



# 情報アクセス論 第4回

## 「索引付け」

情報理工学部  
前田 亮

# 第3回小テストの解説

- 漢字「情」(符号位置: 60C5)をUTF-8に変換せよ.  
変換途中の2進数表記も示すこと.

文字:

情

↓  
符号位置(16進数):

60 C5

↓  
符号位置(2進数):

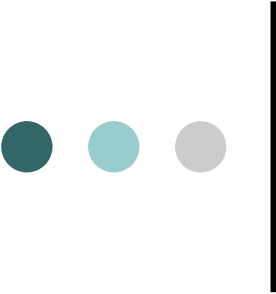
0110 0000 1100 0101

↓  
UTF-8(2進数):

11100110 10000011 10000101

↓  
UTF-8(16進数):

E6 83 85



# 索引付けとは？

## ○ 用語の復習

- **索引付け**: 文書中から**索引語**を抽出し, 索引を作成する処理
- **索引**: 検索を高速化するために, あらかじめ文書集合から作っておくデータ構造
- **索引語**: 文書の内容を特徴付ける重要な語

## ○ 検索が行われる前に, システムが扱う文書集合に対して事前に行っておく必要がある

- 文書が追加・削除・更新されれば, それに合わせて索引も更新が必要

# 索引付けの方法

- **昔は人手で行われていた**（本の索引と同じ）
  - 重要語を正確に取り出せるが、作業コストが膨大
  - 抽出した単語でしか検索できない
- **現在では、文書から自動的に索引語を抽出するのが主流**
  - 本文中のすべての単語を用いることで、**全文検索**が実現できる
  - ただし、単純にすべての単語を抽出すると問題
    - 検索の役に立たない語がある
    - 単語によって重要度は異なる
  - 不要語の除去、索引語の重み付けなどを行う

# 索引付けの手順(1)

## 語の切り出し

- 英語などでは空白・記号で区切れば良い
- 日本語では語の区切りが無いため、**形態素解析**を行って切り出す

文

この条約の締約国は、地球の気候の変動及びその悪影響が人類の共通の関心事であることを確認し、...



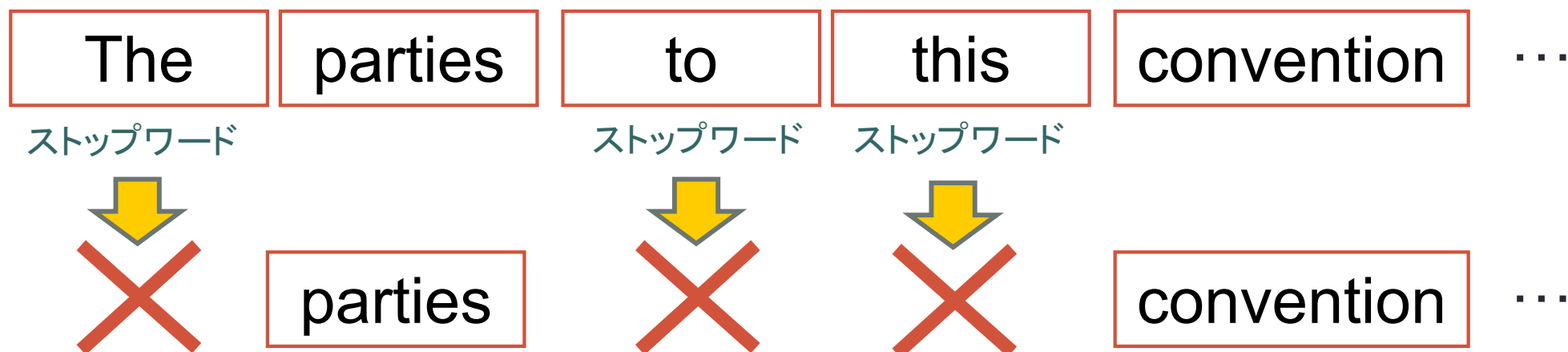
形態素の列

この	条約	の	締約	国	.....
連体詞	名詞・一般	助詞・連体化	名詞・サ変接続	名詞・接尾	

# 索引付けの手順(2)

## 不要語の除去(英語)

- 前置詞・冠詞・接続詞など  
(a, an, and, be, for, from, in, the, to, with...)
  - 「ストップワード(stop words)」と呼ばれる



