

# 情報アクセス論 第6回

## 「問合せ処理・ユーザインタ フェース」

情報理工学部  
前田 亮



## 前回の小テストの解説(1/2)

- 以下の索引語-文書行列 $D$ から, 問合せ $q$ と文書 $d_1 \sim d_3$ それぞれのコサイン類似度を求めよ.

類似度の値に平方根が含まれる場合は, ヒントに記載した近似値を用いて計算し, 小数点以下第1位まで(小数点以下第2位以降を四捨五入)を記入すること.

$$D = \begin{matrix} & d_1 & d_2 & d_3 \\ \begin{matrix} t_1 \\ t_2 \\ t_3 \end{matrix} & \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix} \end{matrix} \quad \mathbf{q} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$



## 前回の小テストの解説(2/2)

$$D = \begin{matrix} & d_1 & d_2 & d_3 \\ t_1 & \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \\ t_2 & \begin{bmatrix} 0 & 1 & 1 \end{bmatrix} \\ t_3 & \begin{bmatrix} 1 & 1 & 0 \end{bmatrix} \end{matrix} \quad \mathbf{q} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

コサイン類似度の公式:

$$\cos(\mathbf{d}_j, \mathbf{q}) = \frac{\mathbf{d}_j \cdot \mathbf{q}}{\|\mathbf{d}_j\| \|\mathbf{q}\|} = \frac{\sum_{i=1}^m d_{ij} q_i}{\sqrt{\sum_{i=1}^m d_{ij}^2} \sqrt{\sum_{i=1}^m q_i^2}}$$

$$\circ \cos(\mathbf{d}_1, \mathbf{q}) = \frac{1 \cdot 1 + 0 \cdot 0 + 1 \cdot 1}{\sqrt{1^2 + 0^2 + 1^2} \sqrt{1^2 + 0^2 + 1^2}} = \frac{2}{\sqrt{2} \sqrt{2}} = 1.0$$

$$\circ \cos(\mathbf{d}_2, \mathbf{q}) = \frac{0 \cdot 1 + 1 \cdot 0 + 1 \cdot 1}{\sqrt{0^2 + 1^2 + 1^2} \sqrt{1^2 + 0^2 + 1^2}} = \frac{1}{\sqrt{2} \sqrt{2}} = 0.5$$

$$\circ \cos(\mathbf{d}_3, \mathbf{q}) = \frac{0 \cdot 1 + 1 \cdot 0 + 0 \cdot 1}{\sqrt{0^2 + 1^2 + 0^2} \sqrt{1^2 + 0^2 + 1^2}} = \frac{0}{\sqrt{1} \sqrt{2}} = 0.0$$



# 情報検索のインタフェース

- 情報検索システムのユーザインタフェースは、**問合せの入力**（問合せ処理）と**検索結果の出力**に関わる
- **問合せの入力**に関しては、利用者の**情報要求**から、より良い問合せの作成を支援
- **検索結果の出力**に関しては、検索結果から必要な情報を見つけやすくする支援



# 問合せ処理とは？

- 問合せを**修正**したり, 語を**追加**して, より良い検索結果を得るための処理
- 伝統的な情報検索における問合せ処理
  - **問合せ拡張**
    - ・ 問合せ中の語の同義語や関連語を追加する
    - ・ 語と語の関連を示す情報が必要
  - **適合性フィードバック**
    - ・ 検索結果から問合せ中の**検索語の重み**を修正して, 再度検索する
    - ・ **ベクトル空間モデル・確率モデル**に適用可能



# Web検索エンジンにおける問合せ処理

- 最近のWeb検索エンジンでは, さまざまな問合せ入力インタフェースが用意されている

- **スペル修正**

- 誤字を自動修正(あるいは修正候補を提示)
- スペリング辞書と問合せの文字の類似度を用いる

- **問合せ候補の提示**

- 1文字入力するごとに問合せの候補のリストを提示
- 主に問合せログの解析により行われる

# ... | スペル修正の例



serect

すべて

動画

画像

ショッピング

地図

もっと見る

約 2,280,000,000 件 (0.55 秒)

