問題ない

```
print(float(first_int)/second_int)
```

```
$ ./my-program.py
0.5
```

未知語の対応

- 未知語が含まれる場合は 1-gram でさえも問題あり

i live in osaka . </s>
i am a graduate student . </s> ⟶
my school is in nara . </s>

P(nara) = 1/20 = 0.05
P(i) = 2/20 = 0.1
P(kyoto) = 0/20 = 0

- 多くの場合（例：音声認識）、未知語が無視される
- 他の解決法
  - 少しの確率を未知語に割り当てる ($\lambda_{unk} = 1-\lambda_1$)
  - 未知語を含む語彙数を N とし、以下の式で確率計算

$$P(w_i) = \lambda_1 P_{ML}(w_i) + (1-\lambda_1)\frac{1}{N}$$

未知語の例

- 未知語を含む語彙数： $N=10^6$
- 未知語確率： $\lambda_{unk}=0.05$ ($\lambda_1 = 0.95$)

$$P(w_i) = \lambda_1 P_{ML}(w_i) + (1-\lambda_1)\frac{1}{N}$$

P(nara) = 0.95*0.05 + 0.05*(1/10^6) = 0.04750005

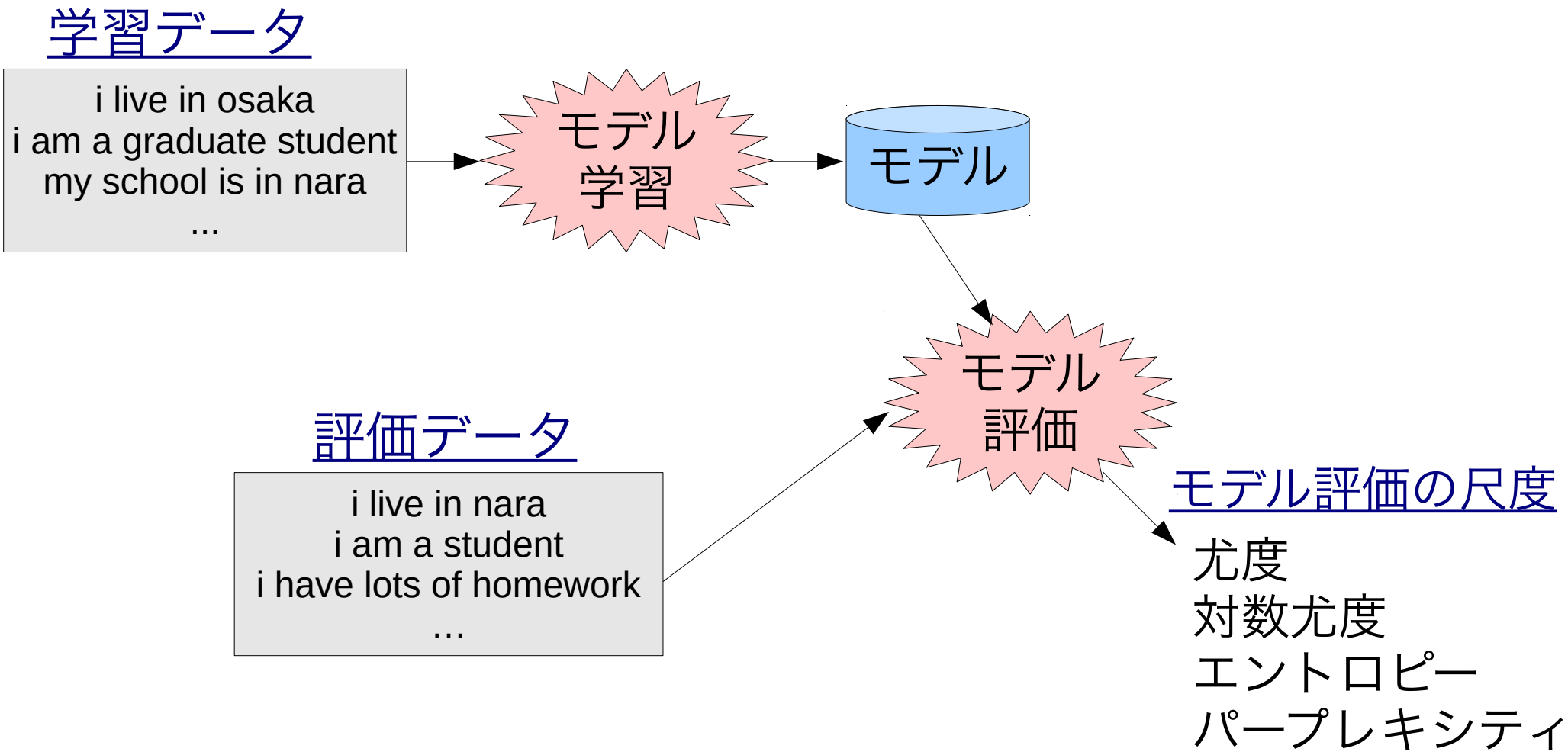
P(i) = 0.95*0.10 + 0.05*(1/10^6) = 0.09500005

