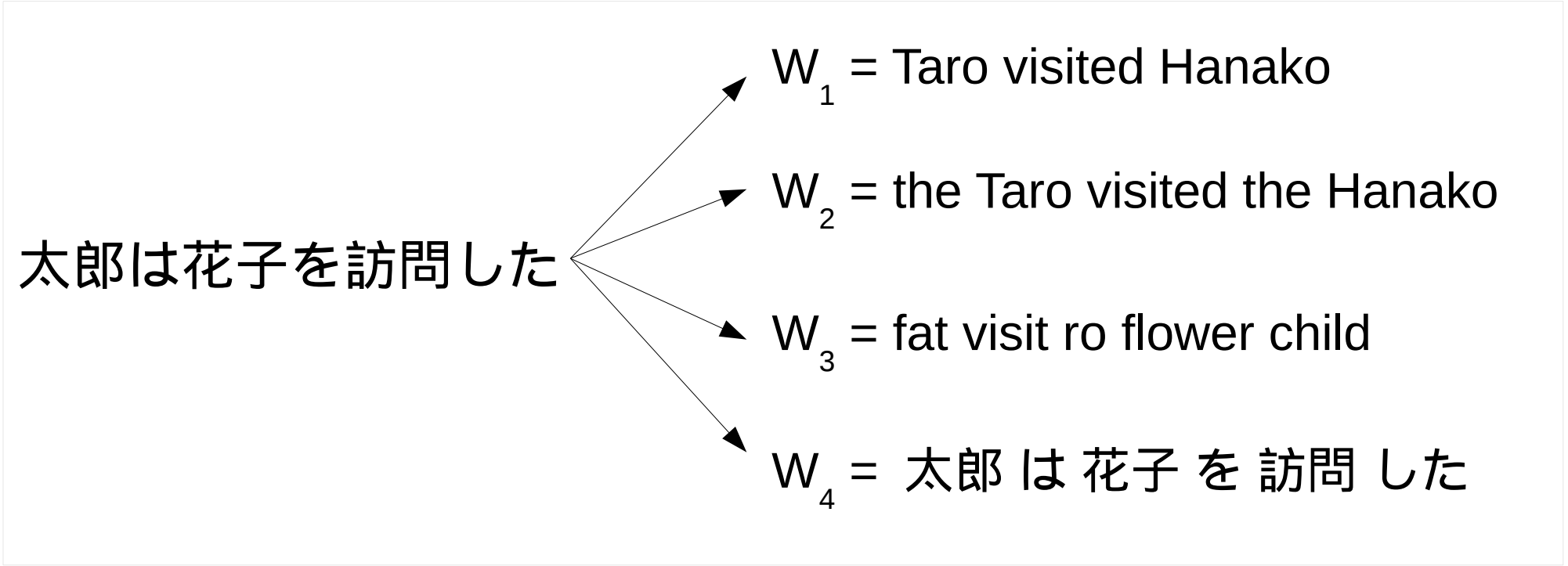
ALAGIN 機械翻訳セミナー
言語モデル

Graham Neubig
奈良先端科学技術大学院大学 (NAIST)
2014 年 3 月 4 日

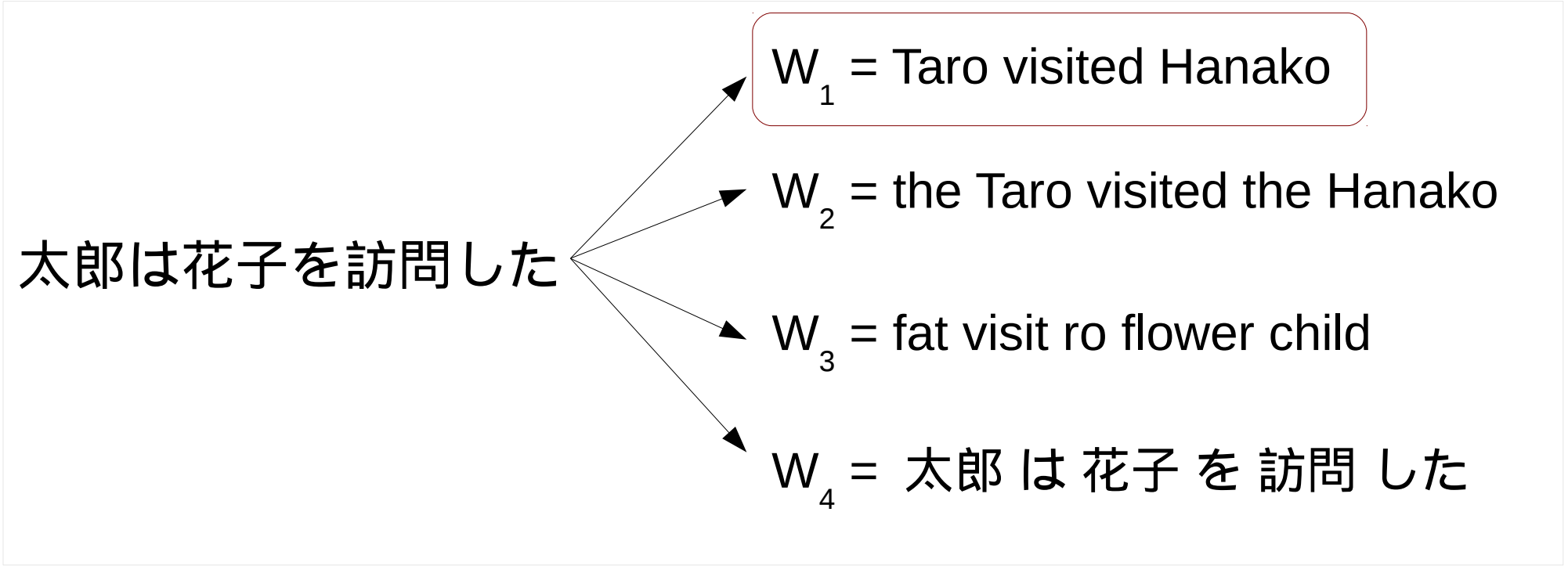
言語モデル？

- 日英翻訳を行いたい時に、どれが正解？



言語モデル？

- 日英翻訳を行いたい時に、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 = taro visited hanako	$P(W_1) = 4.021 * 10^{-3}$
W_2 = the taro visited the hanako	$P(W_2) = 8.932 * 10^{-4}$
W_3 = fat visit ro flower child	$P(W_3) = 2.432 * 10^{-7}$
W_4 = 太郎 は 花子 を 訪問 した	$P(W_4) = 9.124 * 10^{-23}$

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

確率のおさらい

確率の基本

- 確率分布
  変数「x」が値「a」を取る確率

P(x="a") P(y="b")	P(x) P(y)

- 同時確率
  変数「x」と「y」が同時に「a」と「b」を取る確率

P(x="a", y="b")	P(x, y)

- 条件付き確率
  変数「x」が「a」という情報が与えられた場合、変数「y」が「b」になる確率

P(y="b"|x="a")	P(y | x)

それぞれの確率の関係

- 同時確率と条件付き確率の関係

$$P(x, y) = P(y \mid x)\, P(x) \qquad P(y \mid x) = P(x, y) / P(x)$$

- 確率連鎖

$$P(x, y, z) = P(z \mid x, y)\, P(y \mid x)\, P(x)$$

- ベイズの定理（ベイズ則）

$$P(y \mid x) = P(x \mid y)\, P(y) / P(x)$$

言語モデル確率の計算

文の確率計算

- 文の確率が欲しい

W = taro visited hanako

- 変数で以下のように表す

$P(|W| = 3, w_1 = \text{”taro”}, w_2 = \text{”visited”}, w_3 = \text{”hanako”})$

文の確率計算

- 文の確率が欲しい

W = taro visited hanako

- 変数で以下のように表す (連鎖の法則を用いて):