次の検索結果を表示しています **select**

元の検索キーワード: serect

英語 ▾



日本語 ▾

select

編集

sə'lekt



選択

Sentaku

# 問合せ候補の提示機能の例



×



立命館大学

立命館大学 偏差値

立命館守山高校

立命館

立命館大学 図書館

立花孝志

立木観音

立木神社

立山黒部アルペンルート

立川

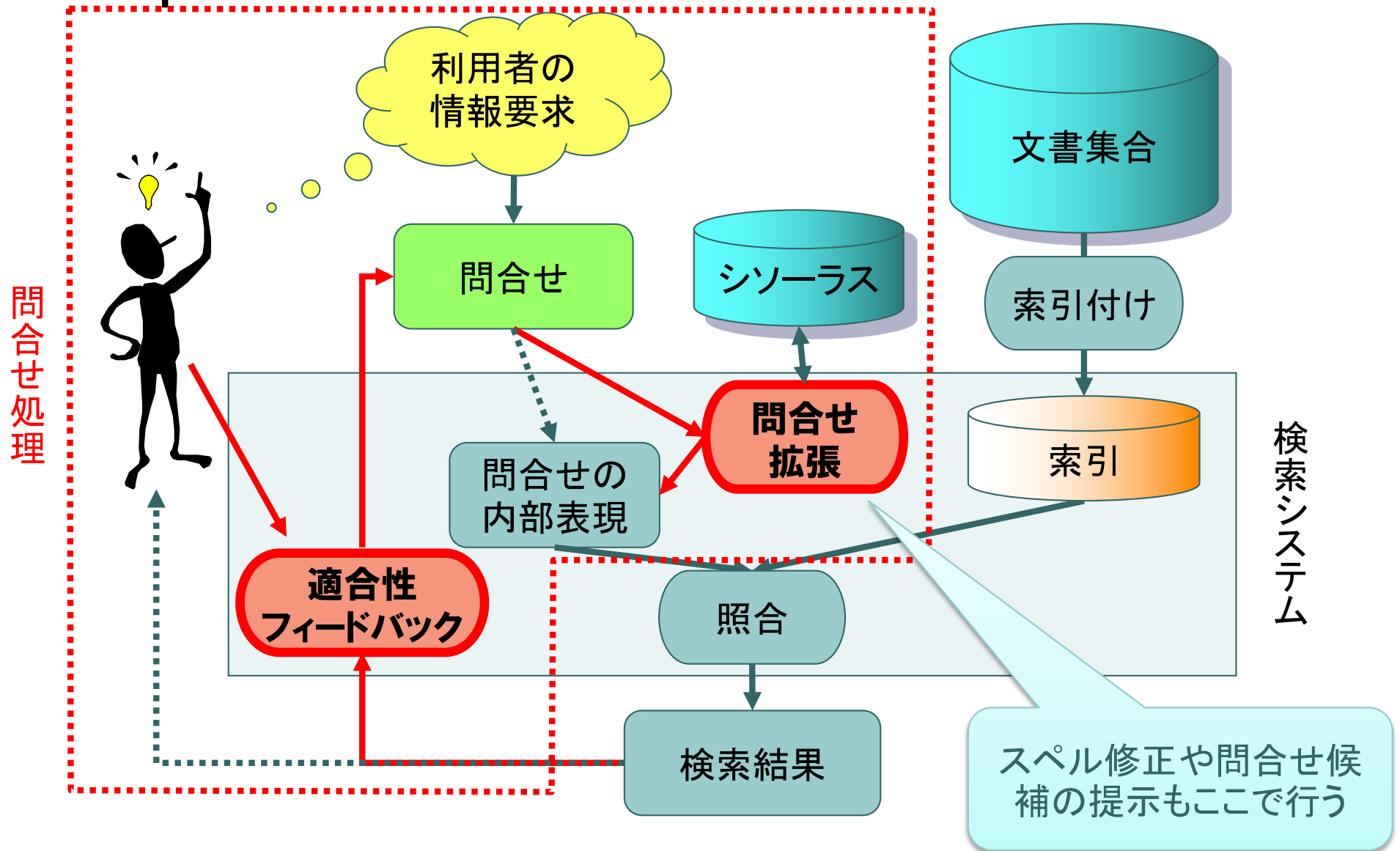
Google 検索

I'm Feeling Lucky

不適切な検索候補の報告



# 情報検索のモデルにおける問合せ処理



# 問合せ拡張とは？

## ○ 言語表現の多様性の問題を解決する

### ● 語形の多様性

- 接辞処理／形態素解析で解決

### ● 語選択の多様性

- 同じ概念を表すのに複数の表現がある
  - 「自動車」「車」「クルマ」…, 「本」「図書」「書籍」「書物」…

- 表記の揺れの問題

- 「沈殿」「沈澱」「超伝導」「超電導」「インターフェース」「インタフェース」「インターフェイス」「インタフェイス」

- 利用者が問合せに用いる語と、文書中に用いられる語が一致しないと、その文書は検索されない

- 問合せ拡張は**検索漏れ**の防止に役立つ

# 問合せ拡張の例

The screenshot shows a Google search for 'インタフェース' (Interface). The search bar contains 'インタフェース' and is highlighted with a red box. Three red arrows originate from this box and point to three different search results, illustrating query expansion. The first arrow points to a result from Weblio国語辞典. The second arrow points to a result from Wikipedia, where the word 'インターフェース' is highlighted with a red box. The third arrow points to another result from Wikipedia, where the words 'インタフェース' and 'インターフェース' are both highlighted with red boxes. The search results list shows approximately 1,460,000 results in 0.42 seconds. The right sidebar contains a definition of 'インタフェース' (Interface) in the context of information technology.

Google インタフェース - Google 検索

google.com/search?q=インタフェース&hl=ja&ei=CSI-ZK7IIJnl2roP6PWUyAU&oq=インタフェース&g...

Google

インタフェース

すべて 画像 ショッピング ニュース 動画 もっと見る ツール

約 1,460,000 件 (0.42 秒)

Weblio国語辞典  
https://www.weblio.jp/content/インタフェース  
「インタフェース」の意味や使い方 わかりやすく解説 ...  
1 異なる種類のものを結びつけるときの共用部分。界面。接触面。2 コンピューターで、機器やプログラムどうしをつなぐ装置、または部分。ハードウェアを接続するハード ...

Wikipedia  
https://ja.wikipedia.org/wiki/インターフェース  
インターフェース Wikipedia  
インターフェース (英: interface) はインタフェース、インタフェイス、インターフェイスとも書き、英語で界面や接触面、中間面などといった意味を持ち、転じて ...

他の人はこちらも質問 :

インターフェイスとはなにか?

