ALAGIN 機械翻訳セミナー
単語アライメント

Graham Neubig
奈良先端科学技術大学院大学 (NAIST)
2014 年 3 月 5 日

https://sites.google.com/site/alaginmt2014/

統計的機械翻訳モデルの構築

- 各モデルを対訳文から学習

対訳文

太郎が花子を訪問した。
Taro visited Hanako.

花子にプレゼントを渡した。
He gave Hanako a present.

...

モデル

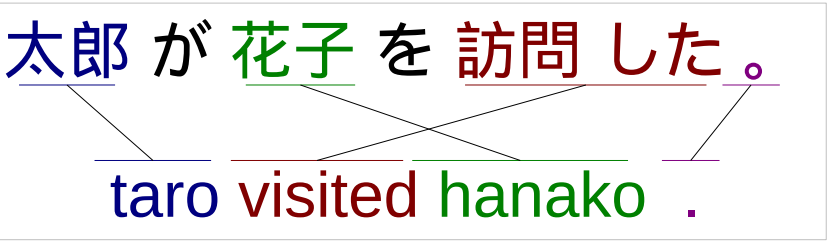
- 翻訳モデル
- 並べ替えモデル
- 言語モデル

単語アライメント

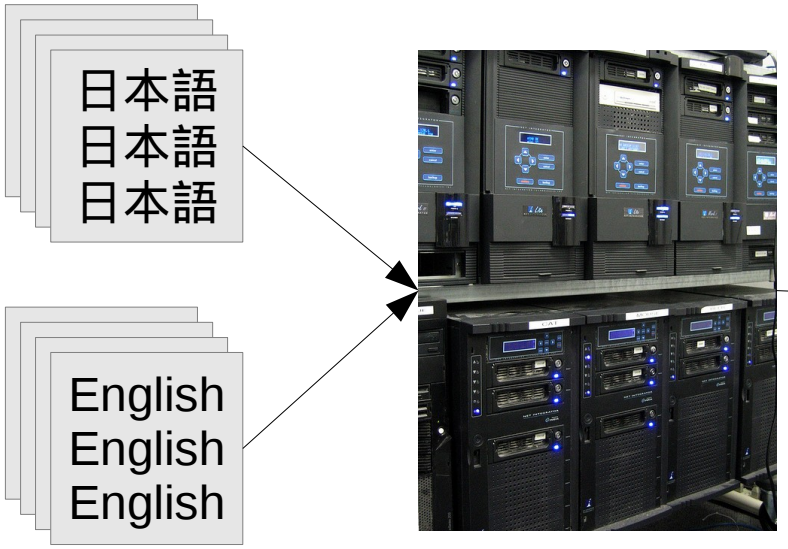
- 単語の対応を取ってくる技術

太郎 が 花子 を 訪問 した 。

taro visited hanako .



- 教師なしの確率モデルが最も広く利用されている



P(花子 |hanako) = 0.99
P(太郎 |taro) = 0.97
P(visited| 訪問) = 0.46
P(visited| した) = 0.04
P(花子 |taro) = 0.0001

ヒューリスティクスに基づくアライメント

アライメントの学習

- 例えば、日本語メニューのあるイタリア料理屋にて

チーズムース
Mousse di formaggi

タリアテッレ 4 種のチーズソース
Tagliatelle al 4 formaggi

本日の鮮魚
Pesce del giorno

鮮魚のソテー お米とグリーンピース添え
Filetto di pesce su “Risi e Bisi”

ドルチェとチーズ
Dolce e Formaggi

- 対応を見つけよう！

アライメントの学習

- 例えば、日本語メニューのあるイタリア料理屋にて

チーズムース
Mousse di formaggi

タリアテッレ 4 種のチーズソース
Tagliatelle al 4 formaggi

本日の鮮魚
Pesce del giorno

鮮魚のソテー お米とグリーンピース添え
Filetto di pesce su “Risi e Bisi”

ドルチェとチーズ
Dolce e Formaggi

- パターンを見つけよう！

共起頻度

- 対応の手がかりとして最も単純なのは共起

チーズ ムース
Mousse di formaggi

タリアテッレ 4 種 の チーズ ソース
Tagliatelle al 4 formaggi

本日 の 鮮魚
Pesce del giorno

鮮魚 の ソテー お 米 と グリーンピース 添え
Filetto di pesce su “Risi e Bisi”

ドルチェ と チーズ
Dolce e Formaggi

頻度

c(チーズ) = 3
c(の) = 3
c(と) = 2
…

c(formaggi) = 3
c(pesce) = 2
c(e) = 2
...

共起頻度

c(チーズ , formaggi) = 3
c(チーズ , mousse) = 1
c(チーズ , di) = 1
c(チーズ , tagliatelle) = 1
…

共起頻度の問題

- 共起頻度は頻度の高い単語に偏る

the banker met a tall man
銀行 員 が 背 の 高い 男 に 会った

a man ran out of the room
男 が 部屋 から 飛び出た

the young boy is good at soccer
あの 男 の 子 は サッカー が 上手 だ

the statue of liberty
自由 の 女神

he enjoys the olympics
彼 は オリンピック が 大好き だ

共起頻度

c(the, 男) = 3
c(man, 男) = 2

ダイス係数 [Dice 45]

- ダイス係数は頻度の高い単語にペナルティを与える

$$\mathrm{dice}(e, f) = \frac{2 * c(e, f)}{c(e) + c(f)}$$

the banker met a tall man
銀行 員 が 背 の 高い 男 に 会った

a man ran out of the room
男 が 部屋 から 飛び出た

the young boy is good at soccer
あの 男 の 子 は サッカー が 上手 だ

the statue of liberty
自由 の 女神

he enjoys the olympics
彼 は オリンピック が 大好き だ

ダイス係数

dice(the, 男) =
(2 * 3) / (5 + 3) = 0.75

dice(man, 男) =
(2 * 2) / (2 + 3) = 0.80

スコア→アライメント

- Now, we need a way to change dice coefficients to alignments

	historical	cold	outbreaks
歴代	0.596	0.018	0.250
の	0.002	0.003	0.000
風邪	0.020	0.909	0.037
大	0.007	0.002	0.085
流行	0.025	0.010	0.240

→

	historical	cold	outbreaks
歴代	●		
の	●		
風邪		●	
大			●
流行			●

最大スコア

- ある単語に対して、最もスコアの高い相手言語の単語を利用

	historical	cold	outbreaks	
歴代	0.596	0.018	0.250	→ historical
の	0.002	0.003	0.000	→ cold
風邪	0.020	0.909	0.037	→ cold
大	0.007	0.002	0.085	→ outbreaks
流行	0.025	0.010	0.240	→ outbreaks

閾値

- スコアが閾値を超える単語を利用

	historical	cold	outbreaks
歴代	0.596	0.018	0.250
の	0.002	0.003	0.000
風邪	0.020	0.909	0.037
大	0.007	0.002	0.085
流行	0.025	0.010	0.240

$t > 0.1$

競合リンク

- 最もスコアの高いアライメントを順に選択
  （１対１対応に限る）

	historical	cold	outbreaks
歴代	0.596	0.018	0.250
の	0.002	0.003	0.000
風邪	0.020	0.909	0.037
大	0.007	0.002	0.085
流行	0.025	0.010	0.240

1. 風邪 → cold
2. 歴代 → historical
3. 流行 → outbreaks

確率モデルによるアライメント：IBM モデル 1

確率モデルに基づくアライメント

- ２つの文の確率モデルを作成

F= チーズ ムース

E= mousse di formaggi

$$P(F \mid E; M)$$

- モデル M を確率的にパラメータ化

P(f= チーズ | e=formaggi) = 0.92
P(f= チーズ | e=di) = 0.001
P(f= チーズ | e=mousse) = 0.02
P(f= ムース | e=formaggi) = 0.07
P(f= ムース | e=di) = 0.002
P(f= ムース | e=mousse) = 0.89

- 確率により、洗練されたモデルが構築可能
- ほかのモデルと組み合わせやすい

IBM モデル 1 [Brown+ 93]

- F の各単語 f_j を以下の過程で生成
  - 単語インデックス a_j をランダムに生成 ($P(a_j) = 1/(|E|+1)$)、特別な NULL 単語を含む
  - 単語 f_j を $P(f \mid e_{aj})$ により生成

2 単語を生成:

チーズ

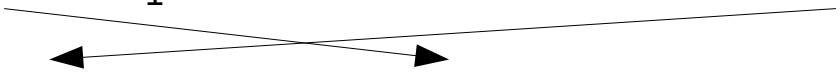
ムース

Choose: チーズ (P(f|e) = 0.92)

Choose: a_1 = 3 ($P(a_1=3)$ = 0.25)

Choose: ムース (P(f|e) = 0.89)

