Breaking down the Language Barrier with Statistical Machine Translation:
2) Alignment/Phrase Extraction

http://www.phontron.com/class/sentan2014

Advanced Research Seminar I/III

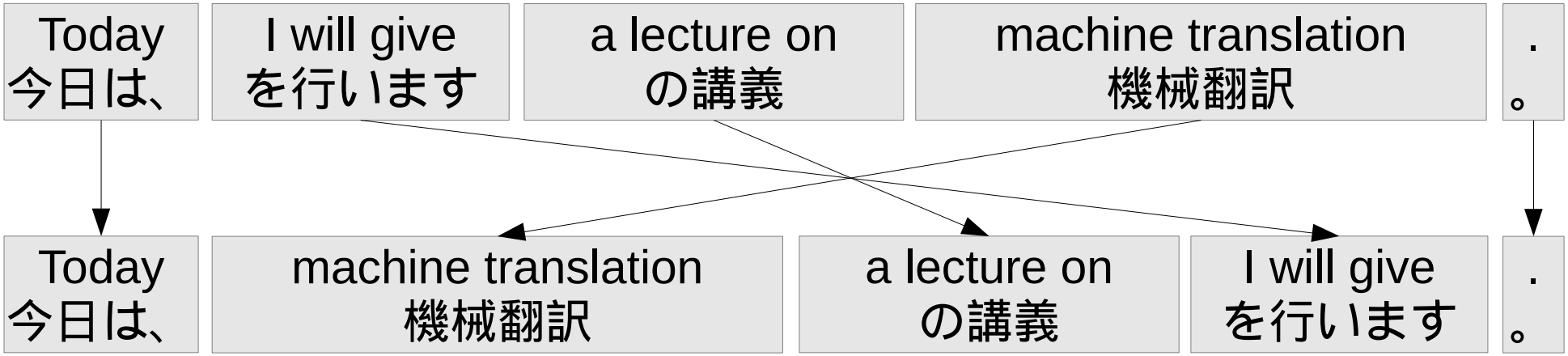
Graham Neubig

2014-1-30

How does machine translation work?

- Divide sentence into translatable patterns, reorder, combine

Today I will give a lecture on machine translation .



今日は、機械翻訳の講義を行います。

Problem

- There are millions of possible translations!

花子 が 太郎 に 会った
Hanako met Taro
Hanako met to Taro
Hanako ran in to Taro
Taro met Hanako
The Hanako met the Taro

- How do we tell which is better?

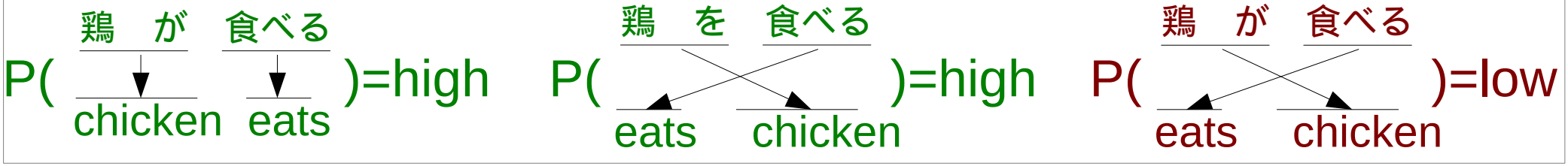
Statistical Machine Translation

- Translation model:

P(“今日” |“today”) = high P(“今日 は 、” |“today”) = medium

P(“昨日” |“today”) = low

- Reordering Model:



- Language Model:

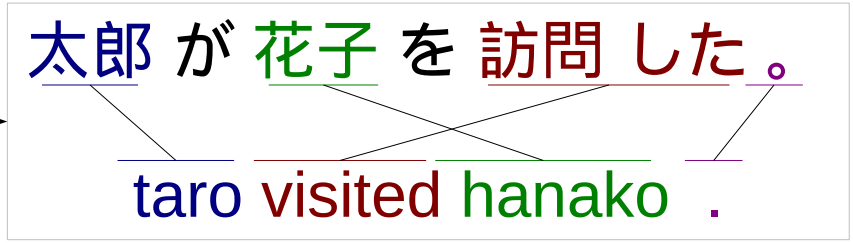
P(“Taro met Hanako”)=high P(“the Taro met the Hanako”)=low

Alignment

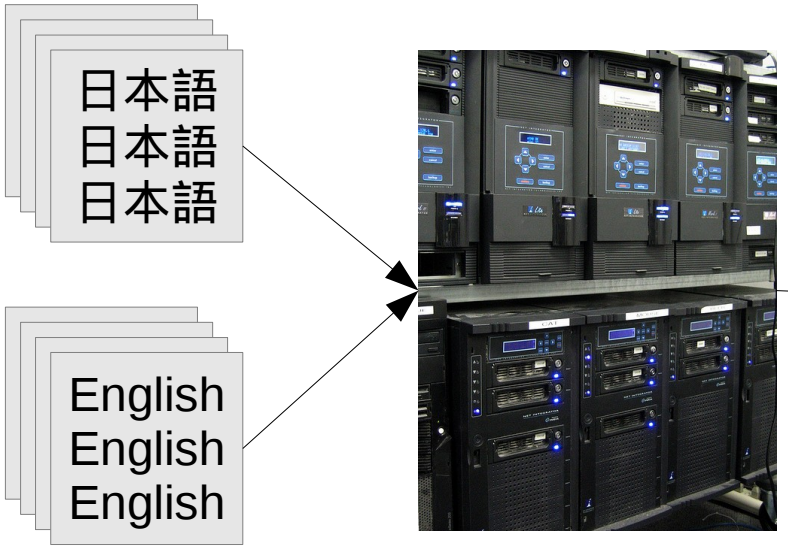
- Find the correspondences between words

太郎 が 花子 を 訪問 した 。

taro visited hanako .



- Unsupervised probabilistic models are most common



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

Assignment

- I have given you code that:
  - 1: Calculates the Dice coefficient for each pair of words
  - 2: Decides the alignment in each direction using Dice coefficient statistics and the Max Score criterion
  - 3: Combines these two alignments using the "intersection" heuristic
  - 4: Measures alignment F-measure
- You should
  - 1: Measure the alignment accuracy on the provided data
  - 2: Make one change that improves the accuracy

Heuristic Alignment

How Do we Learn Alignments?

- For example, we go to an Italian restaurant w/ Japanese menu

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

- Try to find the patterns!

How Do we Learn Alignments?

- For example, we go to an Italian restaurant w/ Japanese menu

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

- Try to find the patterns!