インタフェース <  
情報技術

情報技術において、インタフェースは、情報行うシステム間のプロトコル、または、その部分をいう。コンピュータシステムの各部分はシステム間の接続や、人間と機械の間の入がある。インターフェイスあるいはインタスなどと表記することもある。ウィキペディ

フィ

# 問合せ拡張の手法

- 従来から、**関連語辞書(シソーラス)**を用いる手法が用いられてきた
  - 例: 「**自動車**」→「(自動車 OR 車 OR クルマ)」
  - 新語や専門用語に対応できない
  - 単純に辞書に載っている関連語を追加するだけではうまくいかない
    - ・ 「金」は, money, gold, Fridayのどの意味か？
- 最近では, コーパス中からの**共起頻度**の情報を用いる手法も用いられる

# スペル修正の手法

- **スペリング辞書**に載っていない単語が問合せとして入力された場合に、修正候補を提示
  - **スペリング辞書**: 正しいスペルの単語のリスト
- スペルミスと思われる単語と**スペリング辞書**の各単語との**文字の類似度**を計算
  - 類似度が高いものを修正候補とする
- 単語間の文字の類似度の計算には**編集距離 (Edit Distance)**が良く用いられる
  - ある単語を別の単語に変換するのに必要な**操作**の数



# 編集距離

- 代表的なものは、**レーベンシュタイン距離**
  - ある単語を別の単語に変換するのに必要な操作（1文字の**挿入・削除・置換**）の**最小回数**
  - 編集距離が1の例
    - 立命**位**館大学 → 立命館大学（**削除**1回）
    - 立命大学 → 立命**館**大学（**挿入**1回）
    - 立命**官**大学 → 立命**館**大学（**置換**1回）
  - 編集距離が2の例
    - 立**大**館**命**学 → 立**命**館**命**学 → 立**命**館**大**学（**置換**2回）  
（置換の操作は、同じ位置の文字のみが対象）

# 編集距離の計算の例

## ○ tummorowとtomorrowの編集距離

tummorow



2文字目の「u」を「o」に置換

tommorow



4文字目の「m」を削除

tomorrow



5文字目の後ろに「r」を挿入

tomorrow

3回の操作が必要なので、編集距離は3

## ○ ある単語を別の単語に変換するのに必要な操作は、複数存在する

- 必要な操作の**最小**回数を求める必要

# 編集距離の計算方法(1)

- 変換前の文字列  $a$  の長さを  $n$ , 変換後の文字列  $b$  の長さを  $m$  として,  $n+1$  行  $\times$   $m+1$  列の2次元配列を用意し, 以下を代入

$$\begin{cases} C[i, 0] = i \quad (0 \leq i \leq n) \\ C[0, j] = j \quad (0 \leq j \leq m) \end{cases}$$

- 例: 変換前が3文字 ( $n=3$ ), 変換後が4文字 ( $m=4$ ) であれば,

変換後 ( $b$ )

	0	1	2	3	4
変換前 ( $a$ )	1				
2					
3					



## 編集距離の計算方法(2)

- 配列の要素  $C[i, j]$  ( $i > 0, j > 0$ ) は, 文字列  $a$  の先頭から  $i$  文字と文字列  $b$  の先頭から  $j$  文字を一致させるために必要な最小の操作の回数

		変換後( $b$ )				
		0	1	2	3	4
変換前( $a$ )	1		$C[1,1]$	$C[1,2]$	$C[1,3]$	$C[1,4]$
	2		$C[2,1]$	$C[2,2]$	$C[2,3]$	$C[2,4]$
	3		$C[3,1]$	$C[3,2]$	$C[3,3]$	$C[3,4]$
	4					

## 編集距離の計算方法(3)

- 文字列  $a$  の  $i$  文字目と文字列  $b$  の  $j$  文字目が一致している場合
  - $a$  の  $i-1$  文字目までと  $b$  の  $j-1$  文字目までを一致させれば良い
  - $a$  の  $i$  文字目に対する操作は必要ない
  - したがって,  $C[i, j] = C[i-1, j-1]$  となる

		変換後(b)				
		b	l	u	e	
変換前(a)	b	0	1	2	3	4
	r	1	0			
		2		?		
	u	3			?	

## 編集距離の計算方法(4)

- そうでない場合, 文字列  $a$  の  $i$  文字目までを  $b$  の  $j$  文字目までと一致させる方法は①～③のどれか

①  $a$  の  $i-1$  文字目までを  $b$  の  $j$  文字目までと一致させ ( $C[i-1, j]$ ),  $a$  の  $i$  文字目を**削除**

	$j-1$	$j$
$i-1$		$C[i-1, j]$
$i$		↓ 削除

②  $a$  の  $i$  文字目までを  $b$  の  $j-1$  文字目までと一致させ ( $C[i, j-1]$ ),  $a$  の  $i$  文字目の直後に  $b$  の  $j$  文字目を**挿入**

	$j-1$	$j$
$i-1$		
$i$	$C[i, j-1]$	→ 挿入

③  $a$  の  $i-1$  文字目までを  $b$  の  $j-1$  文字目までと一致させ ( $C[i-1, j-1]$ ),  $a$  の  $i$  文字目を  $b$  の  $j$  文字目に**置換**

	$j-1$	$j$
$i-1$	$C[i-1, j-1]$	
$i$		↘ 置換

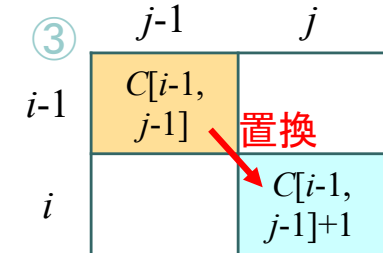
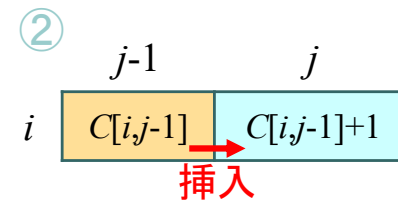
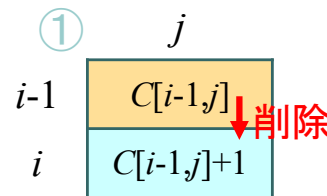
## 編集距離の計算方法(5)