# 索引付けの手順(2)

## 不要語の除去(日本語)

- 機能語(助詞・助動詞・代名詞・連体詞など)
  - 形態素解析によって品詞がわかる



# 索引付けの手順(3)

## ○ 接辞処理(**stemming**, ステミング)

- 語形の多様性を正規化する処理
  - 日本語の場合は不要
- play, plays, played, playing, player → play
- 代表的なものとして, Porterの手法がある
  - 処理前: The parties to this convention, acknowledging
  - 処理後: The parti to thi convent acknowledg
- 正規化によって意味が変わってしまう場合も
  - fish: 魚, fishing: 魚釣り・漁業, fisher: 漁夫





# 索引語の重み付け

- 抽出した索引語が、文書の内容にどれだけ密接に関係しているかを表す**重要度**を付与
  - 検索結果のランキングが可能になる
- 最も単純な重み付けの方法は、索引語の文書中での出現頻度（回数）
  - 一般的な語ほど出現頻度が高くなってしまう
  - 文書の長さに影響される

# 索引語頻度 (TF: Term Frequency)

- 索引語  $t$  が, ある文書  $d$  中に出現する回数  $tf(t, d)$ 
  - 「文書中で何度も繰り返し言及される語は重要である」という仮定に基づく
- TFには, さまざまなバリエーションがある
  - 最も単純には, 単語の出現回数そのもの
  - 出現頻度が高い語の影響を軽減するために, 出現頻度の対数を取る方法

$$TF(t, d) = \begin{cases} 1 + \log(tf(t, d)) & (tf(t, d) > 0) \\ 0 & (tf(t, d) = 0) \end{cases}$$



# 索引語頻度 (索引語の総数による正規化)

- 文書の長さに影響されないように, 文書中の索引語の総数で正規化
  - たとえば1,000単語からなる文書中に10回出現する索引語と, 10,000単語からなる文書中に10回出現する索引語では, 重要度が異なる

$$TF(t, d) = \frac{tf(t, d)}{\sum_{s \in d} tf(s, d)}$$



# 索引語頻度 (最頻語による正規化)

- 文書中で最も出現頻度が高い語の頻度を用いて正規化

$$TF(t, d) = \begin{cases} 0.5 + 0.5 \frac{tf(t, d)}{\max_{t'} tf(t', d)} & (tf(t, d) > 0) \\ 0 & (tf(t, d) = 0) \end{cases}$$

教科書の式(4.3)(p.34)

# 逆文書頻度

(**IDF**: Inverse Document Frequency)

- ある索引語が全文書中のどれだけの文書に出現するかを測る尺度
  - 「どの文書にも出現する語は、文書の特徴付けるのには役に立たない」
  - 「より少ない文書に出現する語は、その文書をより特徴付けている」

$$IDF(t) = \log \frac{N}{df(t)}$$

$N$ : 文書集合の全文書数  
 $df(t)$ : 索引語  $t$  が出現する文書数

教科書の式(4.4)(p.34)

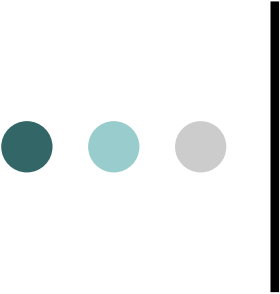
- 少数の文書にしか出現しない語の重みは大きくなり、どの文書にも出現する語の重みは小さくなる

# ● ● ● | 逆文書頻度のバリエーション

- 前ページの式では, すべての文書に出現する語の重みが0になってしまう
  - これを防ぐために, IDFに1を加える

$$IDF(t) = \log \left( \frac{N}{df(t)} + 1 \right)$$

教科書の式(4.5) (p.34)



# 索引語の重み付けの例: **TF・IDF**

- 索引語頻度 (TF) と逆文書頻度 (IDF) を掛け合わせる

$$TF \cdot IDF(t, d) = TF(t, d) \cdot IDF(t)$$

教科書の式 (4.6) (p.34)

- 「より少数の文書で、より多く言及される語は重要である」
- 現在の情報検索システムでは、索引語の重み付けに TF・IDF が最もよく使われる
  - TF/IDF, tf.idf などと表記される場合もある

# TF・IDFの計算例(1)

## 教科書第4章の演習問題4.1

- 右の表のように, 各文書における索引語とその出現頻度が与えられているものとする
- この時, 文書 $d_2$ における索引語 $t_2$ のTF・IDF値を求めよ