Co-occurrence

- Simplest measure of correlation: co-occurrence

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

Occurrence

c(チーズ) = 3
c(の) = 3
c(と) = 2
…

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

Co-occurrence

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

Problem with Co-occurrence

- Co-occurrence favors frequent words

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

Co-occurrence

c(the, 男) = 3
c(man, 男) = 2

Dice Coefficient

- The dice coefficient penalizes highly frequent words

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

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

Dice Coefficient

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

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

Scores → Alignments

- 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			●

Max-Score

- Choose the best target word for each source

	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

Threshold

- Choose a threshold, align all words over threshold

	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

$t > 0.1$

Competitive Linking

- Pick the highest scored alignment in order, but do not allow the same word be aligned to others

	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

Probabilistic Alignment: IBM Model 1

Probabilistic Alignment

- Create probabilistic model of two sentences

F= チーズ ムース

E= mousse di formaggi

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

- The model M has some parameters (e.g. probabilities)

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

- Probabilities are easier to understand (can make more refined models), easier to integrate

IBM Model One Idea

- Generate each word f_j in F according to
  - Pick one word index a_j at random ($P(a_j) = 1/(|E|+1)$), including special "NULL" word for unaligned words
  - Pick the word f_j according to $P(f \mid e_{aj})$

Generate two words

チーズ

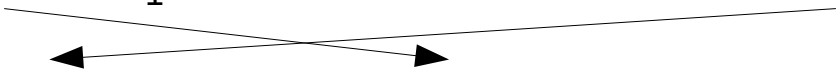
ムース

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 Model 1 Equation

- If we guess the alignment and output words, probability is

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

Alignment — Word translation

- We can also sum over all alignments

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

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

Training Model 1

- We need to estimate our model parameters
- The most natural way to do so is with maximum likelihood

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

- How do we calculate the parameters to maximize likelihood?

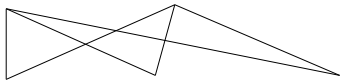
EM Algorithm

- IBM models trained using the EM (Expectation-Maximization) algorithm
- Idea:
  - E Step: Based on the model, estimate how many times each word *e* is translated into *f*
  - M Step: Based on the estimated counts, re-calculate the model parameters
- Every iteration, we increase the likelihood of the model

EM Example

- Initialize: Simply count co-occurrence

チーズ ムース

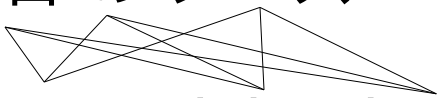


Mousse di formaggi

本日 の 鮮魚

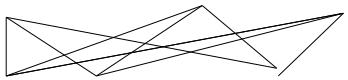
Pesce del giorno

本日 の チーズ



Formaggi del giorno

ドルチェ と チーズ

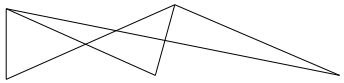


Dolce e Formaggi

EM Example

- M Step: Update the probabilities

チーズ ムース

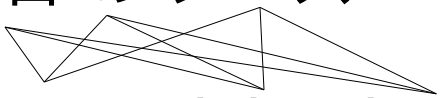


Mousse di formaggi

本日 の 鮮魚

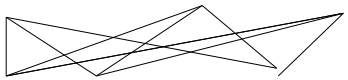
Pesce del giorno

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

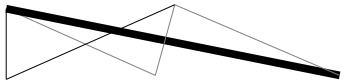
P(本日 |giorno) = 0.33
P(の |giorno) = 0.33
P(鮮魚 |giorno) = 0.16
P(チーズ |giorno) = 0.16

...

EM Example

- E Step: Calculate connections between words

チーズ ムース



Mousse di formaggi

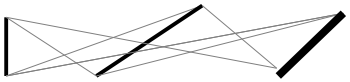
本日 の 鮮魚

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

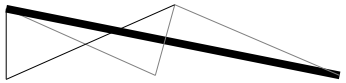
P(本日 |giorno) = 0.33
P(の |giorno) = 0.33
P(鮮魚 |giorno) = 0.16
P(チーズ |giorno) = 0.16

...

EM Example

- M Step: Update the probabilities

チーズ ムース



Mousse di formaggi

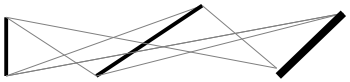
本日 の 鮮魚

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 Example

- E Step: Calculate connections between words

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

...

Initialization in Equations/Words

- Define our expected counts of two words x and y being aligned as

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

- Initialize this to the co-occurrence counts

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

M Step in Equations/Words

- M Step: Calculate the model parameters
  - Simply calculate the maximum likelihood estimate of the conditional probability

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

where

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

E Step in Equations/Words

- E Step: Calculate expectations with model fixed
  - One sentence, probability of a_j=i given model is:

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

Current word

All words

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

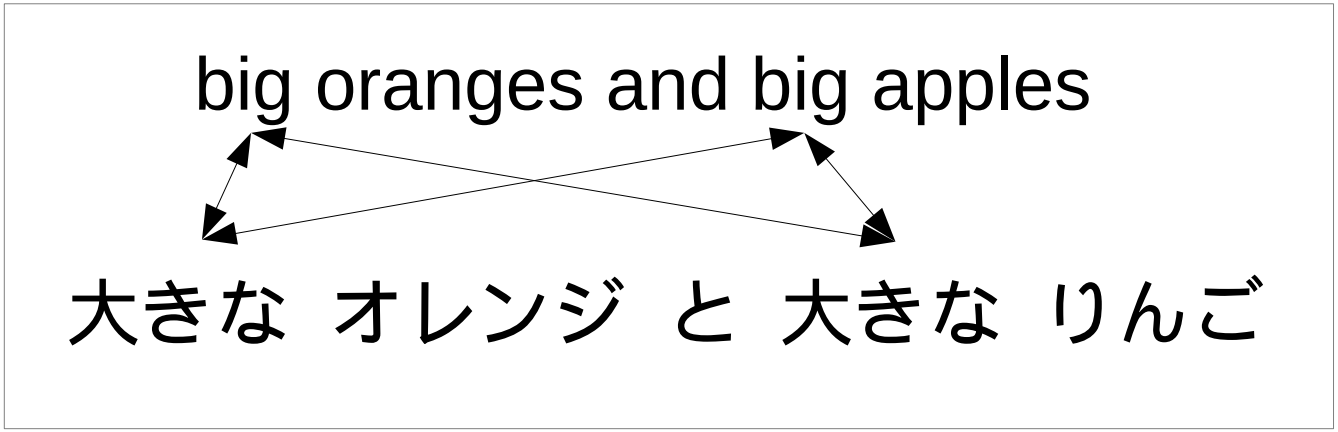
  - Given this, calculate the expected count of translations (δ = Kronecker's delta, 1 when true, 0 otherwise)

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

Probabilistic Alignment: IBM Models 2-5/HMM

Model 1 Problem

- Model 1 cannot handle word order!