- 必要な操作の回数は、一致させる部分の操作回数に加えて、それぞれ削除・挿入・置換のいずれかの操作が1回であるため、以下の通り

①  $C[i-1, j] + 1$

②  $C[i, j-1] + 1$

③  $C[i-1, j-1] + 1$



- $C[i, j]$ は、上記のうち**最小**の値を選べばよい
- 最終的に、 $C[n, m]$ の値が、文字列 $a$ から $b$ への変換に必要な操作の最小回数(=編集距離)

# 編集距離の計算の簡単な例 (1)

変換後

m o d e

変換前

m

o

o

d

0	1	2	3	4
1				
2				
3				
4				

$$\begin{cases} C[i, 0] = i \quad (0 \leq i \leq n) \\ C[0, j] = j \quad (0 \leq j \leq m) \end{cases}$$

# 編集距離の計算の簡単な例 (2)

変換後

	m	o	d	e	
	0	1	2	3	4
m	1				
o	2				
o	3				
d	4				

変換前

文字列 $a$ の $i$ 文字目と文字列 $b$ の $j$ 文字目が一致している場合は  
 $C[i, j] = C[i-1, j-1]$

そうでない場合は以下のうちの最小値:

- ①  $C[i-1, j] + 1$
- ②  $C[i, j-1] + 1$
- ③  $C[i-1, j-1] + 1$

# 編集距離の計算の簡単な例 (3)

変換後

m o d e

m

o

o

d

変換前

0 1 2 3 4

1 0

2

3

4

二つの文字が一致しているため、  
左上の値「0」となる

文字列 $a$ の $i$ 文字目と  
文字列 $b$ の $j$ 文字目が  
**一致している場合**は  
 $C[i, j] = C[i-1, j-1]$

そうでない場合は以  
下のうちの最小値:

- ①  $C[i-1, j] + 1$
- ②  $C[i, j-1] + 1$
- ③  $C[i-1, j-1] + 1$

# 編集距離の計算の簡単な例 (4)

変換後

m o d e

m

変換前

o

o

d

0	1	2	3	4
1	0	1		
2				
3				
4				

二つの文字が一致していないため、  
上・左上・左の値に1を足した値の  
最小値「1」となる

文字列 $a$ の $i$ 文字目と  
文字列 $b$ の $j$ 文字目が  
一致している場合は  
 $C[i, j] = C[i-1, j-1]$

そうでない場合は以  
下のうちの最小値:

- ①  $C[i-1, j] + 1$
- ②  $C[i, j-1] + 1$
- ③  $C[i-1, j-1] + 1$



# 編集距離の計算の簡単な例 (5)

変換後

		m	o	<b>d</b>	e
	0	1	2	3	4
<b>m</b>	1	0	1	2	
o	2				
o	3				
d	4				

変換前

二つの文字が一致していないため、  
上・左上・左の値に1を足した値の  
最小値「2」となる

文字列 $a$ の $i$ 文字目と  
文字列 $b$ の $j$ 文字目が  
一致している場合は  
 $C[i, j] = C[i-1, j-1]$

**そうでない場合**は以  
下のうちの最小値:

- ①  $C[i-1, j] + 1$
- ②  $C[i, j-1] + 1$
- ③  $C[i-1, j-1] + 1$

# 編集距離の計算の簡単な例 (6)

変換後

		m	o	d	e
	0	1	2	3	4
m	1	0	1	2	3
o	2				
o	3				
d	4				

変換前

二つの文字が一致していないため、  
上・左上・左の値に1を足した値の  
最小値「3」となる

文字列 $a$ の $i$ 文字目と  
文字列 $b$ の $j$ 文字目が  
一致している場合は  
 $C[i, j] = C[i-1, j-1]$

**そうでない場合**は以  
下のうちの最小値:

- ①  $C[i-1, j] + 1$
- ②  $C[i, j-1] + 1$
- ③  $C[i-1, j-1] + 1$

# 編集距離の計算の簡単な例 (7)

変換後

m o d e

m

o

o

d

変換前

0	1	2	3	4
1	0	1	2	3
2	1			
3				
4				

二つの文字が一致していないため、  
上・左上・左の値に1を足した値の  
最小値「1」となる

文字列 $a$ の $i$ 文字目と  
文字列 $b$ の $j$ 文字目が  
一致している場合は  
 $C[i, j] = C[i-1, j-1]$

**そうでない場合**は以  
下のうちの最小値:

- ①  $C[i-1, j] + 1$
- ②  $C[i, j-1] + 1$
- ③  $C[i-1, j-1] + 1$

# 編集距離の計算の簡単な例 (8)

変換後

m   o   d   e

m

o

o

d

0	1	2	3	4
1	0	1	2	3
2	1	0		
3				
4				

二つの文字が一致しているため、  
左上の値「0」となる

文字列 $a$ の $i$ 文字目と  
文字列 $b$ の $j$ 文字目が  
**一致している場合**は  
 $C[i, j] = C[i-1, j-1]$

そうでない場合は以  
下のうちの最小値:

- ①  $C[i-1, j] + 1$
- ②  $C[i, j-1] + 1$
- ③  $C[i-1, j-1] + 1$

# 編集距離の計算の簡単な例 (9)

変換後

		m	o	d	e
	0	1	2	3	4
m	1	0	1	2	3
o	2	1	0	1	
o	3				
d	4				

変換前

二つの文字が一致していないため、  
上・左上・左の値に1を足した値の  
最小値「1」となる

文字列 $a$ の $i$ 文字目と  
文字列 $b$ の $j$ 文字目が  
一致している場合は  
 $C[i, j] = C[i-1, j-1]$

**そうでない場合**は以  
下のうちの最小値:

- ①  $C[i-1, j] + 1$
- ②  $C[i, j-1] + 1$
- ③  $C[i-1, j-1] + 1$

# 編集距離の計算の簡単な例 (10)

変換後

		m	o	d	e
	0	1	2	3	4
m	1	0	1	2	3
o	2	1	0	1	2
d	3				
e	4				

変換前

二つの文字が一致していないため、  
上・左上・左の値に1を足した値の  
最小値「2」となる

文字列 $a$ の $i$ 文字目と  
文字列 $b$ の $j$ 文字目が  
一致している場合は  
 $C[i, j] = C[i-1, j-1]$

**そうでない場合**は以  
下のうちの最小値:

- ①  $C[i-1, j] + 1$
- ②  $C[i, j-1] + 1$
- ③  $C[i-1, j-1] + 1$

# 編集距離の計算の簡単な例 (11)

		変換後				
		m	o	d	e	
変換前	m	0	1	2	3	4
	o	1	0	1	2	3
	o	2	1	0	1	2
	o	3	2	1	1	2
	d	4	3	2	1	2

文字列 $a$ の $i$ 文字目と  
文字列 $b$ の $j$ 文字目が  
一致している場合は  
 $C[i, j] = C[i-1, j-1]$

そうでない場合は以  
下のうちの最小値:

- ①  $C[i-1, j] + 1$
- ②  $C[i, j-1] + 1$
- ③  $C[i-1, j-1] + 1$

同様に計算していくと、上記のようになる

# 編集距離の計算の簡単な例 (12)

変換後

m o d e

変換前

m

o

o

d

0	1	2	3	4
1	0	1	2	3
2	1	0	1	2
3	2	1	1	2
4	3	2	1	2

置換

置換

文字列 $a$ の $i$ 文字目と文字列 $b$ の $j$ 文字目が一致している場合は  
 $C[i, j] = C[i-1, j-1]$

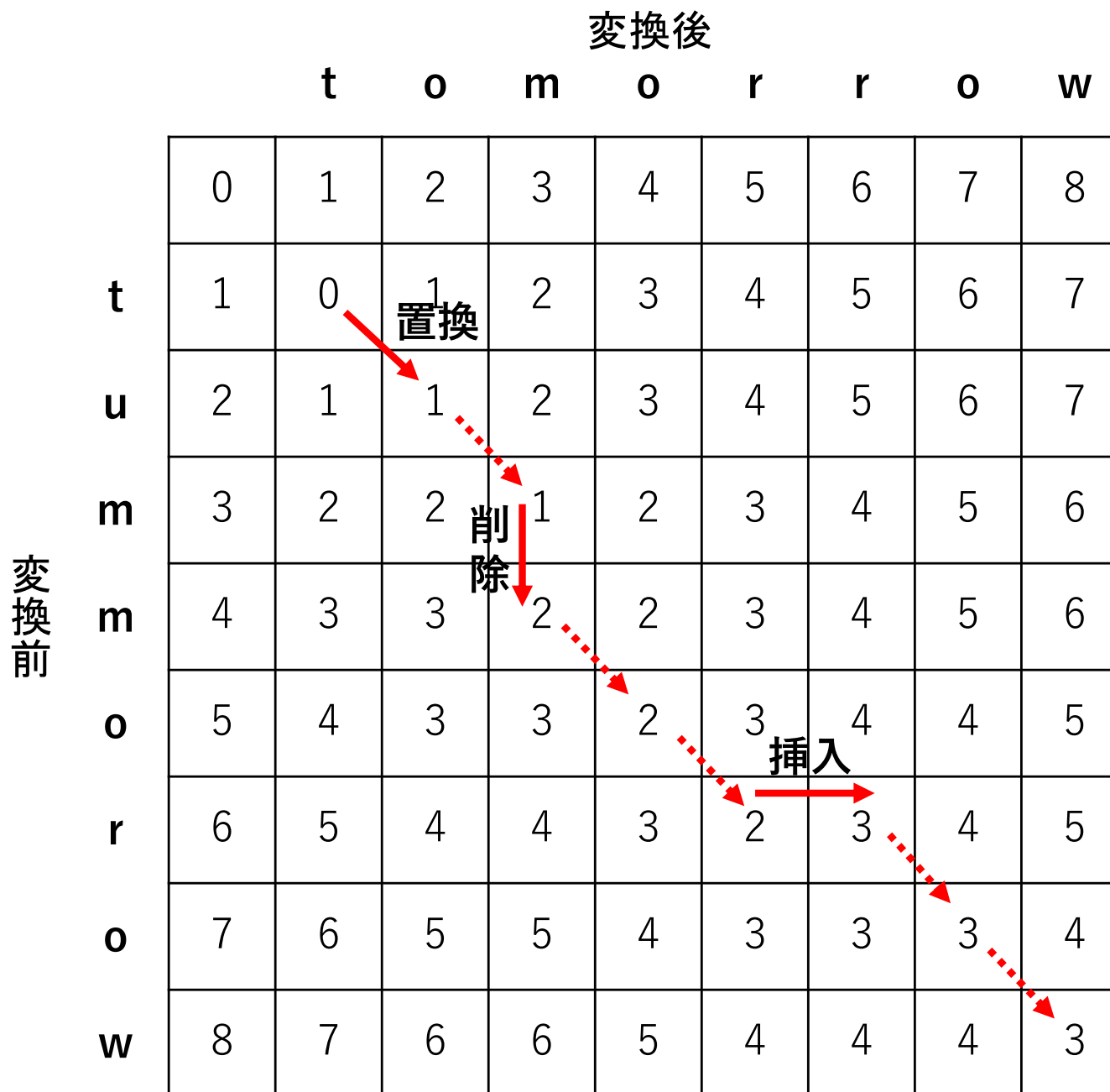
そうでない場合は以下のうちの最小値:

- ①  $C[i-1, j] + 1$
- ②  $C[i, j-1] + 1$
- ③  $C[i-1, j-1] + 1$

$C[n, m]$ の値(=2)が編集距離



# 編集距離の計算結果と、操作が最小となる経路の例





# 適合性フィードバック

- 一回の検索では、利用者が必要とする情報が得られない場合が多い
- 検索結果によってパラメータを調整（フィードバック）し、再度検索する手法を**適合性フィードバック**と呼ぶ
  - 利用者が必要とする情報がより上位に来るようになるため、**検索精度**の向上に役立つ

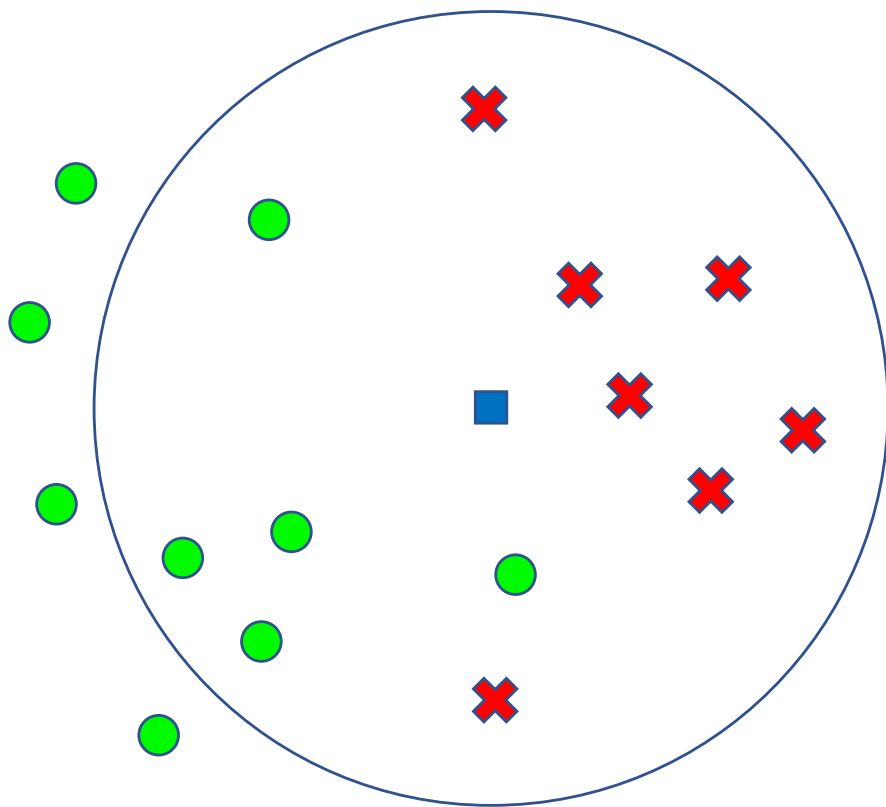


## 手動の適合性フィードバック

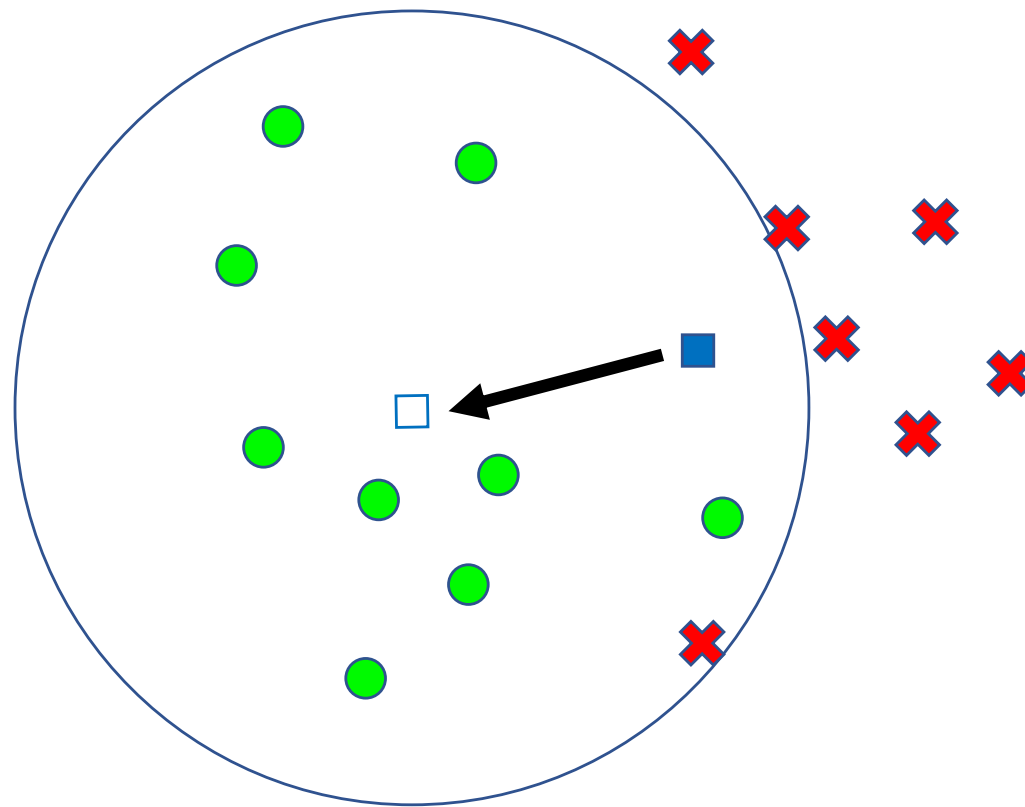
- システムが、利用者の問合せに対して検索結果(上位  $n$  件)を返す
- 利用者は、検索結果(上位  $n$  件)の各文書を**適合文書**と**不適合文書**に分ける
  - もしくは適合文書のみを指定する
- システムは、以下のように問合せを修正
  - 利用者が指定した**適合文書**に含まれる語の重みを大きくする
  - **不適合文書**に含まれる語の重みを小さくする