	$d_1$	$d_2$	$d_3$
$t_1$	1	0	1
$t_2$	1	2	0
$t_3$	0	1	1



## TF・IDFの計算例(2)

- TFに式(4.2), IDFに式(4.4)を用いる場合:

	$d_1$	$d_2$	$d_3$
$t_1$	1	0	1
$t_2$	1	2	0
$t_3$	0	1	1

$$TF \cdot IDF(t, d) = \frac{tf(t, d)}{\sum_{s \in d} tf(s, d)} \cdot \log \frac{N}{df(t)}$$

式(4.2)                      式(4.4)

$$\begin{aligned} & TF \cdot IDF(t_2, d_2) \\ &= \frac{tf(t_2, d_2)}{tf(t_1, d_2) + tf(t_2, d_2) + tf(t_3, d_2)} \cdot \log \frac{3}{2} \\ &= \frac{2}{0 + 2 + 1} \cdot 0.6 = \frac{2}{3} \cdot 0.6 = 0.4 \end{aligned}$$

$\log(3/2)=0.6$   
を用いる



## TF・IDFの計算例(3)

- 同様に計算すると:

	$d_1$	$d_2$	$d_3$
$t_1$	1	0	1
$t_2$	1	2	0
$t_3$	0	1	1

$$\begin{aligned} & TF \cdot IDF(t_3, d_2) \\ &= \frac{tf(t_3, d_2)}{tf(t_1, d_2) + tf(t_2, d_2) + tf(t_3, d_2)} \cdot \log \frac{3}{2} \\ &= \frac{1}{0 + 2 + 1} \cdot 0.6 = \frac{1}{3} \cdot 0.6 = 0.2 \end{aligned}$$

$$\begin{aligned} & TF \cdot IDF(t_2, d_1) \\ &= \frac{1}{1 + 1 + 0} \cdot 0.6 = \frac{1}{2} \cdot 0.6 = 0.3 \end{aligned}$$



## TF・IDFの計算例(4)

- TFに式(4.1), IDFに式(4.4)を用いる場合:

	$d_1$	$d_2$	$d_3$
$t_1$	1	0	1
$t_2$	1	2	0
$t_3$	0	1	1

$$TF \cdot IDF(t, d) = (1 + \log(tf(t, d))) \cdot \log \frac{N}{df(t)}$$

式(4.1)  
( $tf(t, d) > 0$ の場合)      式(4.4)

$$\begin{aligned} TF \cdot IDF(t_2, d_2) &= (1 + \log(tf(t_2, d_2))) \cdot \log \frac{3}{2} \\ &= (1 + \log(2)) \cdot 0.6 = 2 \cdot 0.6 = 1.2 \end{aligned}$$

log(2)=1  
を用いる



## TF・IDFの計算例(5)

- 同様に計算すると:

	$d_1$	$d_2$	$d_3$
$t_1$	1	0	1
$t_2$	1	2	0
$t_3$	0	1	1

$$\begin{aligned} TF \cdot IDF(t_3, d_2) &= (1 + \log(tf(t_3, d_2))) \cdot \log \frac{3}{2} \\ &= (1 + \log(1)) \cdot 0.6 = 1 \cdot 0.6 = 0.6 \end{aligned}$$

$$\begin{aligned} TF \cdot IDF(t_2, d_1) &= (1 + \log(tf(t_2, d_1))) \cdot \log \frac{3}{2} \\ &= (1 + \log(1)) \cdot 0.6 = 1 \cdot 0.6 = 0.6 \end{aligned}$$

- もし $tf(t, d)$ が3であれば,  $TF \cdot IDF$ は約1.55となる



## TF・IDFの計算例(6)

- TFに式(4.3), IDFに式(4.4)を用い場合:

	$d_1$	$d_2$	$d_3$
$t_1$	1	0	1
$t_2$	1	2	0
$t_3$	0	1	1

$$TF \cdot IDF(t, d) = \underbrace{\left(0.5 + 0.5 \frac{tf(t, d)}{\max_{t'} tf(t', d)}\right)}_{\substack{\text{式(4.3)} \\ (tf(t,d)>0\text{の場合})}} \cdot \underbrace{\log \frac{N}{df(t)}}_{\text{式(4.4)}}$$

$$\begin{aligned} TF \cdot IDF(t_2, d_2) &= \left(0.5 + 0.5 \frac{tf(t_2, d_2)}{\max_{t'} tf(t', d_2)}\right) \cdot \log \frac{3}{2} \\ &= \left(0.5 + 0.5 \cdot \frac{2}{2}\right) \cdot 0.6 = 1 \cdot 0.6 = 0.6 \end{aligned}$$