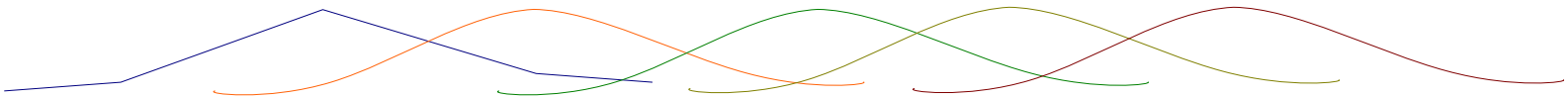


- Many new models proposed, all to handle this problem

Model 2 Idea

- “Words in both languages should be in about the same order”

big oranges and big apples

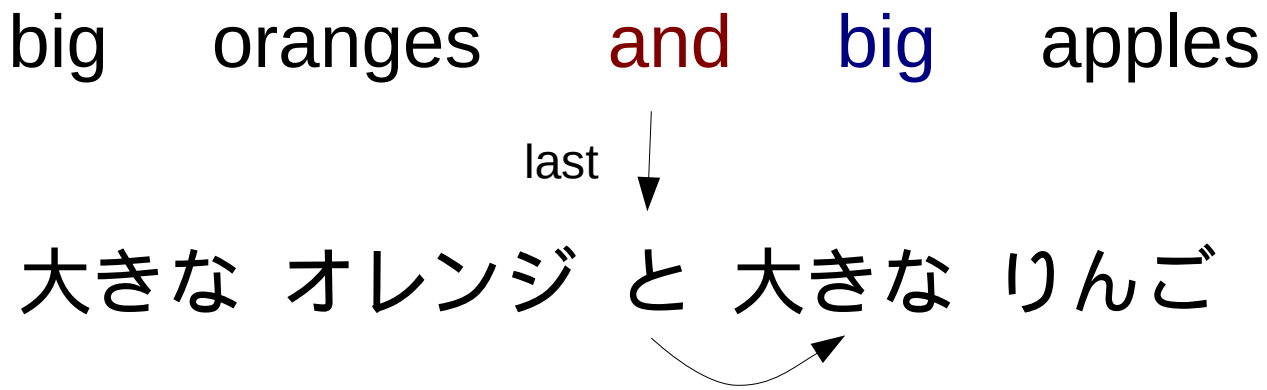


大きな オレンジ と 大きな りんご

- But, in most languages this is not true

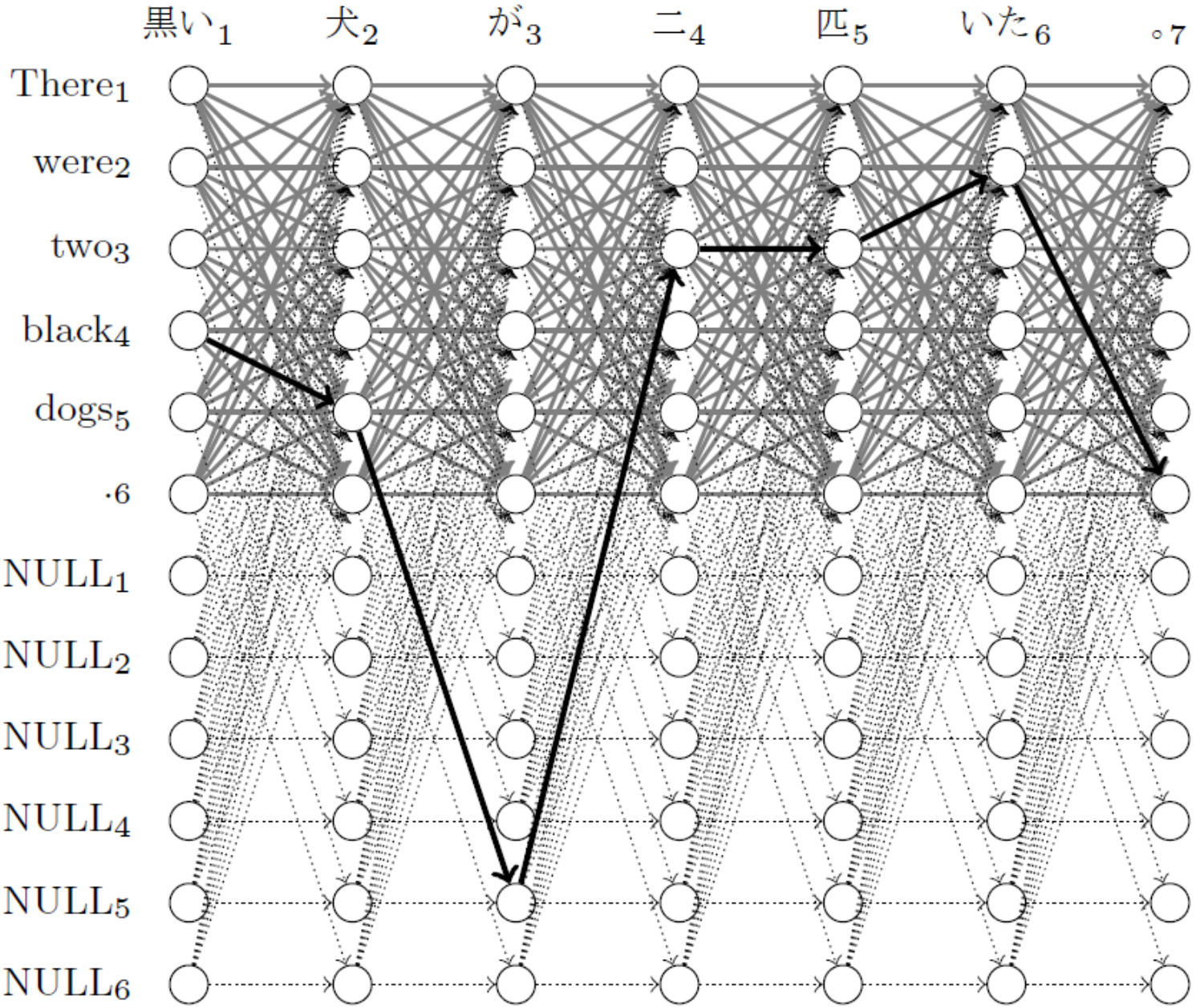
HMM Idea

- “Words that were previously aligned should give an idea about the next word”



- More effective than 2

HMM Graph


黒い1
犬2
が3
二4
匹5
いた6
。7
There1
were2
two3
black4
dogs5
·6
NULL1
NULL2
NULL3
NULL4
NULL5
NULL6

IBM Models 3-5

- Remaining IBM models are based on “fertility”
- “Given this word, how many words will it produce?”