# 適合性フィードバックによる 問合せの修正のイメージ

元の問合せによる検索結果



修正された問合せによる検索結果



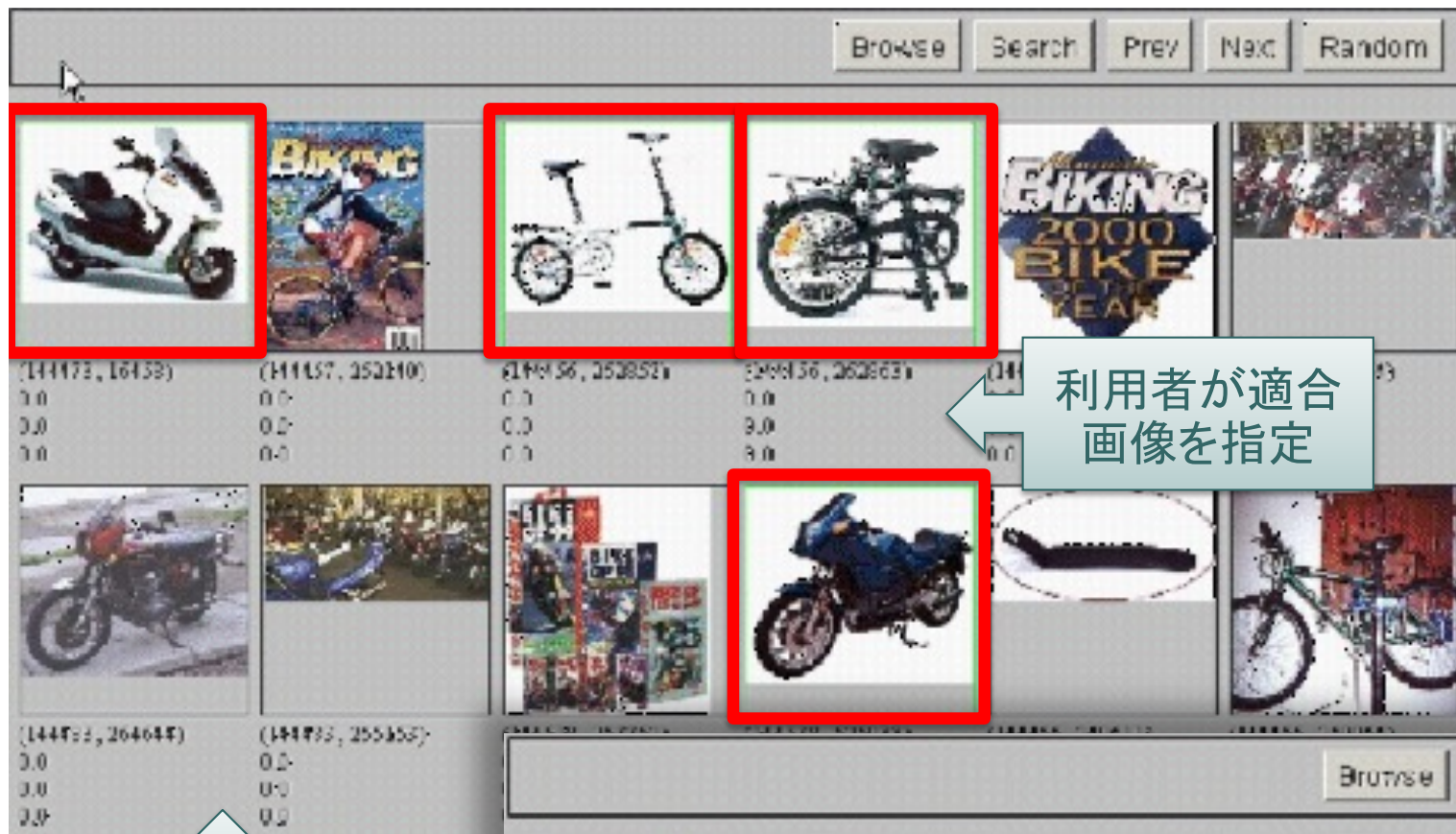
■ 元の問合せ

□ 修正された問合せ

● 適合と判断された文書

✕ 不適合と判断された文書

# 画像検索における適合性フィードバックの例



最初の検索結果

修正された  
問合せによる  
検索結果





# 自動適合性フィードバック

- 「検索結果の上位文書の多くは適合文書である」と仮定
  - 検索結果の上位  $n$  件のベクトルをすべて**適合文書**として、自動的に適合性フィードバックを適用
  - **不適合文書**はないものとする
- **利点**: 利用者が自分で選ぶ必要がない
- **欠点**: 必ずしも上位の文書が適合文書とは限らない



# 適合性フィードバックの課題

- **手動**の場合は、利用者に手間がかかる
  - 利用者自身が検索結果を見て**適合**／**不適合**を判断し、再検索するため
  - 一般的な利用者は、そこまで手間をかけたくない
- **自動**の場合は、必ずしも検索精度が向上しない
  - 利用者の問合せが適切であれば向上するが、不適切な場合は逆に低下