## TF・IDFの計算例(7)

- 同様に計算すると:

	$d_1$	$d_2$	$d_3$
$t_1$	1	0	1
$t_2$	1	2	0
$t_3$	0	1	1

$$\begin{aligned} TF \cdot IDF(t_3, d_2) &= \left(0.5 + 0.5 \frac{tf(t_3, d_2)}{\max_{t'} tf(t', d_2)}\right) \cdot \log \frac{3}{2} \\ &= \left(0.5 + 0.5 \cdot \frac{1}{2}\right) \cdot 0.6 = 0.75 \cdot 0.6 = 0.45 \end{aligned}$$

$$\begin{aligned} TF \cdot IDF(t_2, d_1) &= \left(0.5 + 0.5 \frac{tf(t_2, d_1)}{\max_{t'} tf(t', d_1)}\right) \cdot \log \frac{3}{2} \\ &= \left(0.5 + 0.5 \cdot \frac{1}{1}\right) \cdot 0.6 = 1 \cdot 0.6 = 0.6 \end{aligned}$$



# 索引のデータ構造

- 書籍の索引では, ある用語が**どのページ**に記載されているかを特定できる
- 文書の検索では, 入力された検索語が含まれた文書を探す
  - 検索語が**どの文書**に含まれるかという情報が必要
- **転置索引**(inverted index)
  - 与えられた単語が含まれる文書IDを取得するためのデータ構造

# 転置索引作成の流れ (文書IDのみ)

## 索引付けの対象文書

文書1	文書2	文書3
情報検索 とは、情 報システ ムの...	イベント情 報およびイ ベント会場 は...	このサッ カー大会 の試合会 場は...

「どの索引語がどの文書に含まれるか」という情報を格納

例:「情報」という索引語は  
文書1と文書2に出現する

「情報 1,2」

## 索引語 文書ID

情報	1,2
検索	1
システム	1
イベント	2
会場	2,3
サッカー	3
試合	3

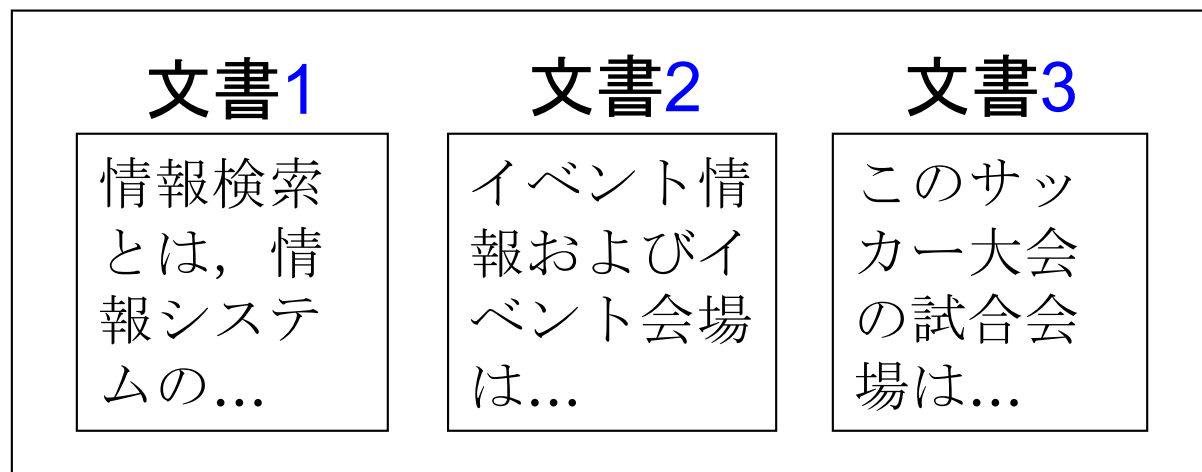
転置索引の例  
(文書IDのみを格納)