Fertility 1

I

私
僕
俺

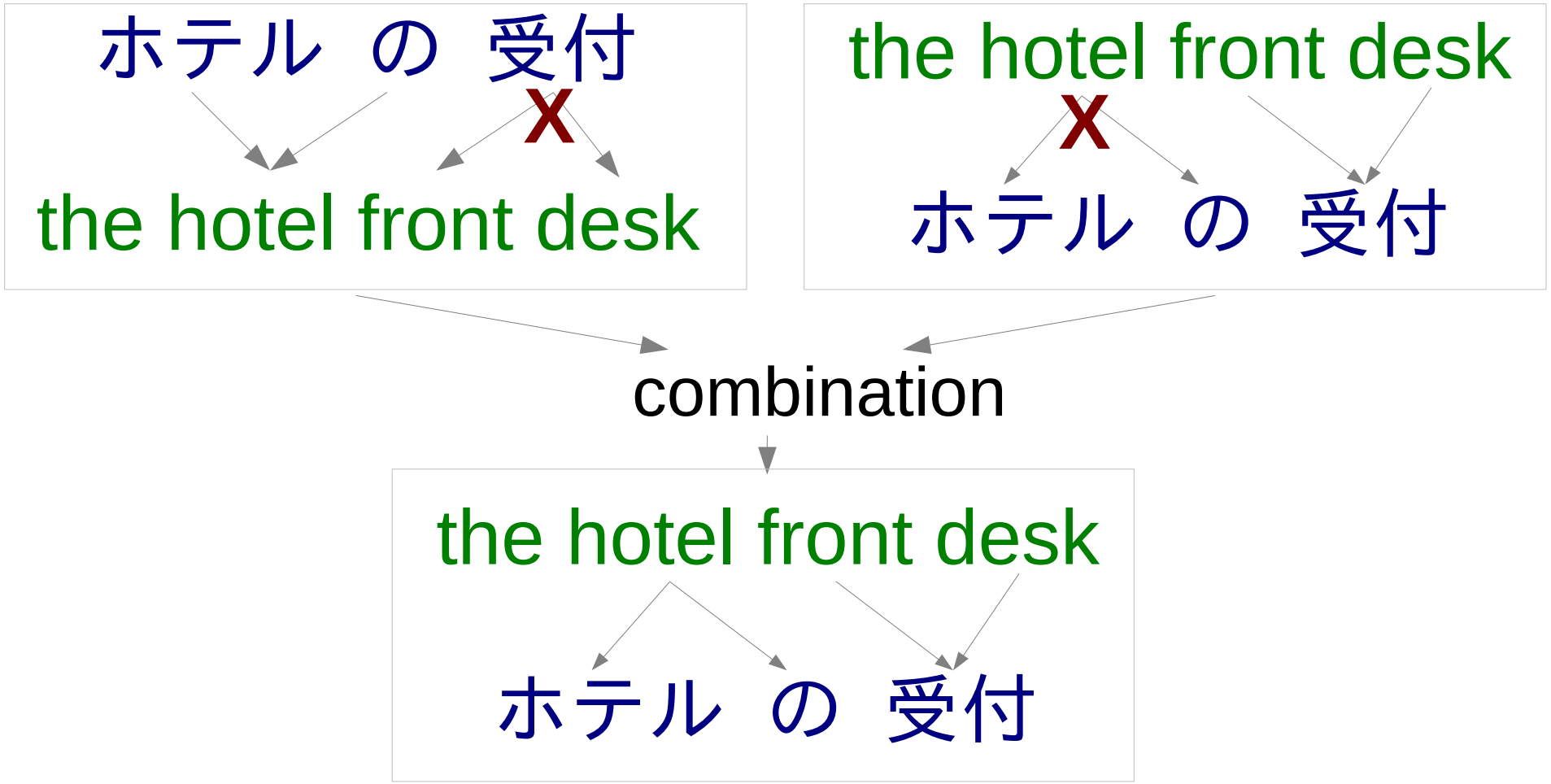
Fertility 3.5

adopted

採用 され た
養子 に なっ た

Combining Alignments

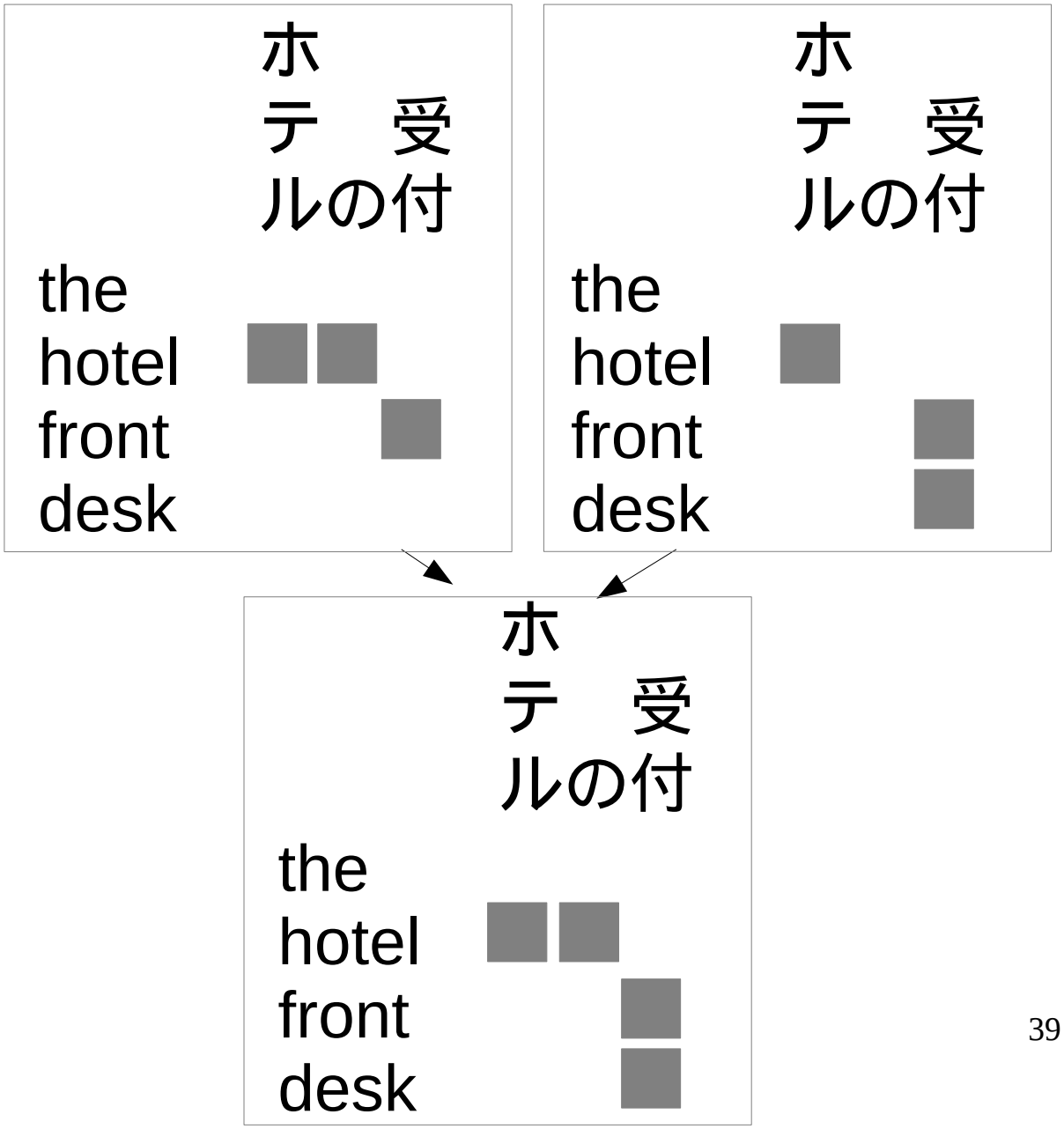
Combining 1-to-many Alignments



- Various heuristics for combination

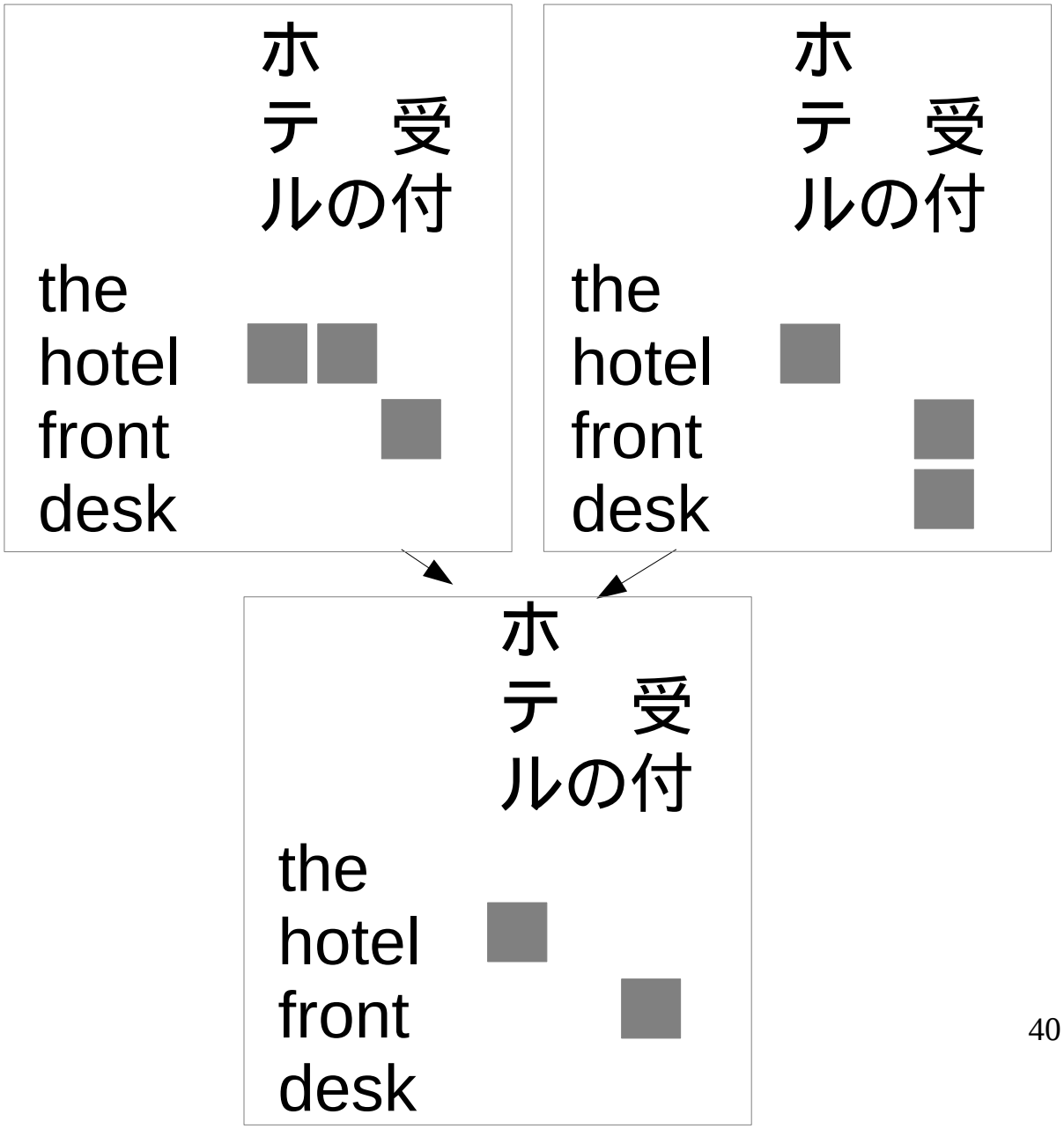
Union

- Union: Take all alignments in either direction



Intersection