$P(|W| = 3, w_1=\text{"taro"}, w_2=\text{"visited"}, w_3=\text{"hanako"}) =$

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

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

$* P(w_3=\text{"hanako"} \mid w_0 = \text{"<s>"}, w_1=\text{"taro"}, w_2=\text{"visited"})$

$* P(w_4=\text{"</s>"} \mid w_0 = \text{"<s>"}, w_1=\text{"taro"}, w_2=\text{"visited"}, w_3=\text{"hanako"})$

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

1-gram モデル

最尤推定の問題

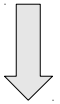
- 頻度の低い現象に弱い：

学習：

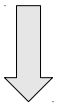
i live in osaka . </s>
i am a graduate student . </s>
my school is in nara . </s>

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 = $5.625 * 10^{-7}$

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

n-gram モデルと平滑化

1-gram モデルは語順を考慮しない

- 以下の確率は同等

P_{uni}(w=taro visited hanako) =
P(w=taro) * P(w=visited) * P(w=hanako) * P(w=</s>)

=

P_{uni}(w=hanako visited taro) =
P(w=taro) * P(w=visited) * P(w=hanako) * P(w=</s>)

1-gram モデルは単語の関係性を考慮しない

- 文法的な文：（名詞と活用が一致）

P_{uni}(w=i am) =
P(w=i) * P(w=am) * P(w=</s>)

P_{uni}(w=we are) =
P(w=we) * P(w=are) * P(w=</s>)

- 文法的でない文：（名詞と活用が矛盾）

P_{uni}(w=we am) =
P(w=we) * P(w=am) * P(w=</s>)

P_{uni}(w=i are) =
P(w=i) * P(w=are) * P(w=</s>)