Choose: a_2 = 1 ($P(a_2=1)$ = 0.25)



mousse di formaggi NULL

IBM モデル 1 の式

- インデックスと単語の確率を計算すると：

$$P(F, A|E) = \prod_{j=1}^{J} \frac{1}{I+1} P(f_j|e_{a_j})$$

インデックス　単語

- 全てのアライメントに対して和を取ることも可能

$$P(F|E) = \sum_A \prod_{j=1}^{J} \frac{1}{I+1} P(f_j|e_{a_j})$$

$$= \prod_{j=1}^{J} \frac{1}{I+1} \sum_{i=1}^{I+1} P(f_j|e_i)$$

モデル１の学習

- モデルのパラメータを学習したい
- 尤度が最大になるように求める（最尤推定）

$$\hat{M} = \underset{M}{\operatorname{argmax}}\, P\left(F \middle| E\right)$$

- 最尤のパラメータをいかにして求めるのか？

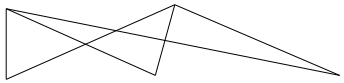
EM アルゴリズム

- モデルの尤度を最大化する標準的な手法：
  EM (Expectation-Maximization) アルゴリズム
- アイデア：
  - E ステップ：モデルに基づいて、 *e* が *f* へと翻訳される頻度を計算
  - M ステップ：計算された頻度に基づいてモデルのパラメータを更新
- 反復を何度も繰り返して、反復ごとにモデルの尤度が向上

EM の例

- 初期化：共起を数える

チーズ ムース

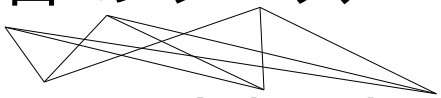


Mousse di formaggi

本日 の 鮮魚

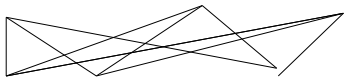
Pesce del giorno

本日 の チーズ



Formaggi del giorno

ドルチェ と チーズ

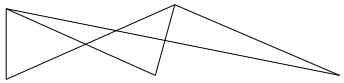


Dolce e Formaggi

EM の例

- M ステップ: パラメータを更新

チーズ ムース

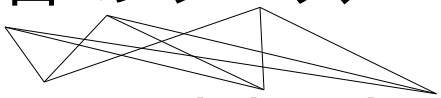


Mousse di formaggi

本日 の 鮮魚

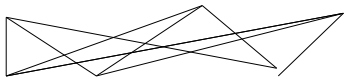
Pesce del giorno

本日 の チーズ



Formaggi del giorno

ドルチェ と チーズ



Dolce e Formaggi

P(チーズ |formaggi) = 0.375
P(ムース |formaggi) = 0.125
P(本日 |formaggi) = 0.125
P(の |formaggi) = 0.125
P(ドルチェ |formaggi) = 0.125
P(と |formaggi) = 0.125

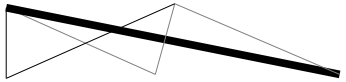
P(本日 |giorno) = 0.33
P(の |giorno) = 0.33
P(鮮魚 |giorno) = 0.16
P(チーズ |giorno) = 0.16

...

EM の例

- E ステップ： 単語の翻訳頻度を計算

チーズ ムース



Mousse di formaggi

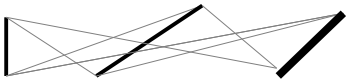
本日 の 鮮魚

Pesce del giorno

本日 の チーズ

Formaggi del giorno

ドルチェ と チーズ



Dolce e Formaggi

P(チーズ |formaggi) = 0.375
P(ムース |formaggi) = 0.125
P(本日 |formaggi) = 0.125
P(の |formaggi) = 0.125
P(ドルチェ |formaggi) = 0.125
P(と |formaggi) = 0.125

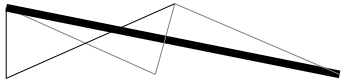
P(本日 |giorno) = 0.33
P(の |giorno) = 0.33
P(鮮魚 |giorno) = 0.16
P(チーズ |giorno) = 0.16

...

EM の例

- M ステップ: パラメータを更新

チーズ ムース



Mousse di formaggi

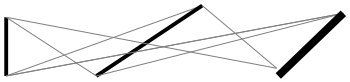
本日 の 鮮魚

Pesce del giorno

本日 の チーズ

Formaggi del giorno

ドルチェ と チーズ



Dolce e Formaggi

P(チーズ |formaggi) =	0.9
P(ムース |formaggi) =	0.02
P(本日 |formaggi) =	0.02
P(の |formaggi) =	0.02
P(ドルチェ |formaggi) =	0.02
P(と |formaggi) =	0.02
P(本日 |giorno) =	0.48
P(の |giorno) =	0.48
P(鮮魚 |giorno) =	0.02
P(チーズ |giorno) =	0.02

...

EM の例

- E ステップ: 単語の翻訳頻度を計算

P(チーズ |formaggi) = 0.9
P(ムース |formaggi) = 0.02
P(本日 |formaggi) = 0.02
P(の |formaggi) = 0.02
P(ドルチェ |formaggi) = 0.02
P(と |formaggi) = 0.02

P(本日 |giorno) = 0.48
P(の |giorno) = 0.48
P(鮮魚 |giorno) = 0.02
P(チーズ |giorno) = 0.02

...

初期化の式

- x と y がそれぞれ対応付けられる頻度の期待値を定義

$$q(e=x, f=y)$$

- 初期化では、共起頻度として初期化

$$q(e=x, f=y)=c(e=x, f=y)$$

M ステップの式

- モデルパラメータを更新
- 単純に、共起頻度を x の頻度で割る（最尤推定）

$$P(f=y|e=x)=\frac{q(e=x,f=y)}{q(e=x)}$$

where

$$q(e=x)=\sum_y q(e=x,f=y)$$

E ステップの式

- E ステップ : パラメータに基づいて頻度の期待値計算
  - ある文において a_j=i の確率は：

$$P(a_j=i|F,E,M)=\frac{1}{I+1}P(f_j|e_i) \;/\; \sum_{\tilde{i}=1}^{I+1}\frac{1}{I+1}P(f_j|e_{\tilde{i}})$$

現在の単語　　全ての単語

$$P(a_j=i|F,E,M)=P(f_j|e_i) \;/\; \sum_{\tilde{i}=1}^{I+1}P(f_j|e_{\tilde{i}})$$

  - 全ての文を考慮すると期待値を以下のように計算
    (δ = クロネッカーのδ、真の場合は 1、偽の場合は 0)

$$q(e=x,f=y)=\sum_{E,F}\sum_{i=1}^{I+1}\sum_{j=1}^{J}P(a_j=i|F,E,M)\delta(e_i=x,f_j=y)$$

対応の求め方

- 学習後、翻訳確率が最も高い単語を用いる：

$$\hat{a}_j = \underset{a_j}{\operatorname{argmax}} P\left(a_j | F, E, M\right)$$

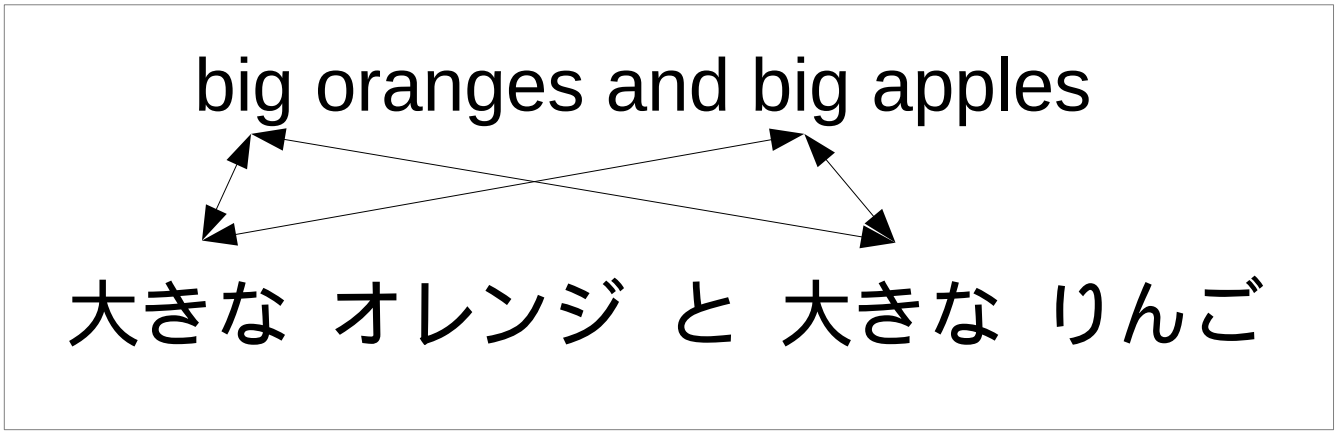
F

E	historical	cold	outbreaks	NULL	
歴代	0.596	0.018	0.250	0.001	→ historical
の	0.002	0.003	0.000	0.010	→ NULL
風邪	0.020	0.909	0.037	0.000	→ cold
大	0.007	0.002	0.085	0.005	→ outbreaks
流行	0.025	0.010	0.240	0.001	→ outbreaks