P(kyoto) = 0.95*0.00 + 0.05*(1/10^6) = 0.00000005

言語モデルの評価

言語モデルの評価の実験設定

- 学習と評価のための別のデータを用意



尤度

- 尤度はモデル M が与えられた時の観測されたデータ (評価データ W_{test}) の確率

$$P(W_{test}|M) = \prod_{w \in W_{test}} P(w|M)$$

i live in nara i am a student my classes are hard	P(w="i live in nara"|M) =	$2.52*10^{-21}$
	x	
	P(w="i am a student"|M) =	$3.48*10^{-19}$
	x	
	P(w="my classes are hard"|M) =	$2.15*10^{-34}$
	=	
		$1.89*10^{-73}$

対数尤度

- 尤度の値が非常に小さく、桁あふれがしばしば起こる
- 尤度を対数に変更することで問題解決

$$\log P(W_{test}|M) = \sum_{w \in W_{test}} \log P(w|M)$$

i live in nara	log P(w="i live in nara"|M) =	-20.58
	+	
i am a student	log P(w="i am a student"|M) =	-18.45
	+	
my classes are hard	log P(w="my classes are hard"|M) =	-33.67
	=	
		-72.60

対数の計算

- Python の math パッケージで対数の log 関数

```
import math

print(math.log(100))      # ln(100)
print(math.log(100, 10)) # log10(100)
```

```
$ ./my-program.py
4.60517018599
2.0
```

エントロピー

- エントロピー H は負の底２の対数尤度を単語数で割った値

$$H(W_{test}|M) = \frac{1}{|W_{test}|} \sum_{w \in W_{test}} -\log_2 P(w|M)$$

i live in nara

i am a student

my classes are hard

$\log_2$ P(w="i live in nara"|M)= (68.43

+

$\log_2$ P(w="i am a student"|M)= 61.32

+

$\log_2$ P(w="my classes are hard"|M)= 111.84)

/

単語数： 12

=

20.13

* </s> を単語として数えることもあるが、ここでは入れていない

パープレキシティ

- ２のエントロピー乗

$$PPL = 2^{H}$$

- 一様分布の場合は、選択肢の数に当たる

$$V = 5 \quad H = -\log_2 \frac{1}{5} \quad PPL = 2^{H} = 2^{-\log_2 \frac{1}{5}} = 2^{\log_2 5} = 5$$

カバレージ

- 評価データに現れた単語（n-gram）の中で、モデルに含まれている割合

a bird a cat a dog a </s>

“dog” は未知語

カバレージ : 7/8 *

* 文末記号を除いた場合は → 6/7

演習問題

演習問題

- ２つのプログラムを作成
  - train-unigram: 1-gram モデルを学習
  - test-unigram: 1-gram モデルを読み込み、エントロピーとカバレージを計算
- テスト
  学習 test/01-train-input.txt →
  　正解 test/01-train-answer.txt
  テスト test/01-test-input.txt →
  　正解 test/01-test-answer.txt
- data/wiki-en-train.word でモデルを学習
- data/wiki-en-test.word に対してエントロピーとカバレージを計算

train-unigram 擬似コード

create a **map** *counts*
create a **variable** *total_count* = 0

for each *line* **in** the *training_file*
  split *line* into an array of *words*
  append “</s>” to the end of *words*
  for each *word* **in** *words*
    add 1 to *counts*[*word*]
    add 1 to *total_count*

open the *model_file* for writing
for each word, count **in** counts
  probability = *counts*[*word*]/*total_count*
  print *word*, *probability* **to** *model_file*

test-unigram 擬似コード

$\lambda_1 = 0.95$, $\lambda_{unk} = 1-\lambda_1$, V = 1000000, W = 0, H = 0

モデル読み込み

```
create a map probabilities
for each line in model_file
  split line into w and P
  set probabilities[w] = P
```

評価と結果表示

```
for each line in test_file
  split line into an array of words
  append "</s>" to the end of words
  for each w in words
    add 1 to W
    set P = λ_unk / V
    if probabilities[w] exists
      set P += λ_1 * probabilities[w]
    else
      add 1 to unk
    add -log_2 P to H

print "entropy = "+H/W
print "coverage = " + (W-unk)/W
```