しかし、確率は上記の文と同等

文脈を考慮することで解決！

- 1-gram モデルは文脈を考慮しない

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

- 2-gram は 1 単語の文脈を考慮

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

- 3-gram は 2 単語の文脈を考慮

$$P(w_i|w_0 \ldots w_{i-1}) \approx P(w_i|w_{i-2}w_{i-1})$$

- 4-gram 、 5-gram 、 6-gram などなど

n-gram 確率の最尤推定

- n 単語と n-1 単語からなる文字列の頻度を利用

$$P(w_i|w_{i-n+1}\cdots w_{i-1}) = \frac{c(w_{i-n+1}\cdots w_i)}{c(w_{i-n+1}\cdots w_{i-1})}$$

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

n=2 →
P(osaka | in) = c(in osaka)/c(in) = 1 / 2 = 0.5
P(nara | in) = c(in nara)/c(in) = 1 / 2 = 0.5

低頻度 n-gram の問題

- n-gram 頻度が 0→n-gram 確率も 0

P(osaka | in) = c(in osaka)/c(in) = 1 / 2 = 0.5
P(nara | in) = c(in nara)/c(in) = 1 / 2 = 0.5
P(school | in) = c(in school)/c(in) = 0 / 2 = **0**!!

- 1-gram モデルと同じく、線形補間を用いる

2-gram: $$P(w_i|w_{i-1}) = \lambda_2 P_{ML}(w_i|w_{i-1}) + (1-\lambda_2) P(w_i)$$

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

補間係数の選択法：グリッド探索

- λ_2 と λ_1 の様々な値を試し、尤度が最も高くなるように選択

$$\lambda_2 = 0.95, \lambda_1 = 0.95$$
$$\lambda_2 = 0.95, \lambda_1 = 0.90$$
$$\lambda_2 = 0.95, \lambda_1 = 0.85$$
$$\dots$$
$$\lambda_2 = 0.95, \lambda_1 = 0.05$$
$$\lambda_2 = 0.90, \lambda_1 = 0.95$$
$$\lambda_2 = 0.90, \lambda_1 = 0.90$$
$$\dots$$
$$\lambda_2 = 0.05, \lambda_1 = 0.10$$
$$\lambda_2 = 0.05, \lambda_1 = 0.05$$

問題：

選択肢が多すぎる
→ 選択に時間がかかる！

全ての n-gram に対して同じ λ
→ 尤度が最適とは限らない！

文脈を考慮した補間係数の選択

頻度の高い単語： Tokyo	頻度の低い単語： Tottori
c(Tokyo city) = 40 c(Tokyo is) = 35 c(Tokyo was) = 24 c(Tokyo tower) = 15 c(Tokyo port) = 10 …	c(Tottori is) = 2 c(Tottori city) = 1 c(Tottori was) = 0
ほとんどの 2-gram が既観測 → 大きなλが最適	未観測の 2-gram が多い → 小さなλが最適

- 補間係数の選択にも文脈を考慮：

$$P(w_i|w_{i-1}) = \lambda_{w_{i-1}} P_{ML}(w_i|w_{i-1}) + (1-\lambda_{w_{i-1}}) P(w_i)$$

Witten-Bell 平滑化

- $\lambda_{w_{i-1}}$ を選ぶ方法の１つ

$$\lambda_{w_{i-1}} = 1 - \frac{u(w_{i-1})}{u(w_{i-1}) + c(w_{i-1})}$$

$u(w_{i-1})$ = w_{i-1} の後に続く単語の異なり数

- 例えば、

c(Tottori is) = 2　c(Tottori city) = 1
c(Tottori) = 3　u(Tottori) = 2

$$\lambda_{Tottori} = 1 - \frac{2}{2+3} = 0.6$$

c(Tokyo city) = 40 c(Tokyo is) = 35 ...
c(Tokyo) = 270　u(Tokyo) = 30

$$\lambda_{Tokyo} = 1 - \frac{30}{30+270} = 0.9$$

絶対割引法

- 各頻度から少し (*d*) を引く

$$c'(w_{i-1}, w_i) = c(w_{i-1}, w_i) - d$$

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

- 例えば：

d=0.5
c(Tottori is) = 2
c(Tottori city) = 1
c'(Tottori is) = 1.5
c'(Tottori city) = 0.5
u(Tottori) = 2

$$P(w_i = \text{is} | w_{i-1} = \text{Tottori}) = \frac{1.5}{3} + \frac{2 * 0.5}{3} P(w_i = \text{is})$$

$$P(w_i = \text{city} | w_{i-1} = \text{Tottori}) = \frac{1.5}{3} + \frac{2 * 0.5}{3} P(w_i = \text{city})$$