確率モデルによるアライメント：Model 2-5, HMM

モデル１の問題

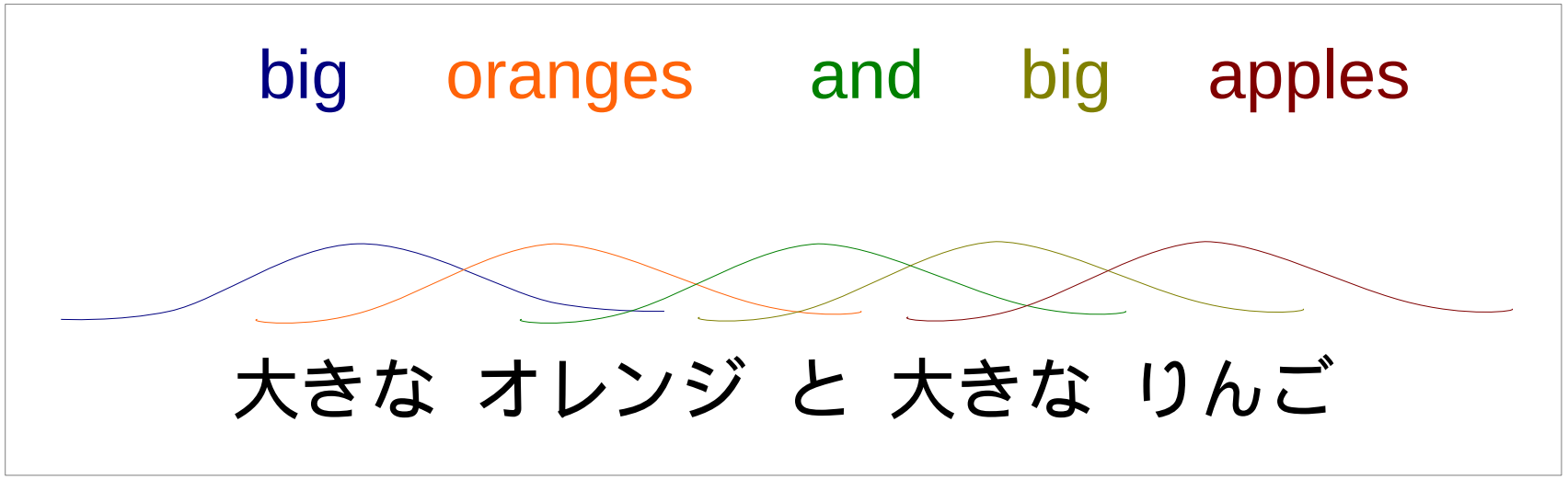
- 単語の順番を全く気にしない



- この問題に対応するために多くのモデルが提案

モデル２のアイデア

- 両言語の単語はだいたい同じ語順でしょう



モデル１→モデル２

- モデル１

$$P(F, A|E) = \prod_{j=1}^{J} \frac{1}{I+1} P(f_j|e_{a_j})$$

インデックス　単語

- モデル２

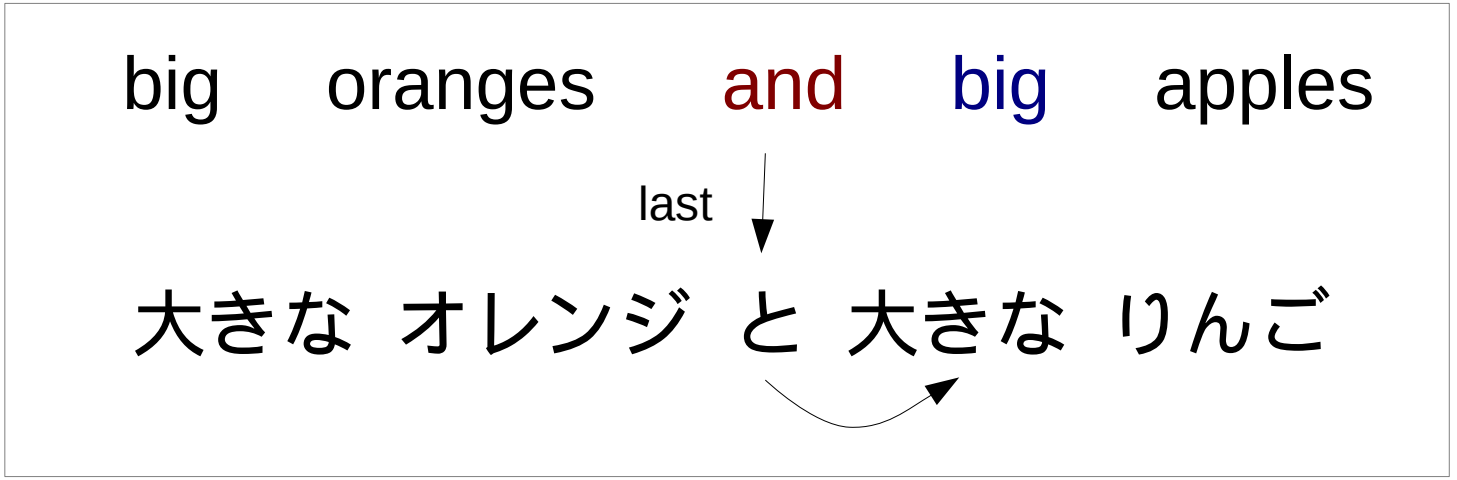
$$P(F, A|E) = \prod_{j=1}^{J} P(a_j|j) P(f_j|e_{a_j})$$

インデックス　単語

- モデル１と同じ効率的な学習が可能

隠れマルコフモデル (HMM) に基づくアライメント [Vogel+ 96]