- Intersection: Only take alignments in one direction



Grow/Diag

- Grow: Use intersection, but add corner pieces included in the union

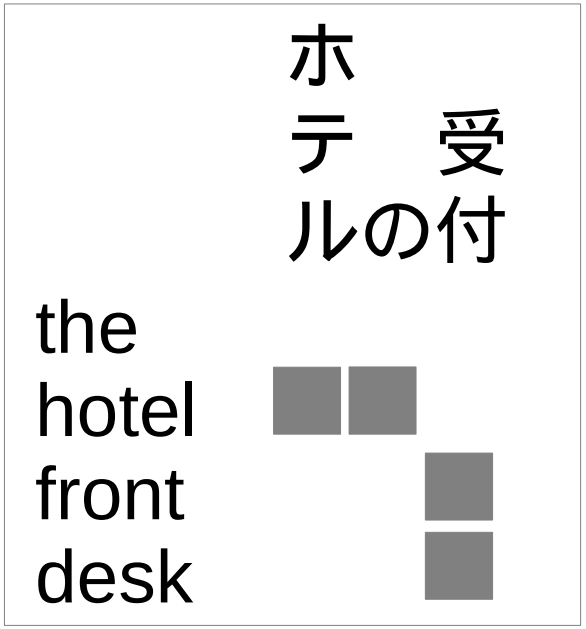
- Diag: Use intersection, but add corners

Evaluating Alignments

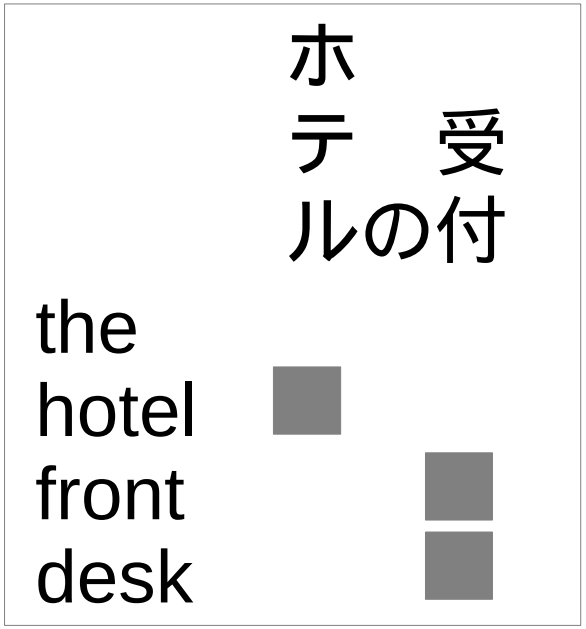
How Good are Our Alignments?

- Let's say we have two systems, which is better?