Kneser-Ney 平滑化

- 機械翻訳で最も広く利用
- 絶対割引法と類似、$P(w_i)$のみを変更
- アイデア：
  - 平滑化された言語モデルでは 1-gram 分布は「2-gram 分布が信頼できない時」に利用
  - →1-gram は新しい文脈で現れやすい単語を重視すべき
  - 頻度の代わり、 1-gram を文脈の異なり数$x(w_i)$で計算

c(Barack Obama) = 50
c(President Obama) = 20

x(Obama) = 2　c(Obama) = 70

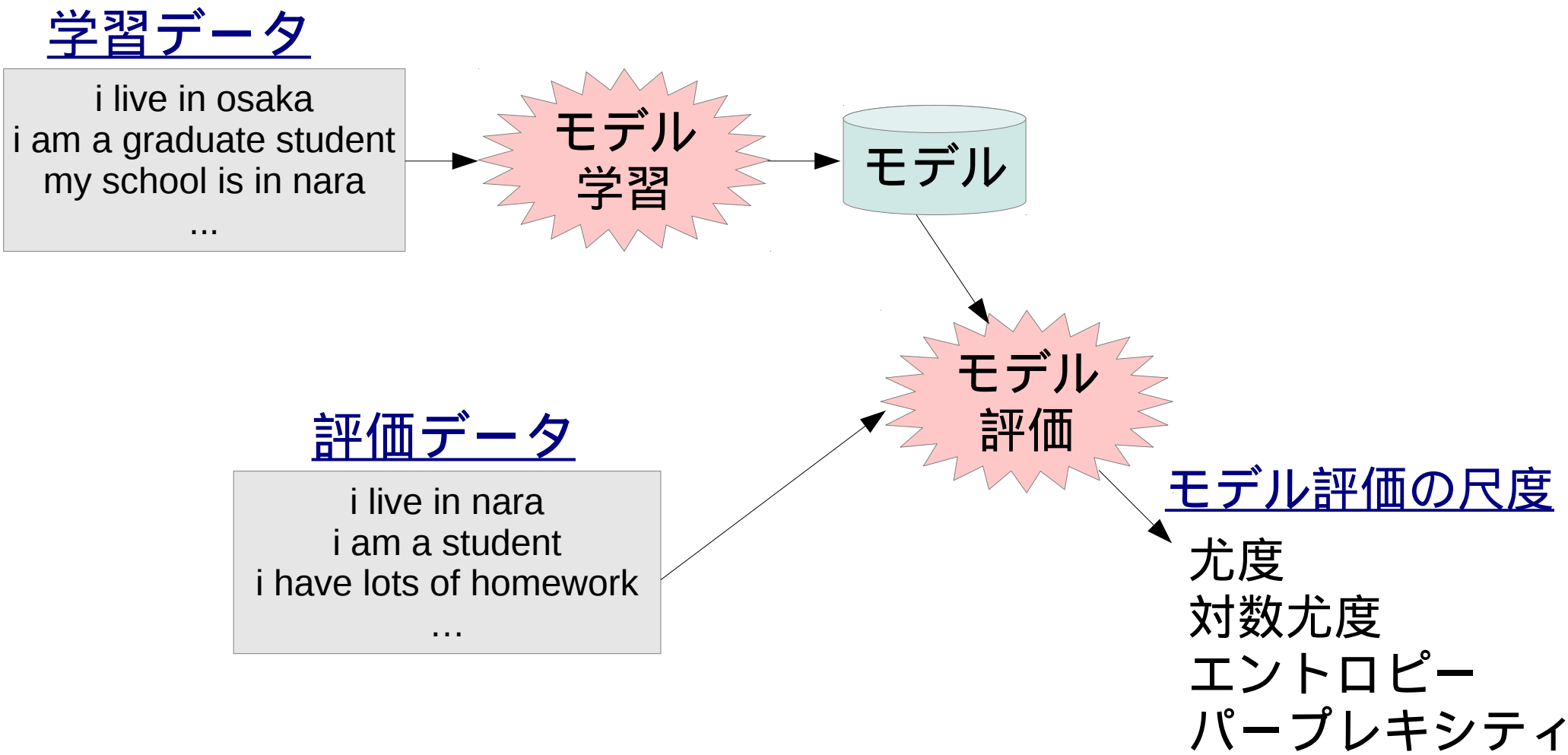
c(John Smith) = 7
c(Mary Smith) = 4
c(Fred Smith) = 3
...

x(Smith) = 20　c(Smith) = 50

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	P(w="i live in nara"|M) =	$2.52*10^{-21}$
		x
i am a student	P(w="i am a student"|M) =	$3.48*10^{-19}$
		x
my classes are hard	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 i am a student my classes are hard	log P(w="i live in nara"|M) = -20.58 + log P(w="i am a student"|M) = -18.45 + log P(w="my classes are hard"|M) = -33.67 = -72.60