- f_j に対応する単語は f_{j-1} に対応する単語に近いことが多い



- 語順が大きく変わる言語でも局所的に成り立つ

モデル 1 → HMM

- モデル 1

$$P(F, A|E) = \prod_{j=1}^{J} \frac{1}{I+1} P(f_j|e_{a_j})$$

インデックス　単語

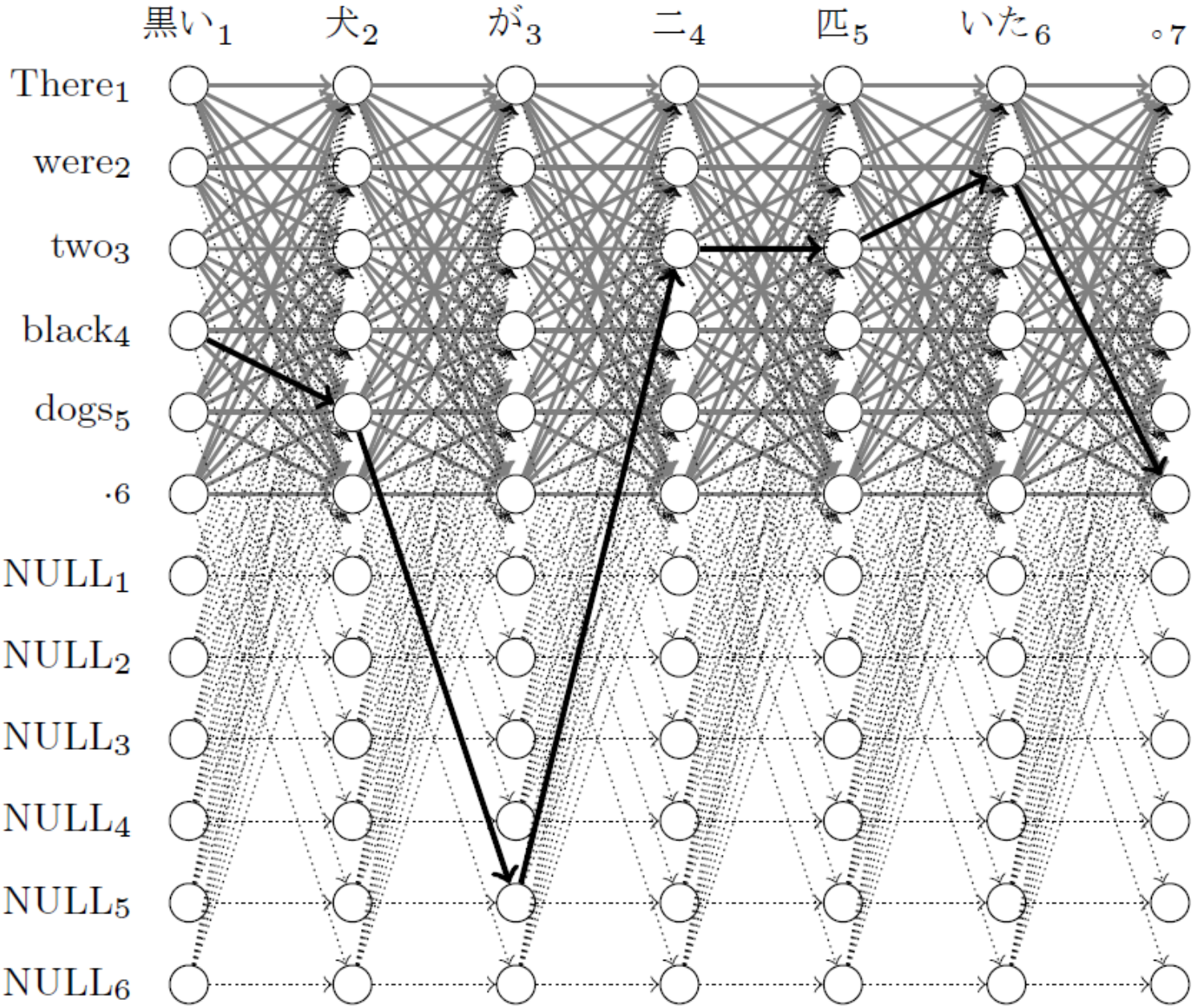
- モデル 2

$$P(F, A|E) = \prod_{j=1}^{J} P(a_j|a_{j-1}) P(f_j|e_{a_j})$$

インデックス　単語

- HMM で広く使われる前向き後ろ向きアルゴリズムで学習可能

HMM Graph



「機械翻訳」より

IBM モデル 3-5

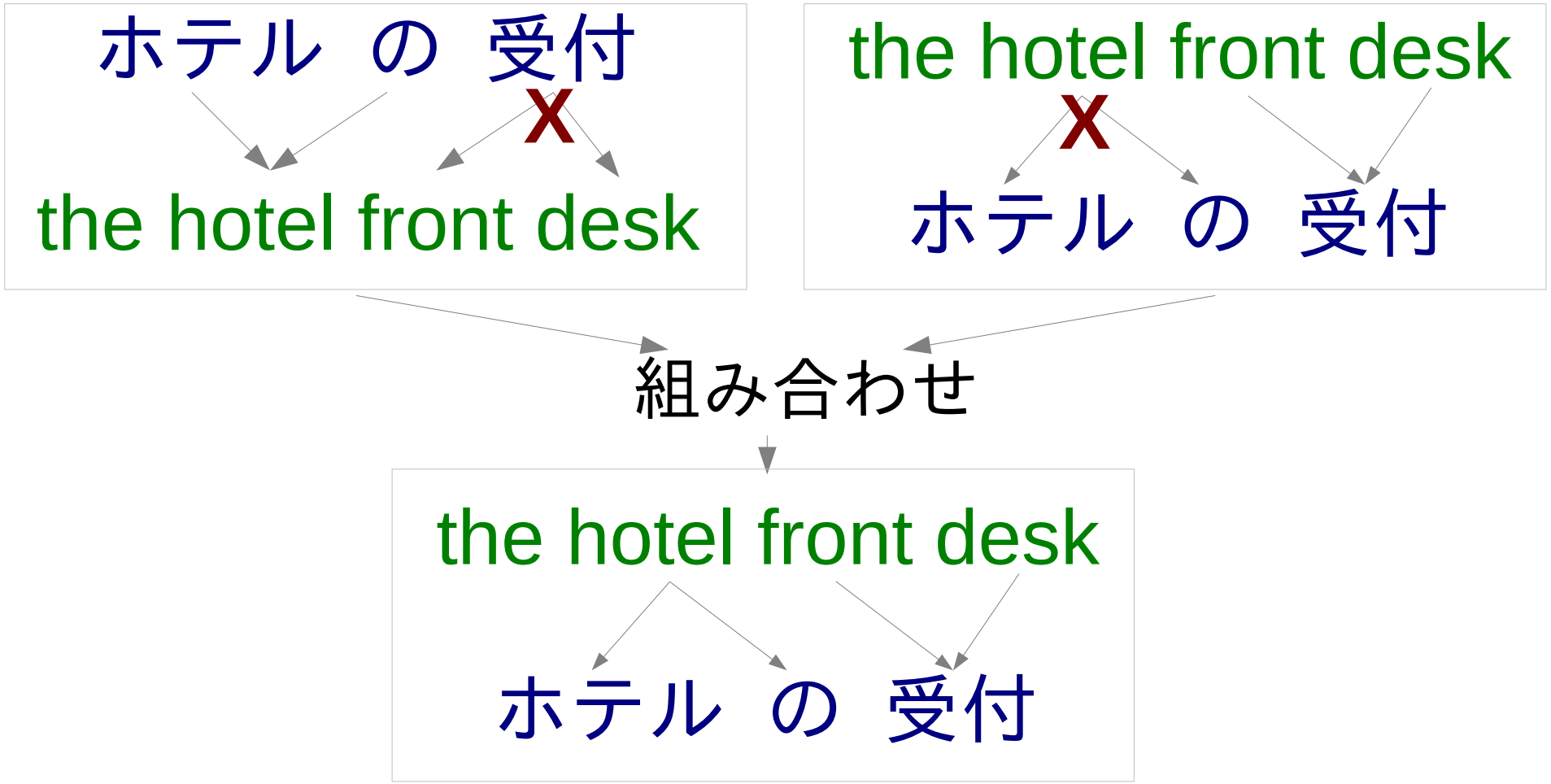
- 「稔性」という、１単語が何単語に対応するかを考慮

Fertility 1	Fertility 3.5
I	adopted
私 僕 俺	採用 され た 養子 に なっ た

- モデル化、学習、対応付けが全体的に複雑で、近似が必要

アライメントの組み合わせ

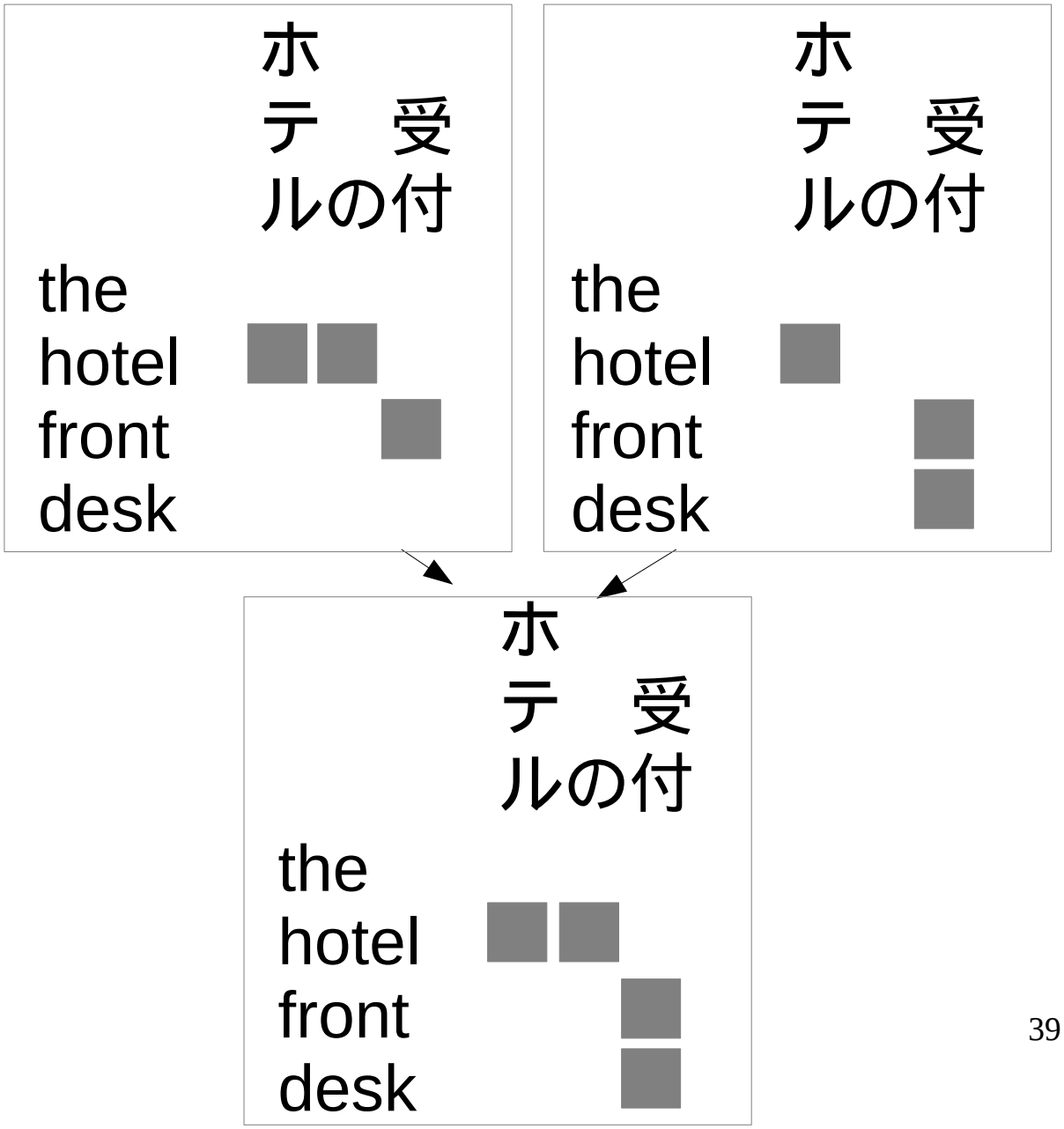
一対多アライメントの組み合わせ [Koehn+ 03]