# 検索結果の出カインタフェース

- 現在の情報検索システムでは、検索結果の出カはリスト形式がほとんど
- たとえばWeb検索エンジンでは、各ページについて以下の情報を表示
  - ページのタイトル
  - ページ内容の抜粋(スニペット)
  - ページのURL
  - その他(最終更新日時, アイコンなど)





# 検索結果のスニペット

- 検索結果の内容を素早く把握するために、内容の短い**要約**(抜粋)を表示
- 問合せに関係なく文書の要約を行う**静的要約**
  - 伝統的な**自動要約**の技術が用いられる
- 問合せに応じて要約を行う**動的要約**
  - 検索語が含まれる前後の抜粋を表示
  - その文書がなぜヒットしたかが一目でわかる

約 526,000 件 (0.65 秒)

他のキーワード: [立命館大学合格発表](#)

## 立命館大学 - Beyond Borders -

[www.ritsumei.ac.jp/](http://www.ritsumei.ac.jp/) ▼

立命館大学 公式Webサイト。受験情報、学部、学科、大学院、就職、学生生活、研究活動紹介。

ritsumei.ac.jp からの検索結果



### Campus Web

Campus Web - Web MAIL - 立命館大学 manaba+R - 情報理工学部

### 受験生の方

大学院受験生の方へ. 大学院受験生のための入試サイト. ページトップ ...

### 大学紹介

大学紹介. 未来を創る取り組み; NEWS & TOPICS; 理念・歴史 ...

### オンラインシラバス

授業アンケートの結果を学内向けオンラインシラバスにアップしています。

### Web MAIL

2016年度より、本学のメールシステムが現在のCyberMailから ...

### 教育

教育. NEWS & TOPICS; EVENTS; 学びの立命館モデル; 学部; 大学 ...

## 立命館大学 - Wikipedia

<https://ja.wikipedia.org/wiki/立命館大学> ▼

立命館大学（りつめいかんだいがく、英語: Ritsumeikan University）は、京都府京都市中京区西ノ京朱雀町1に本部を置く日本の私立大学である。1922年に設置された。大学の略称は立命、立命館、立命大。近畿圏では立大も使用される（立教大と混同され ...

## 立命館大学 入試情報サイト

[ritsnet.ritsumei.jp/](http://ritsnet.ritsumei.jp/) ▼

立命館大学受験生のための入試情報サイト。各学部の入試情報ははじめ入学試験結果や入試に関するQ&A、イベント情報など受験や入試に役立つ情報を総合的にお伝えします。



## 立命館大学

[ウェブサイト](#)

[経路案内](#)

りつめいかんだいがく  
日本 京都市の私立大学

立命館大学は、京都府京都市中京区西ノ京朱雀町1に本部を置く日本の私立大学である。1922年に設置された。大学の略称は立命、立命館、立命大。近畿圏では立大も使用される。地理院地図 Googleマップ 立命館大学 ウィキパディア

所在地: 〒603-8577 京都府京都市北区等持院北町 5 6 - 1

在籍: 35,113 (2015年5月1日)

創立者: [西園寺公望](#)

設立: 1922年

合格率: 36% (2015年)

[情報の修正を提案](#)

### 著名な卒業生

他 25 件以上を表示



### 通常のスニペット(ページから抜粋)



西尾維新

倉木麻衣

岡村 隆史

小山 力也

棚橋弘至

### クチコミ

Google のクチコミ (54)

[レビューを書く](#)

[写真を追加](#)

他の人はこちらも検索

他 10 件以上を表示

metaタグのdescription  
属性の内容



京都市



すべて

地図

ニュース

画像

ショッピング

もっと見る

設定

ツール

約 49,300,000 件 (0.84 秒)

京都市：トップページ

[www.city.kyoto.lg.jp/](http://www.city.kyoto.lg.jp/) ▼

京都市の公式サイト。市政についてのニュースやバスや地下鉄の時刻表。

metaタグのdescription  
属性の内容

京都観光Navi

桜だより - 桜ライトアップ - 紅葉だより - 嵐山 - ...

健康・福祉・教育

健康・福祉・教育. 「子育て支援総合電話窓口（はぐくみコール）」を設置 ...

暮らしの情報

暮らしの情報. 戸籍のコンピュータ化の取組・マイナンバー（個人番号 ...

市政情報

広報資料・お知らせ・府市協調・京都都市圏自治体ネットワーク・広域計 ...

観光・文化・産業

観光・文化・産業. 安心して京都観光を楽しんでいただくための情報・民 ...

申請書等の一覧

申請書等の一覧 ... 給与・人材育成; 申請書等の一覧; 補助金・助成金 ...

← サイトリンク

[kyoto.lg.jp](http://kyoto.lg.jp) からの検索結果 »

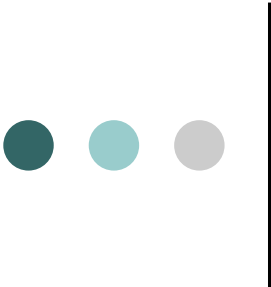
京都市 - Wikipedia

<https://ja.wikipedia.org/wiki/京都市> ▼

通常のスニペット（ページから抜粋）

地理院地図 Googleマップ Bing GeoHack・MapFan Mapion Yahoo! NAVITIME ゼンリン. 特記事項, 市章:1960年（昭和35年）1月1日制定略章:1891年（明治24年）10月2日制定. 表示 ウィキプロジェクト ...

京都府・愛宕山・北区・高野川



# 問合せ処理・インタフェースの まとめ

- 問合せを改良する手法として、**問合せ拡張**や**適合性フィードバック**などがある
- 問合せ中の誤字を修正する**スペル修正**では、文字の類似度を表す**編集距離**が用いられる
- 検索結果から必要な情報を見つけやすくするために文書の抜粋(**スニペット**)が用いられる