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

(68.43 + 61.32 + 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

エントロピーと情報圧縮

- エントロピーHは与えられたデータを圧縮するのに必要なビット数でもある（シャノンの情報理論により）

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

a bird a cat a dog a </s>

Encoding

a	→ 1
bird	→ 000
cat	→ 001
dog	→ 010
</s>	→ 011

P(w= “a”) = 0.5 $-\log_2 0.5 = 1$

P(w= “bird”) = 0.125 $-\log_2 0.125 = 3$

P(w= “cat”) = 0.125 $-\log_2 0.125 = 3$

P(w= “dog”) = 0.125 $-\log_2 0.125 = 3$

P(w= “</s>”) = 0.125 $-\log_2 0.125 = 3$

10001001101011

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

言語モデルの発展

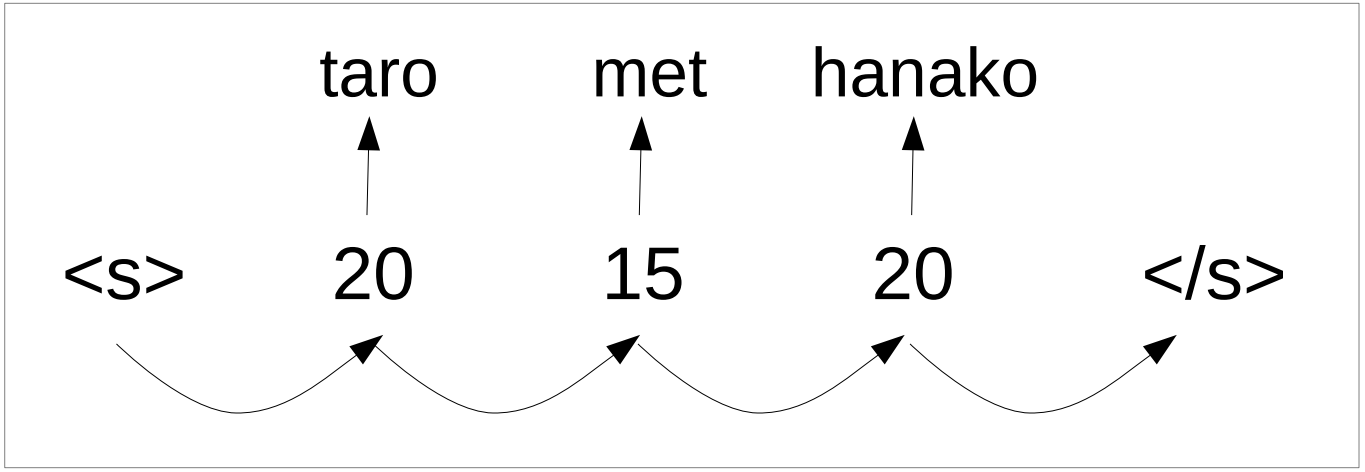
n-gram 言語モデル以外の手法

- およそ 90% の機械翻訳システムは単語 n-gram のみを利用
- その他の手法（だいたい人気の降順？）
  - クラス、品詞言語モデル
  - ニューラルネット言語モデル
  - 統語情報に基づく言語モデル
  - 識別言語モデル
- その他の考慮事項
  - データのサイズ
  - 言語モデルの補間