- 主にヒューリスティクスによって行われる

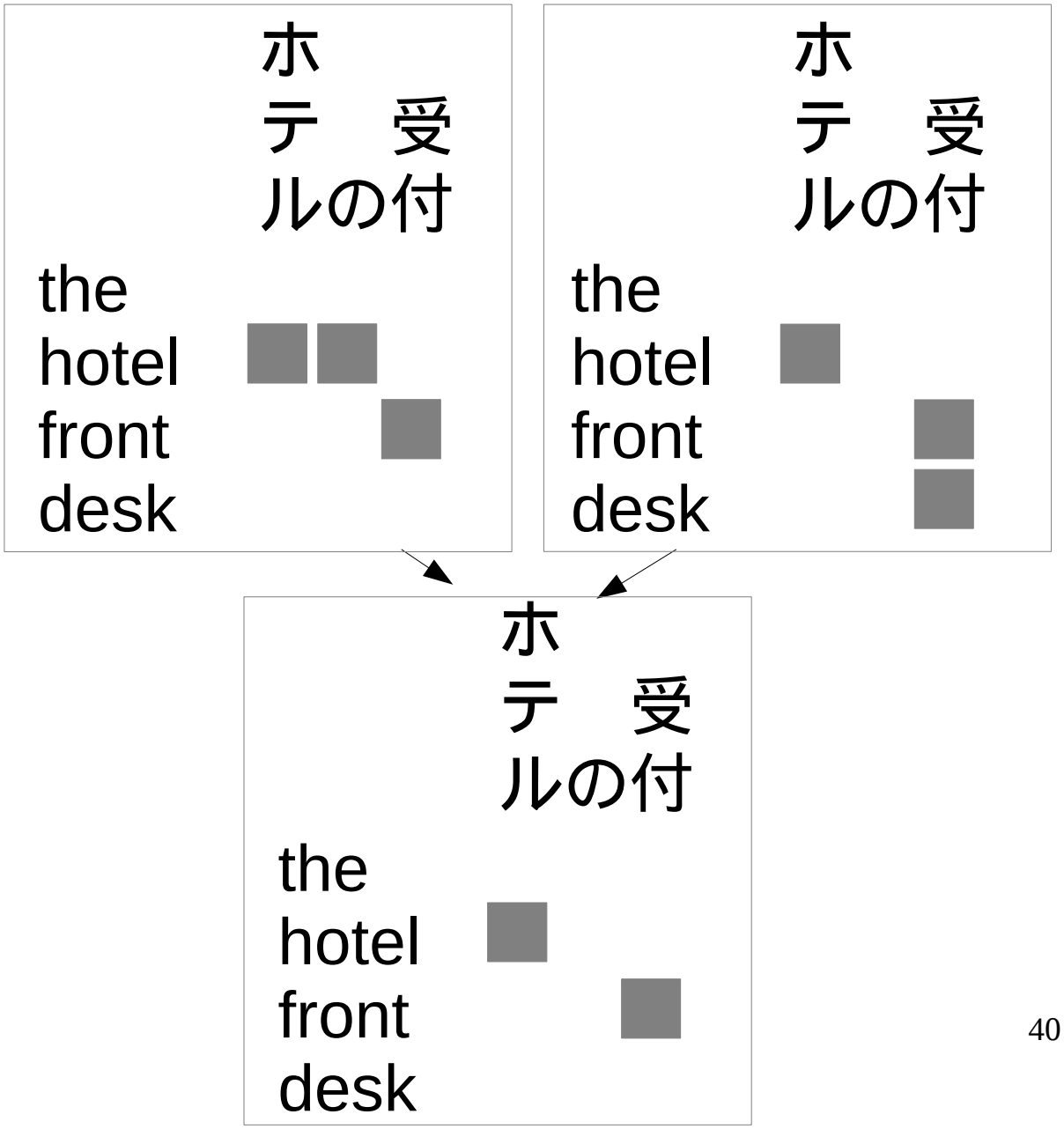
和集合

- いずれかの方向に存在すれば採用



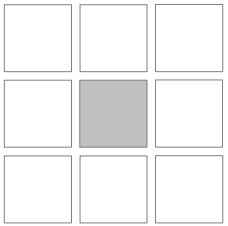
積集合

- 両方向に存在する場合のみ採用



Grow

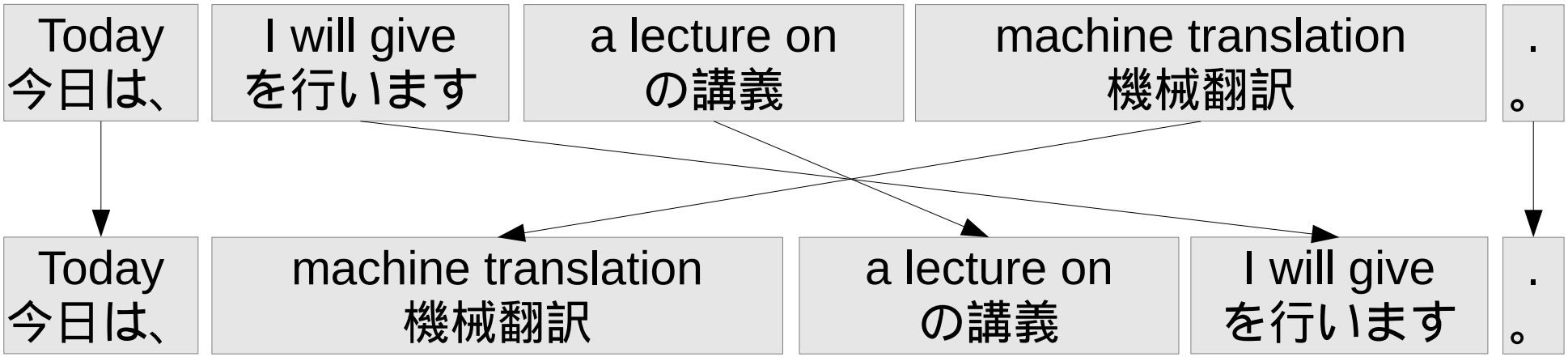
- 積集合を利用するが、積集合に隣接するものを追加 (斜めも考慮したものは grow-diag ）