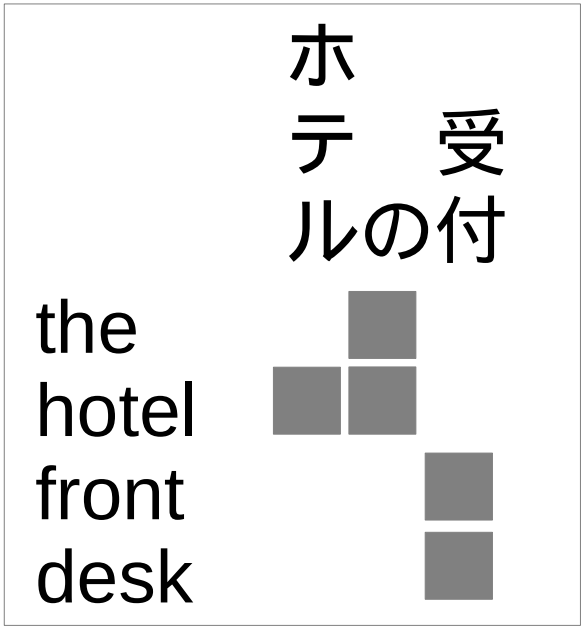
Reference



System A

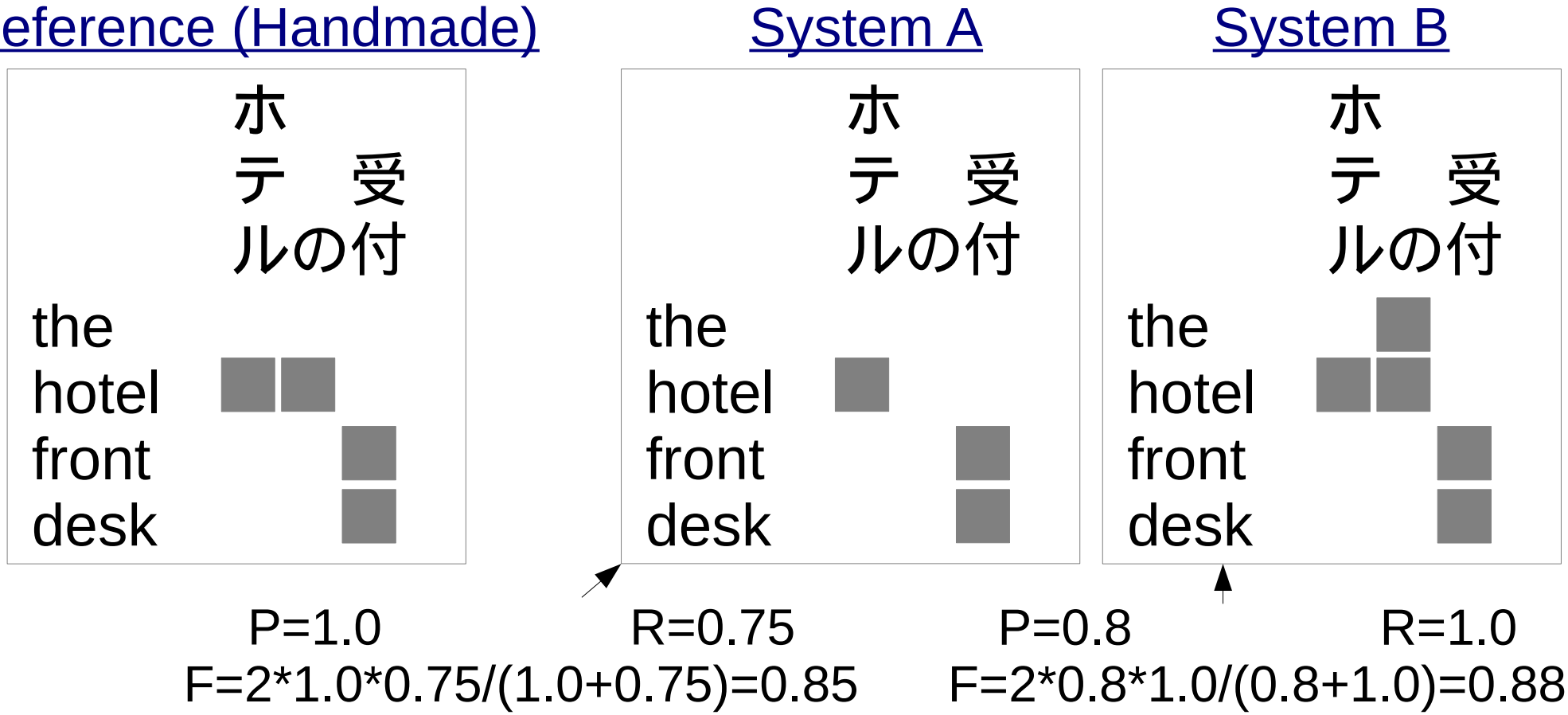


System B



Precision/Recall/F-Measure

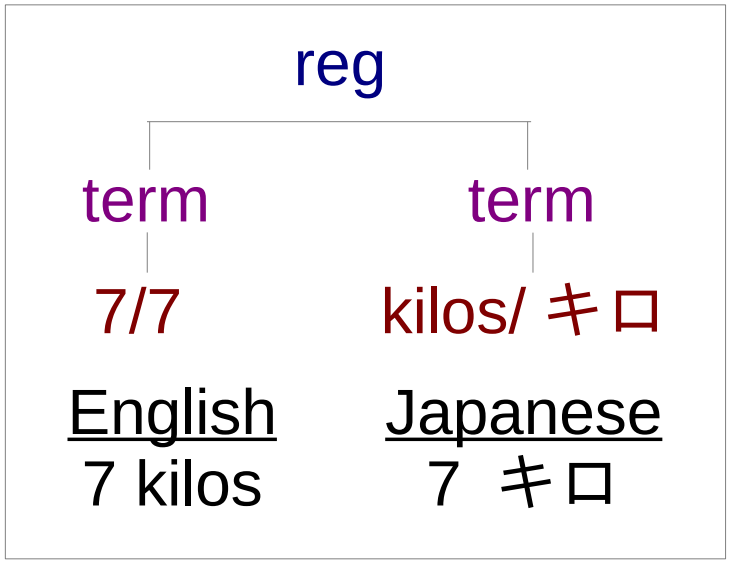
- Precision: ratio of system alignments correct
- Recall: ratio of reference alignments correct
- F-measure: geometric mean of precision and recall

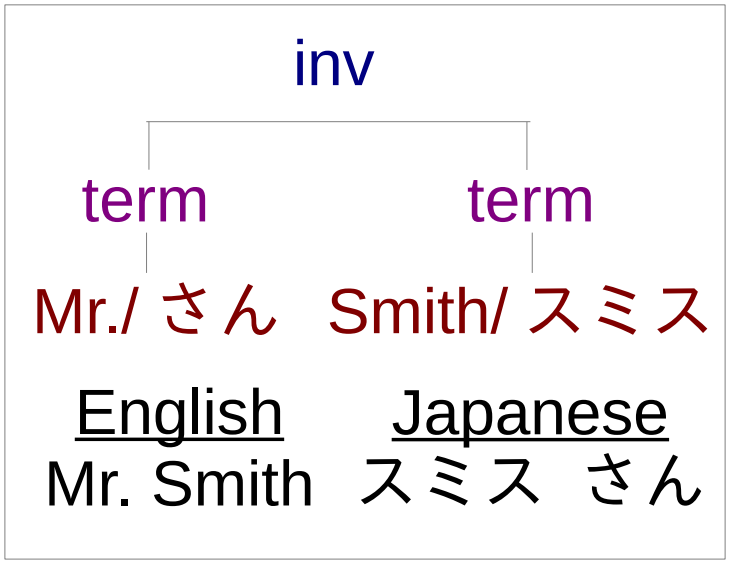


Advanced Topics

Inversion Transduction Grammar (ITG)

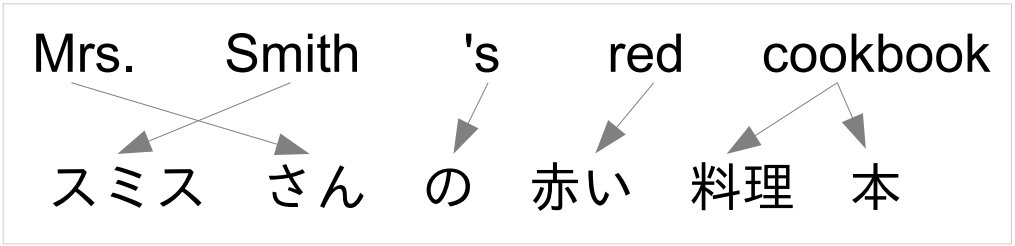
- A context-free grammar over two languages
  - non-terminal “regular (reg)” and “inverted (inv)”
  - one pre-terminal 「term」
  - terminals represent a phrase pair