クラスに基づく言語モデル

- 各単語をクラスに割り当てる
- 単語クラスを推定してから単語を推定する

$$P(W)=\prod_{i=1}^{|W|+1} P(c_i|c_{i-1})\,P(w_i|c_i)$$



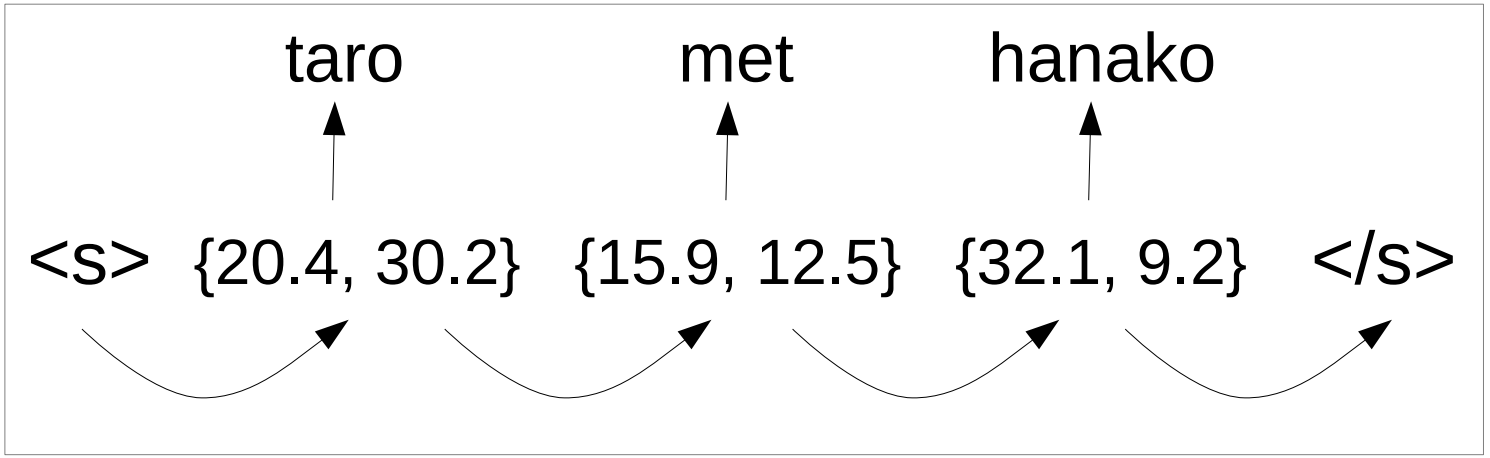
- 品詞もしくは自動的に獲得されたクラスを利用

単語のクラスタ化

- Brown クラスタ化 [Brown+ 92]
  - k 個のクラスを作りたい場合
  - 頻度の多い単語 k 個を別のクラスに
  - 残りの単語を頻度の昇順に処理し、言語モデル確率が一番高くなるクラスに追加
- 交換アルゴリズム [Martin+ 98]
  - クラスタを初期化（ランダム、 Brown などで）
  - 単語を 1 個ずつ処理し、任意のクラスタへ移動

ベクトル表現に基づく言語モデル

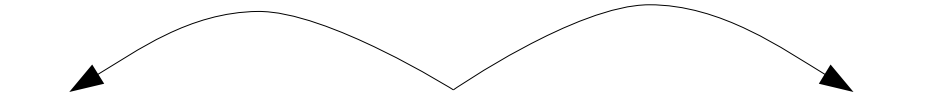
- クラスではなく、連続値のベクトルで表現



- ニューラルネットに基づく推定法が主流
  [Bengio 06, Mikolov+ 10, Lu+ 11, Niehues+ 12]
- 様々な軸における単語の類似性が考慮可能

統語情報に基づく言語モデル

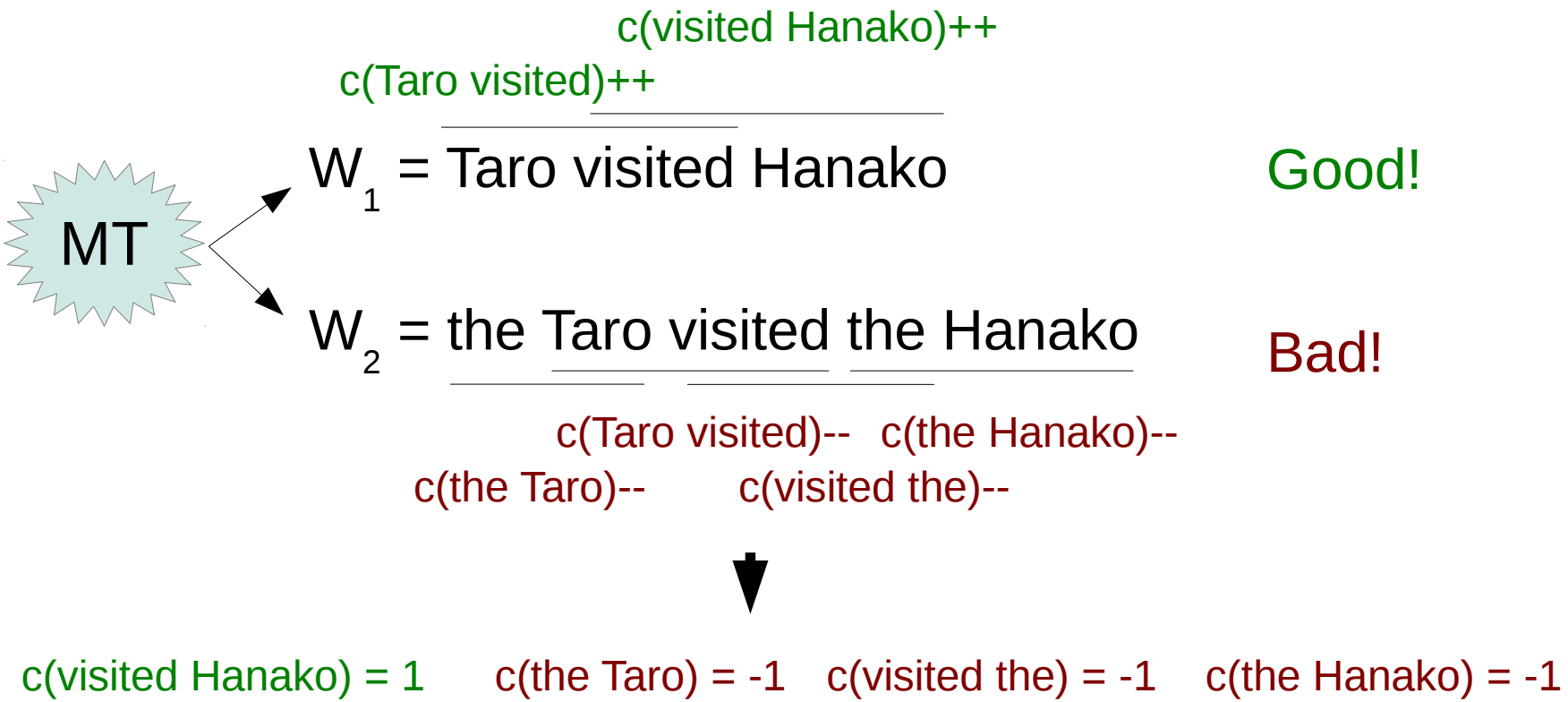
- 隣の単語だけではなく、係り受けも考慮 [Shen+ 08]