フレーズ抽出

「フレーズ」とは？

- 言語学で「フレーズ（句）」は名詞句、動詞句など、文法的な役割を持つ
- 「フレーズベース翻訳」では単なる単語列

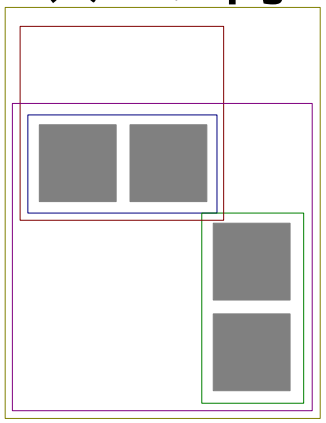


フレーズ抽出

- アライメント情報に基づきフレーズ対を抽出

ホ
テ　受
ルの付

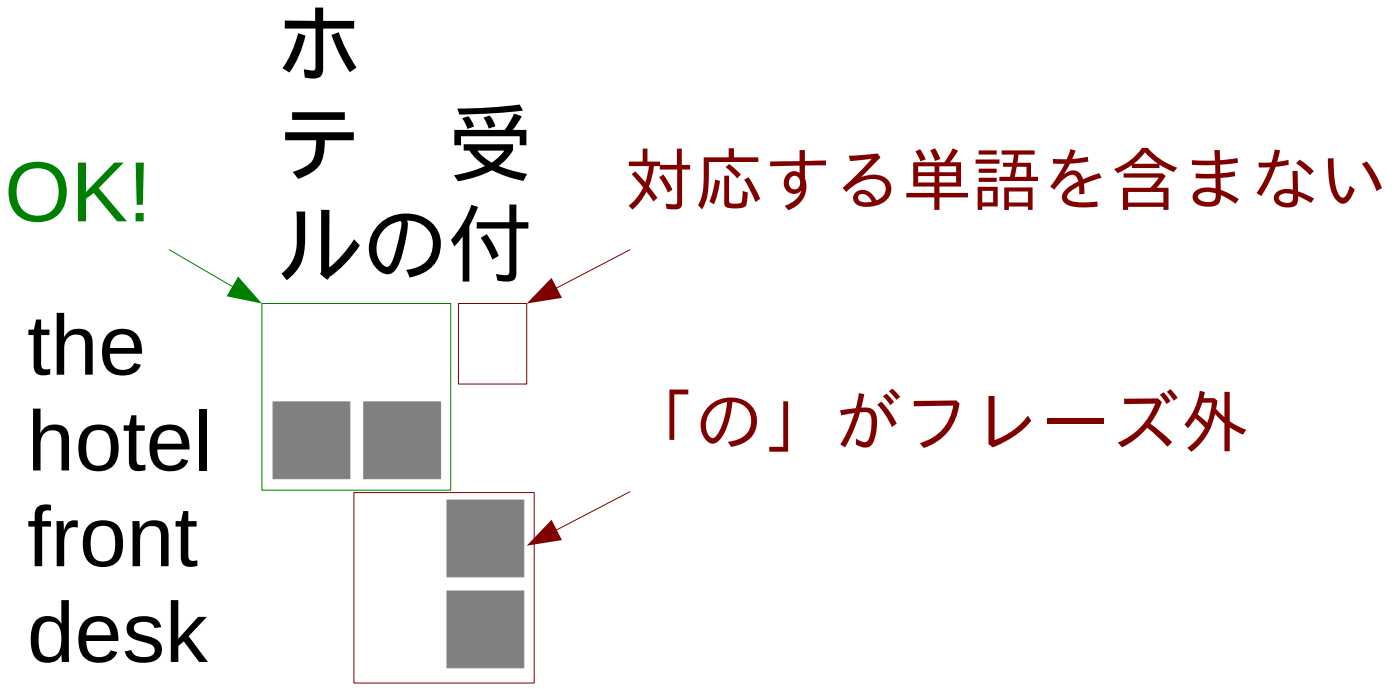
the
hotel
front
desk



ホテル の → hotel
受付 → front desk
ホテル の → the hotel
ホテルの受付 → hotel front desk
ホテルの受付 → the hotel front desk

フレーズ抽出の条件

- すべての単語列対の中で以下の条件に合致するもの
  1) 少なくとも１つの対応する単語対が中に含まれる
  2) フレーズ内の単語がフレーズ外の単語に対応しない



フレーズのスコア計算

- ５つの標準的なスコアでフレーズの信頼性・使用頻度
  - フレーズ翻訳確率
    $P(\mathbf{f}|\mathbf{e}) = c(\mathbf{f},\mathbf{e})/c(\mathbf{e})$　　$P(\mathbf{e}|\mathbf{f}) = c(\mathbf{f},\mathbf{e})/c(\mathbf{f})$

    例：　c(ホテル　の , the hotel) / c(the hotel)
  - 語彙 (lexical) 翻訳確率
    - フレーズ内の単語の翻訳確率を利用 (IBM Model 1)
    - 低頻度のフレーズ対の信頼度判定に役立つ

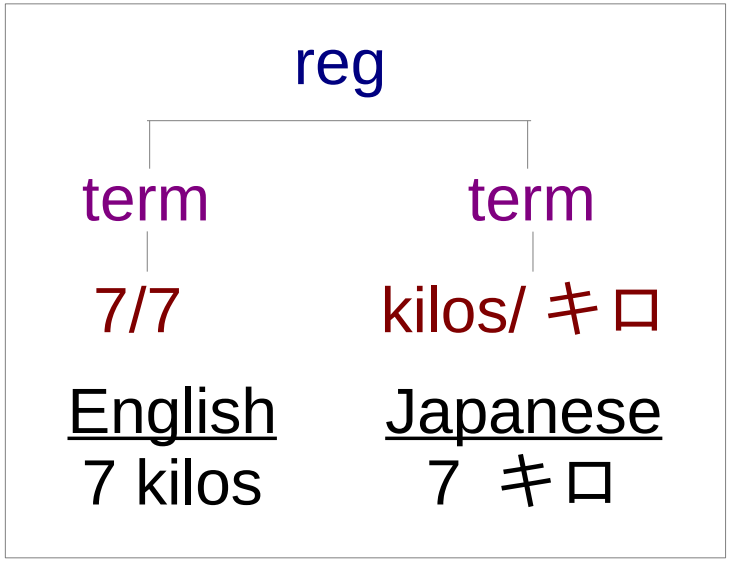
    $P(f|e) = \Pi_f \, 1/|\mathbf{e}| \sum_e P(f|e)$

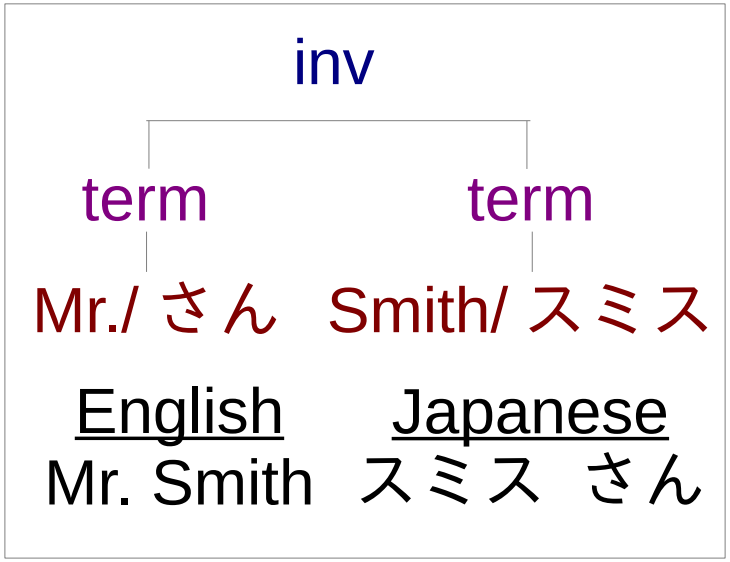
    例：
    (P(ホテル |the)+P(ホテル |hotel))/2 * (P(の |the)+P(の |hotel))/2
  - フレーズペナルティ：すべてのフレーズで 1

アライメントの発展

反転トランスダクション文法 (ITG) [Wu 97]

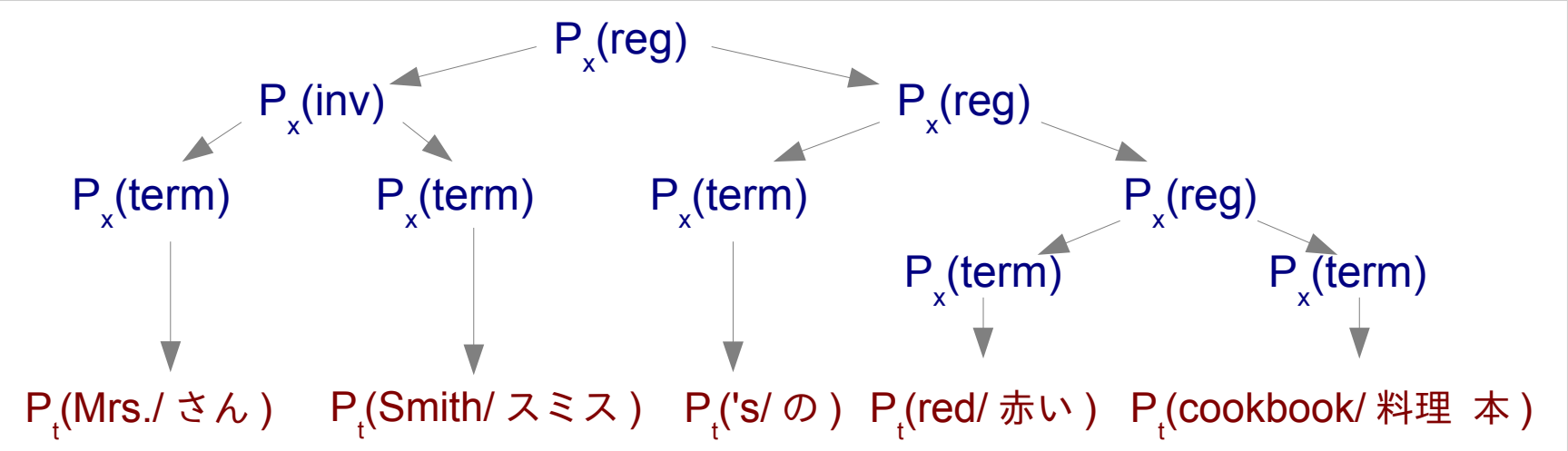
- 2 言語に対して定義される文脈自由文法の一種
  - 非終端記号　単調 (reg)　反転 (inv)
  - 前終端記号　(term)
  - 終端記号　フレーズ対