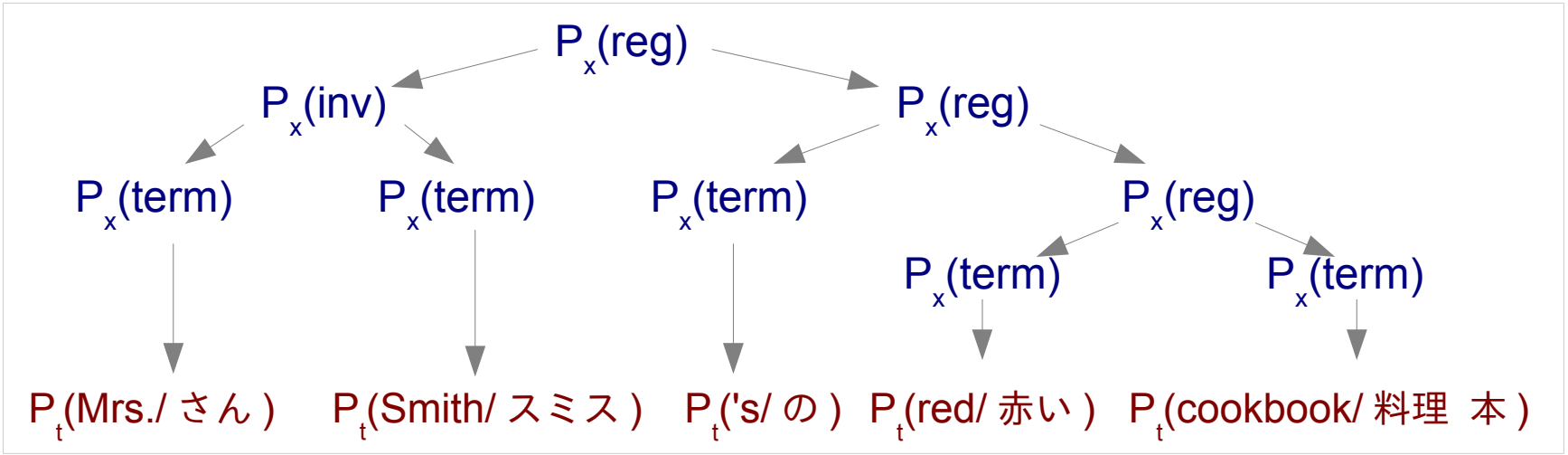




Parsing with ITGs

- Define a probability over non-terminals/pre-terminals P_x, define another over terminals P_t





- Can use a variant of the CKY algorithm (standard algorithm for parsing context-free grammars)

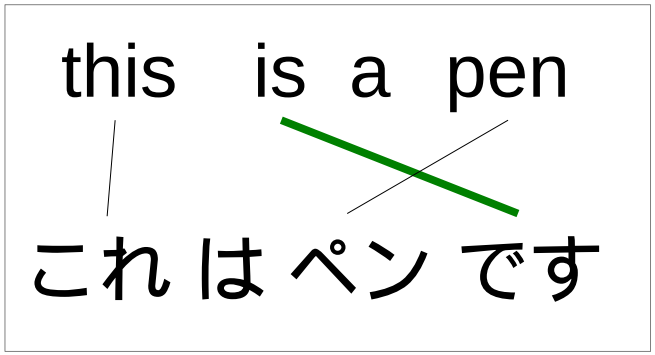
Advantages/Disadvantages of ITGs

- Advantage:
  - Simple model, no heuristics
  - Can handle many-to-many alignments in polynomial ($O(n^6)$) time
- Disadvantage:
  - Compared to other models, this is slow, especially for longer sentences

Supervised Alignment

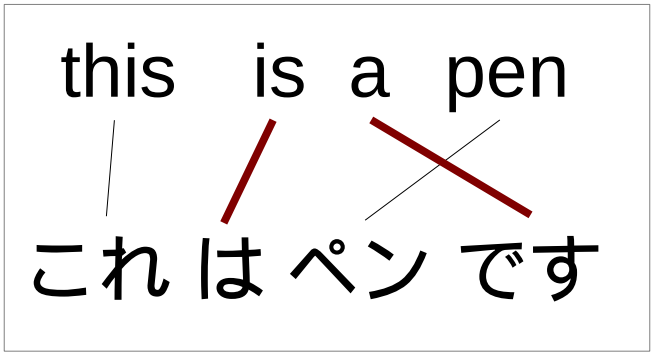
- Prepare a hand-made training corpus
- Use this corpus to train a model to “fix” unsupervised alignment