P(*met*) *
P(*taro* | *met*-as-head) *
P(*hanako* | *taro*, *met*-as-head)

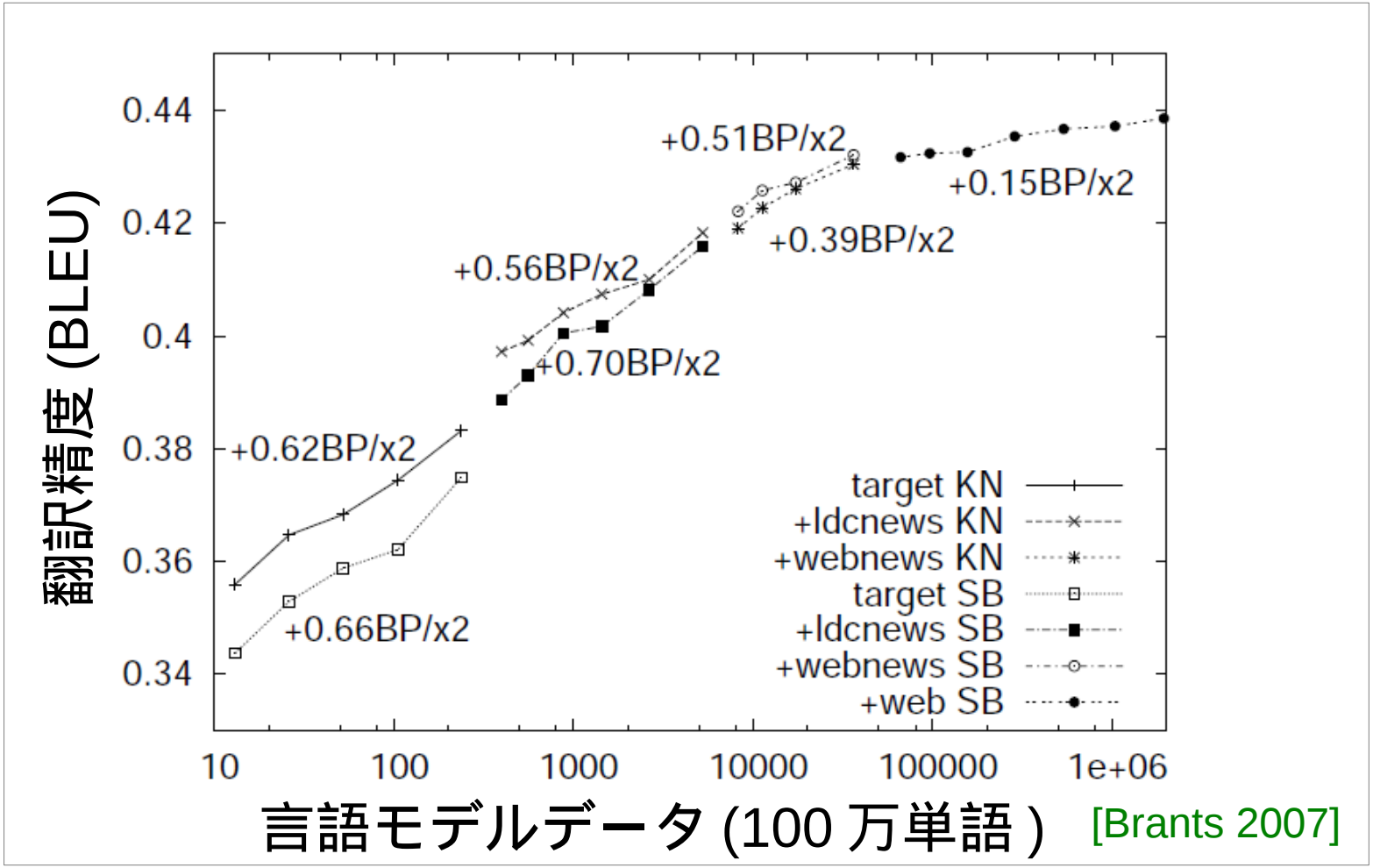
識別言語モデル

- 実際の翻訳結果を学習データに、「良い n-gram」と「悪い n-gram」を重み付きで学習



言語モデルにとって良いデータは

- 大きい →



[Brants 2007]

- ノイズを含まない
- 訳したいテキストと同じ分野

言語モデルの線形補間 [Jelinek+ 80]