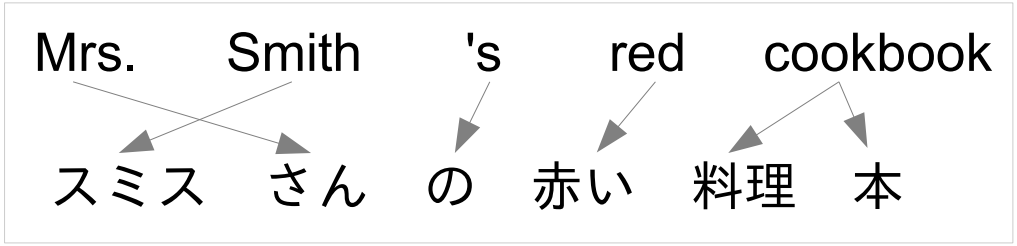


ITG の構文解析

- 確率分布を定義し、構文解析を行う
- 構文解析で広く利用される CKY アルゴリズムの一種が適応可能



- 解析結果からアライメントが一意に決まる



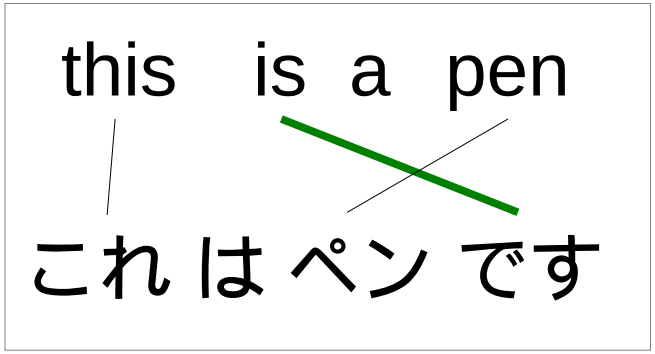
ITG の利点・欠点

- 利点：
  - 多対多アライメントをヒューリスティクスなしで対応 (ベイズ推定を使ったモデルで過学習を防ぐ [DeNero+ 08, Neubig+ 11] ）
  - 多項式時間で計算可能 $O(n^6)$
- 欠点：
  - 一対多の IBM モデルに比べて計算量が多い

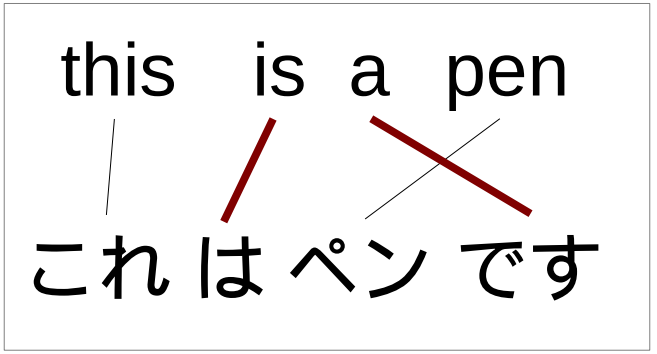
教師ありアライメント [Haghighi+ 09]

- 人手で正解を用意し、学習データとする
- 教師なしモデルの誤りを訂正するモデルを構築

正解



教師なし



重み：　c(is, です)++

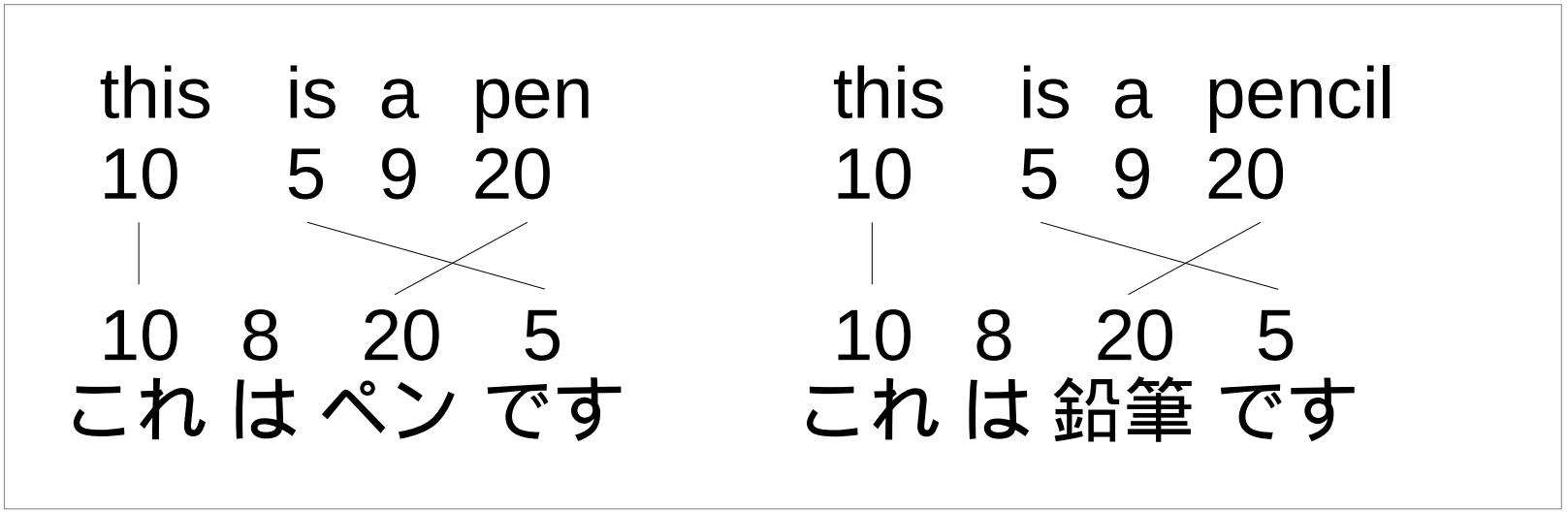
c(is, は)--

c(a, です)--

- 統語情報など、色々な情報が利用可能 [Riesa+ 10]

クラスに基づく単語アライメント [Och 99, Och+ 03]