Reference



c(is, です)++

Unsupervised

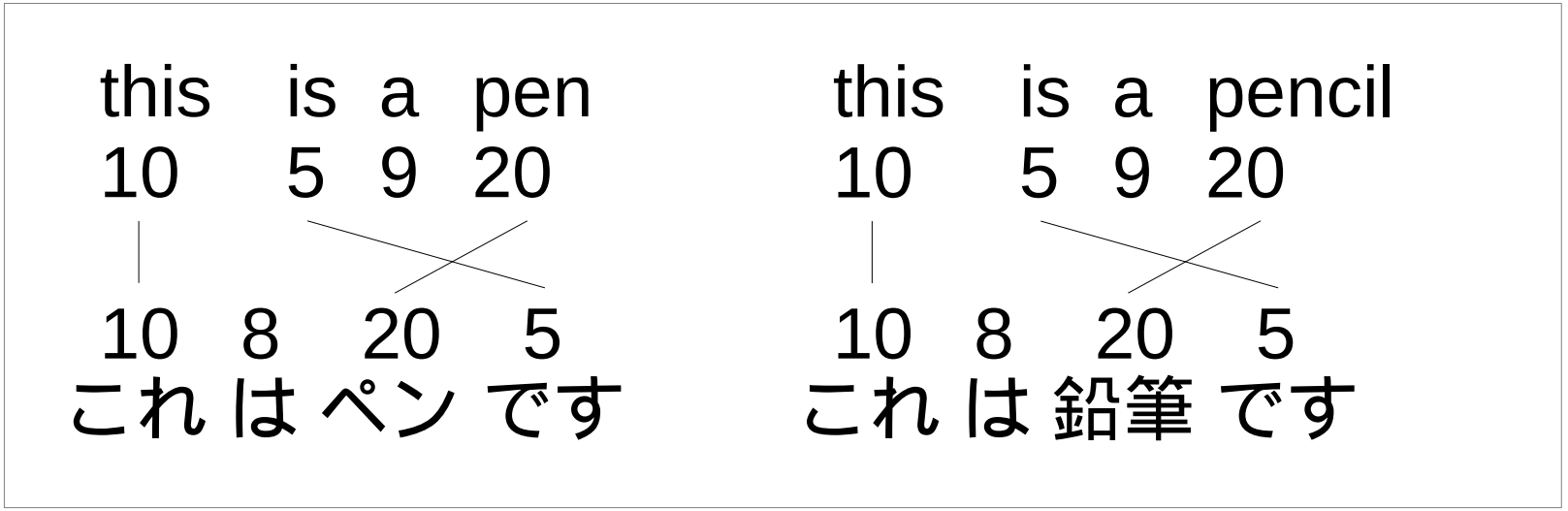


c(is, は)--

c(a, です)--

Class-Based Alignment

- Classes can be used to smooth the probability distribution

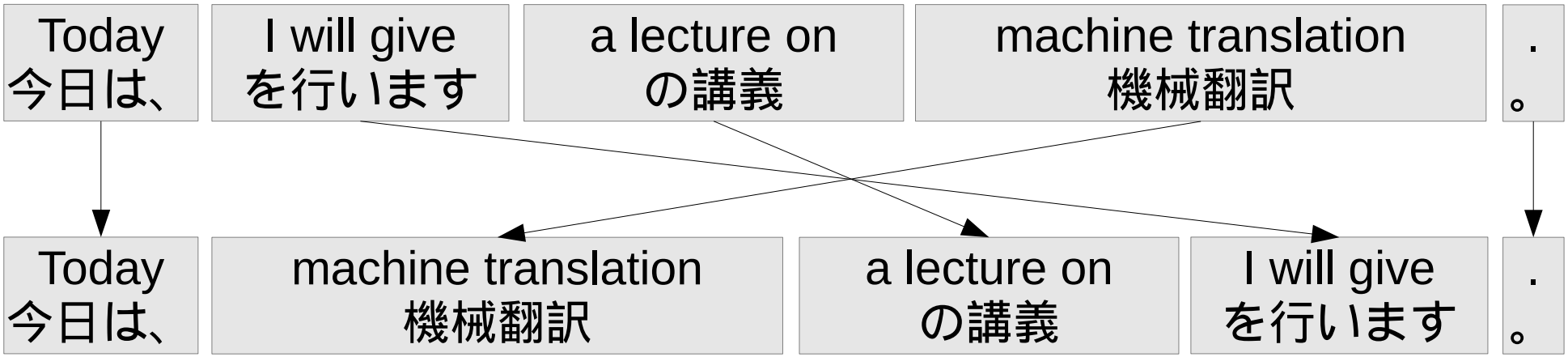


- Classes are learned automatically over both languages

Phrase Extraction/Scoring

What is a “Phrase”

- In linguistics a “phrase” has a special meaning (“noun phrase, verb phrase”)
- In translation, it generally just means a string of words

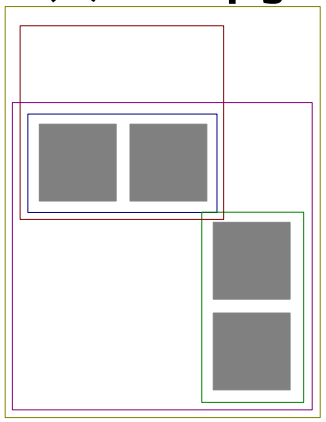


Phrase Extraction

- We extract bilingual phrases from data

ホ
テ　受
ルの付

the
hotel
front
desk



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

- Check all bilingual phrase pairs
- Extract only ones that have at least one good alignment, and no conflicts

Phrase Scoring

- We calculate probabilities for the phrases to measure how good a translation it is
  - Phrase translation probability (in both directions)
    $P(\mathbf{f}|\mathbf{e}) = c(\mathbf{f},\mathbf{e})/c(\mathbf{e})$ $P(\mathbf{e}|\mathbf{f}) = c(\mathbf{f},\mathbf{e})/c(\mathbf{f})$

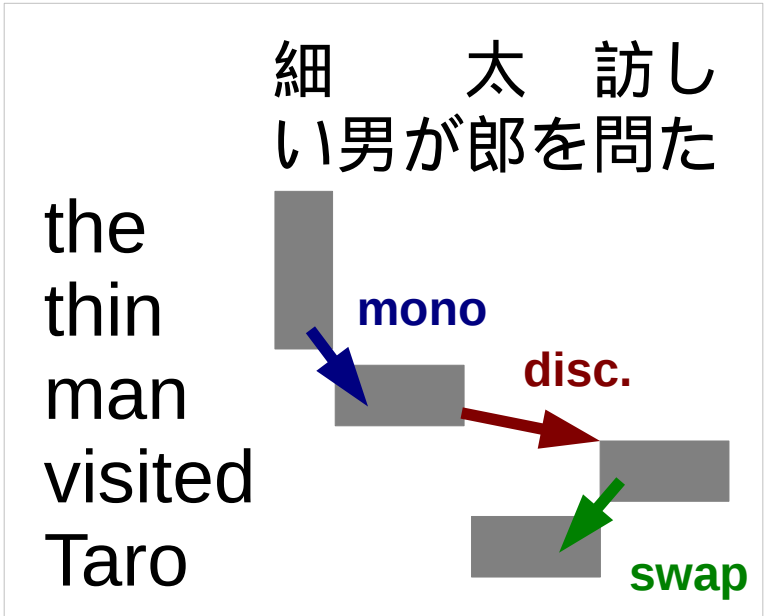
    e.g.: c(ホテル の , the hotel) / c(the hotel)
  - Lexical probabilities
    - Calculate phrase probabilities using IBM model 1 probabilities, and the words contained within
    - Helps relieve sparsity problems

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

    e.g.:
    (P(ホテル |the)+P(ホテル |hotel))/2 * (P(の |the)+P(の |hotel))/2

Reordering Varieties

- Probability of monotone, swap, discontinuous

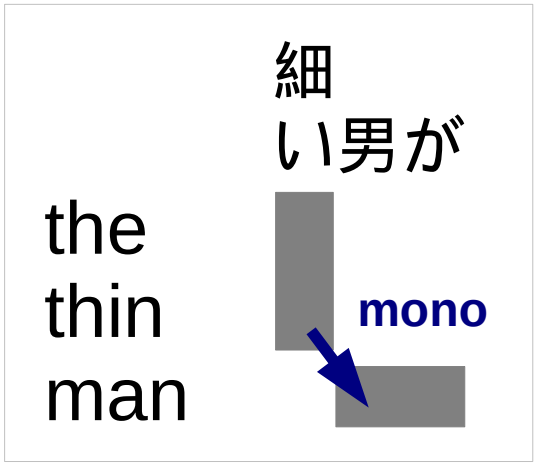


細い → the thin
high monotone probability

太郎 を → Taro
high swap probability

Reordering Probabilities

- Calculate reordering probabilities from data, like translation
- Condition on previous, next phrase



c(after, the thin → 細い , mono) ++

c(before, man → 男 が , mono) ++

Assignment

Assignment

- I have given you code that:
  - 1: Calculates the Dice coefficient for each pair of words
  - 2: Decides the alignment in each direction using Dice coefficient statistics and the Max Score criterion
  - 3: Combines these two alignments using the "intersection" heuristic
  - 4: Measures alignment F-measure
- You should
  - 1: Measure the alignment accuracy on the provided data
  - 2: Make one change that improves the accuracy

Assignment Details

- Download the exercise from the web
- You can find a list of commands to run in run-alignment.sh
  - This is currently using smaller data (kyoto-tunesmall)
  - You can use larger data (kyoto-tunetrain) if you have a powerul computer (4GB memory, 5 minutes)
- Send your code, alignment f-measure before/after, a short description of the change, and a “username”
  - Due date: February 5th, 23:59
  - Address: neubig@is.naist.jp