- 用意できるデータとして
  - Web などの一般分野の大規模データ
  - 訳したい分野（社内文章、技術講演）の少量データ
- この場合、２つ以上の言語モデルを構築し、線形補間

$$P\left(w_i \middle| w_{i-n+1}^{i-1}\right) = \lambda_{gen} P_{gen}\left(w_i \middle| w_{i-n+1}^{i-1}\right) + \lambda_{trg} P_{trg}\left(w_i \middle| w_{i-n+1}^{i-1}\right)$$

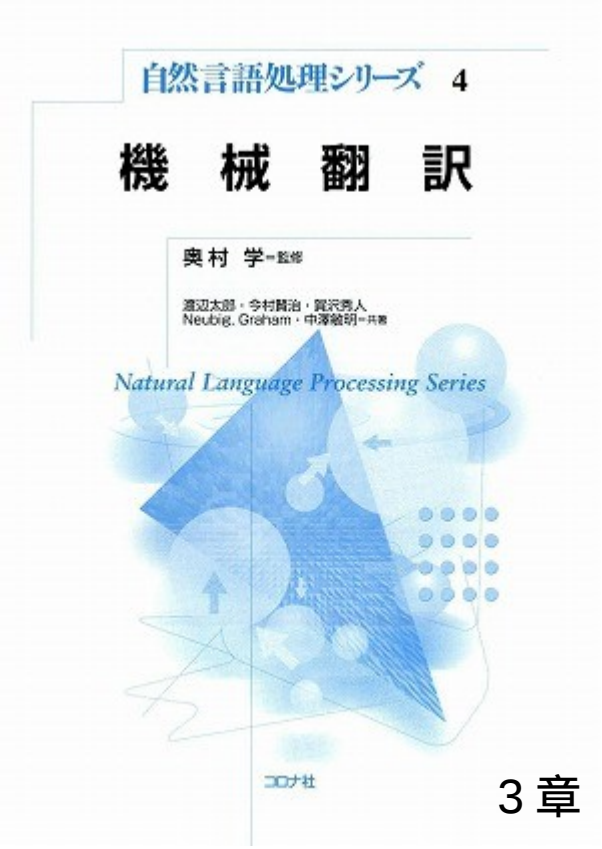
- 別のデータで尤度が高くなるように係数を決定

ツール・資料

(機械翻訳で広く使われる)
言語モデルツールキット

- KenLM:
  - 大規模な学習が可能であるが、オプションが少ない
  - 効率的な言語モデル格納が可能
  - オープンソース、研究、商用はともに無料
- SRILM:
  - 最も広く使われる言語モデル学習・格納ツールキット
  - オープンソース、研究利用は無料、商用利用は有料
- IRSTLM:
  - 他の言語モデル学習ツールキット
  - オープンソース、研究、商用はともに無料

更に勉強するには



3 章

- 自然言語処理プログラミングチュートリアル (1,2 回目)
  http://www.phontron.com/teaching.php
- “A bit of progress in language modeling”
  [Goodman 01]

参考文献

- [1] Y. Bengio, H. Schwenk, J.-S. Senecal, F. Morin, and J.-L. Gauvain. Neural probabilistic language models. In Innovations in Machine Learning,volume 194, pages 137-186. 2006.
- [2] T. Brants, A. C. Popat, P. Xu, F. J. Och, and J. Dean. Large language models in machine translation. In Proc. EMNLP, pages 858-867, 2007.
- [3] P. F. Brown, P. V. deSouza, R. L. Mercer, V. J. D. Pietra, and J. C.Lai. Class-based n-gram models of natural language. Comput. Linguist.,18(4):467-479, Dec. 1992.
- [4] S. Chen. Shrinking exponential language models. In Proc. NAACL,pages 468-476, 2009.
- [5] J. T. Goodman. A bit of progress in language modeling. Computer Speech & Language, 15(4), 2001.
- [6] F. Jelinek and R. L. Mercer. Interpolated estimation of Markov source parameters from sparse data. pages 381-397, 1980.
- [7] P. Koehn and J. Schroeder. Experiments in domain adaptation for statistical machine translation. In Proc. WMT, pages 224-227, 2007.
- [8] H.-S. Le, I. Oparin, A. Allauzen, J. Gauvain, and F. Yvon. Structured output layer neural network language model. In Proc. ICASSP, pages5524-5527, 2011.
- [9] S. Martin, J. Liermann, and H. Ney. Algorithms for bigram and trigramword clustering1. Speech Communication, 24(1):19-37, 1998.
- [10] T. Mikolov, M. Karafiat, L. Burget, J. Cernocky, and S. Khudanpur.Recurrent neural network based language model. In Proc. 11th InterSpeech, pages 1045-1048, 2010.
- [11] J. Niehues and A. Waibel. Continuous space language models usingrestricted boltzmann machines. In Proc. IWSLT, 2012.
- [12] L. Shen, J. Xu, and R.Weischedel. A new string-to-dependency machine translation algorithm with a target dependency language model. In Proc.ACL, pages 577-585, 2008.