- クラスを使ってアライメント確率を平滑化



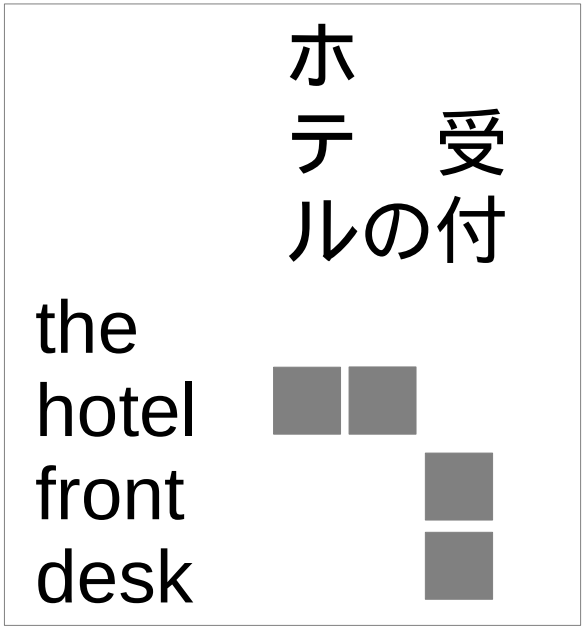
- クラスを言語間で同時に学習

アライメントの評価

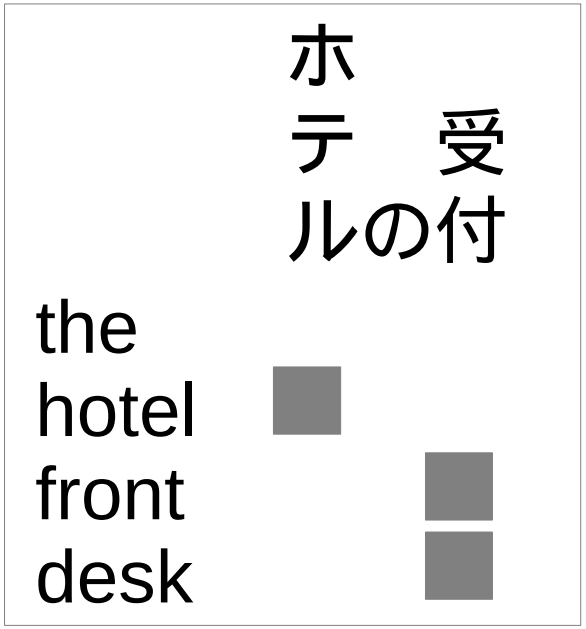
アライメントの評価

- ２つのアライメント法があった時、どれを採用？

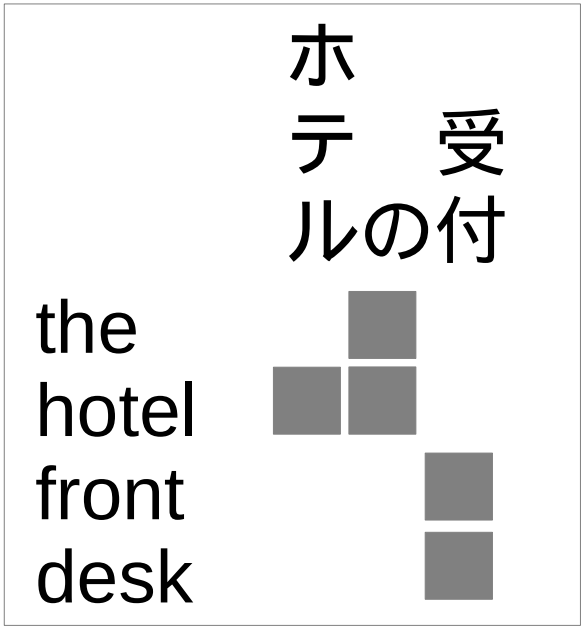
正解



システム A

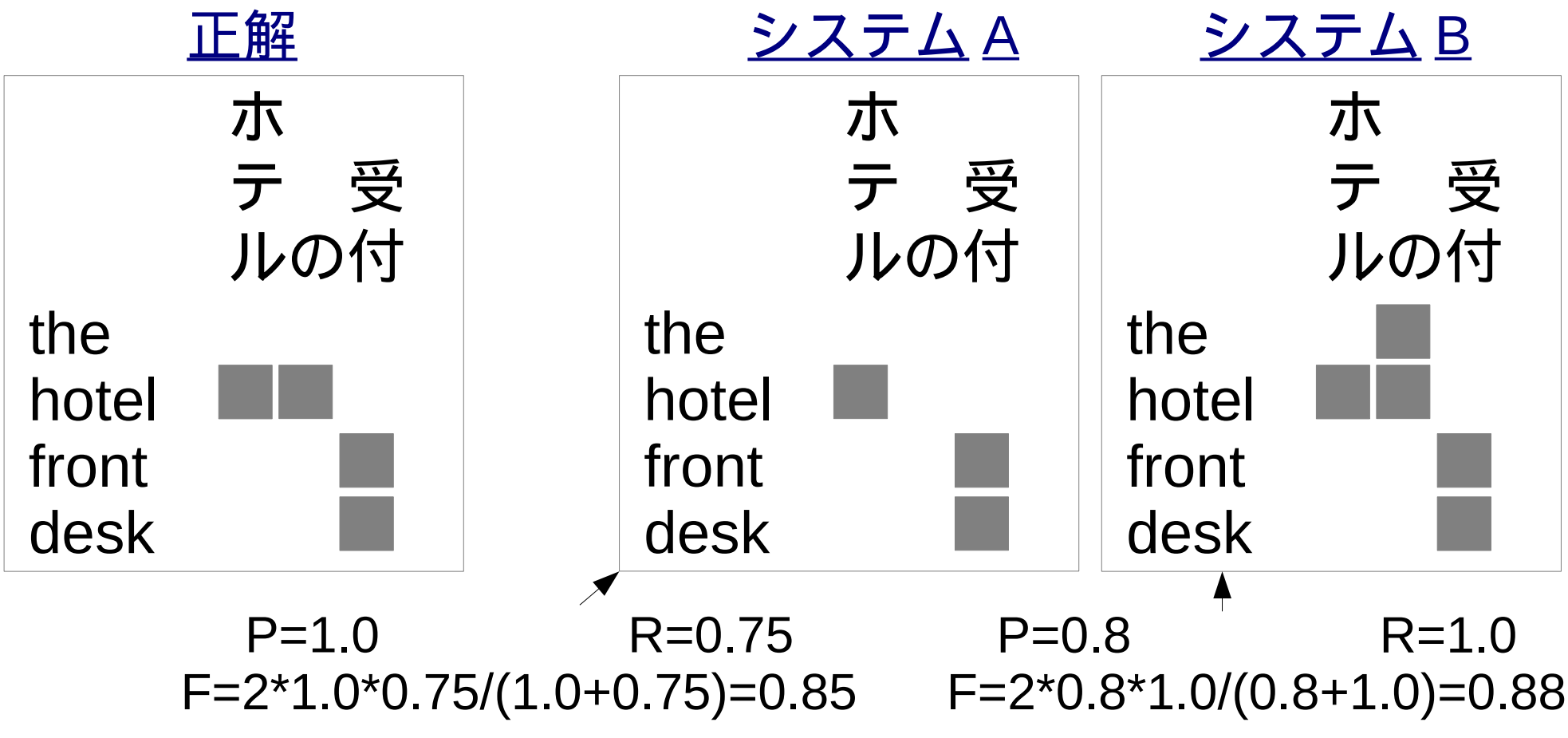


システム B



適合率・再現率・F 値

- 適合率 : システムアライメントの中で正解の割合
- 再現率 : 正解の中で、システムが出力した割合
- F 値 : 適合率と再現率の調和平均



ツール・資料

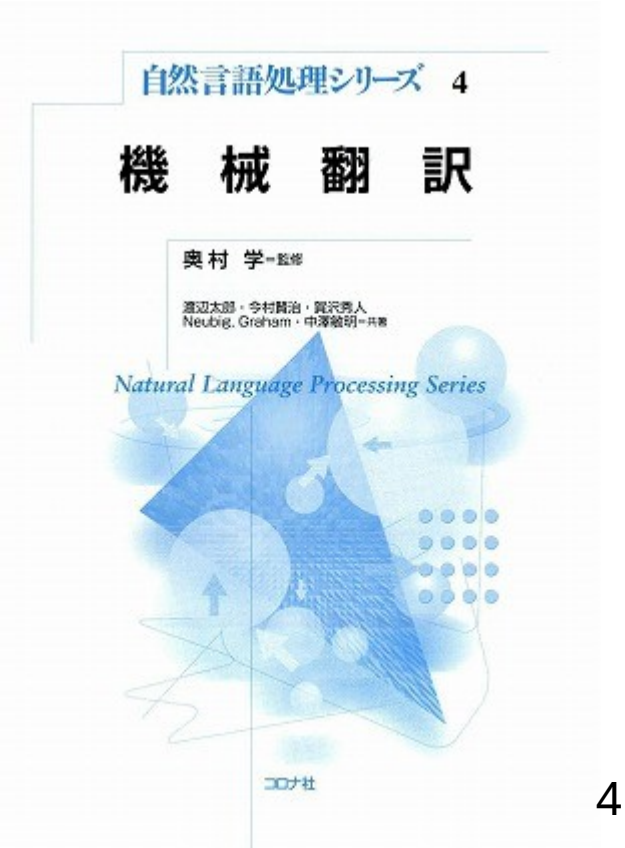
アライメントツールキット

- GIZA++:
  - 最も標準的なツール
  - IBM/HMM モデルとクラスを実装
- Nile:
  - 統語情報を用いた教師ありアライメント
  - 日英で高い精度を確認 [Neubig 13]
- Pialign:
  - ITG モデルを実装
  - フレーズベース翻訳のためのコンパクトなモデル
- fast_align:
  - IBM Model 2 の拡張版の超高速な実装
  - ただ、語順が異なる言語には不向き

人手対応付きデータ

- 日本語
  - 日英：京都フリー翻訳タスクの対応付きデータ
    http://www.phontron.com/kftt/#alignments
  - 日本語はこれ以外ない？
  - 日中近日公開？
- その他
  - 仏英・独英・チェコ英はダウンロード可
  - 中英などは購入可

更に勉強するには


自然言語処理シリーズ 4
機 械 翻 訳
奥村 学 監修
Natural Language Processing Series
コロナ社

4章

参考文献

- [1] P. F. Brown, V. J. Pietra, S. A. D. Pietra, and R. L. Mercer. The mathematics of statistical machine translation: Parameter estimation. Computational Linguistics, 19:263-312, 1993.
- [2] J. DeNero, A. Bouchard-Cote, and D. Klein. Sampling alignment structure under a Bayesian translation model. In Proc. EMNLP, pages 314- 323, Honolulu, USA, 2008.
- [3] L. R. Dice. Measures of the amount of ecologic association between species. Ecology, 26(3):297-302, 1945.
- [4] A. Haghighi, J. Blitzer, J. DeNero, and D. Klein. Better word alignments with supervised ITG models. In Proc. ACL, pages 923-931, Singapore, 2009.
- [5] P. Koehn, F. J. Och, and D. Marcu. Statistical phrase-based translation. In Proc. HLT, pages 48-54, Edmonton, Canada, 2003.
- [6] G. Neubig. Travatar: A forest-to-string machine translation engine based on tree transducers. In Proc. ACL Demo Track, Sofia, Bulgaria, August 2013.
- [7] G. Neubig, T. Watanabe, E. Sumita, S. Mori, and T. Kawahara. An unsupervised model for joint phrase alignment and extraction. In Proc. ACL, pages 632-641, Portland, USA, June 2011.
- [8] F. J. Och. An efficient method for determining bilingual word classes. In Proc. EACL, 1999.
- [9] F. J. Och and H. Ney. A systematic comparison of various statistical alignment models. Computational Linguistics, 29(1):19-51, 2003.
- [10] J. Riesa and D. Marcu. Hierarchical search for word alignment. In Proc. ACL, pages 157-166, 2010.
- [11] S. Vogel, H. Ney, and C. Tillmann. HMM-based word alignment in statistical translation. In Proc. COLING, pages 836-841, Copenhagen, Denmark, 1996.
- [12] D. Wu. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. Computational Linguistics, 23(3):377-403, 1997. 2