# 転置索引の作成の流れ

## (文書IDと出現頻度)

### 索引付けの対象文書



「どの索引語がどの文書に**何回**  
含まれるか」という情報を格納

例:「情報」という索引語は  
文書1に2回出現する

「情報 1:2」

文書ID

出現頻度

ポスティングリスト

索引語 文書ID:出現頻度

情報	1:2,2:1
検索	1:1
システム	1:1
イベント	2:2
会場	2:1,3:1
サッカー	3:1
試合	3:1

転置索引  
(文書IDおよび  
出現頻度を格納)

# 索引付けの実際

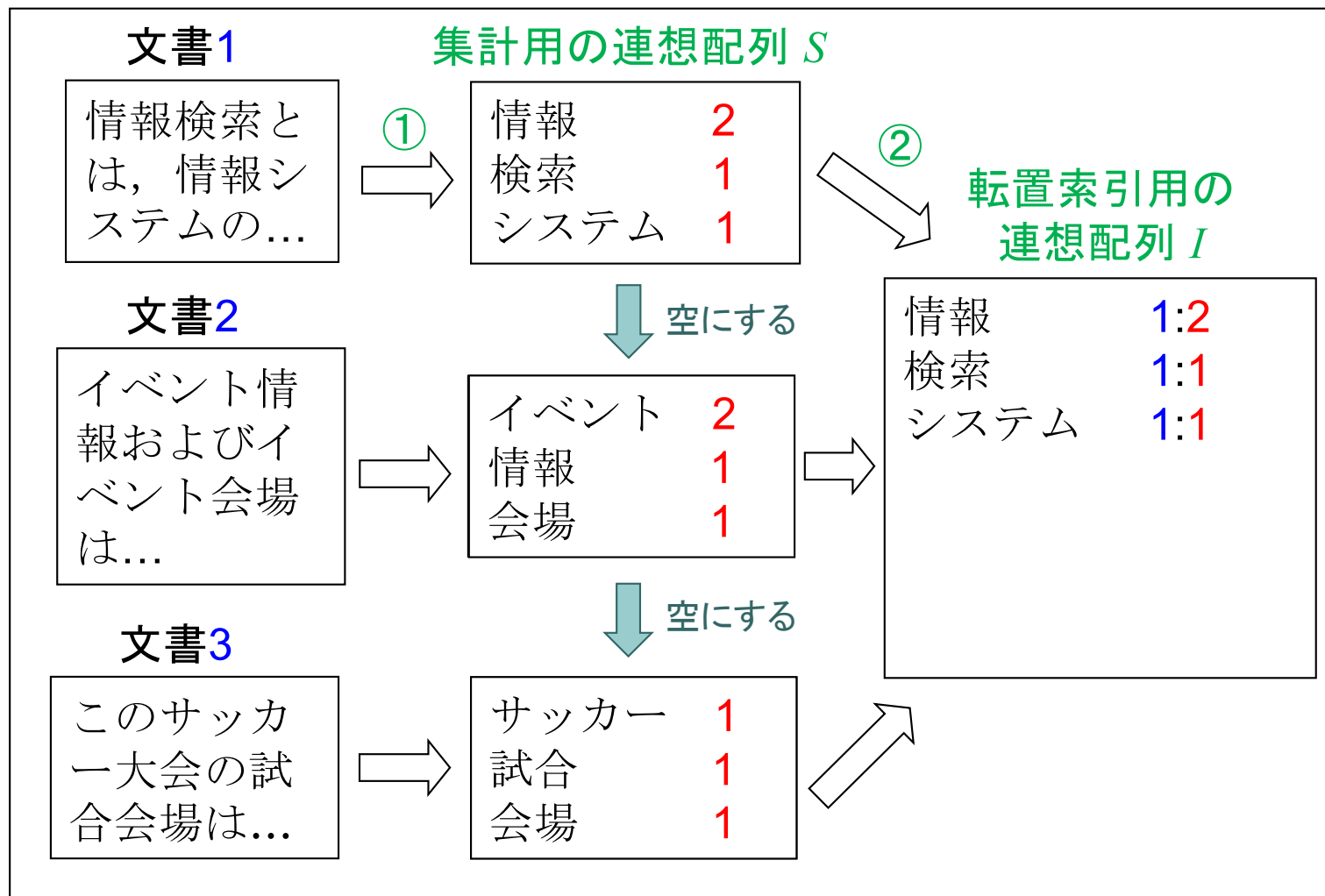
## ○ 転置索引の作成アルゴリズムの例

1. 集計用の連想配列  $S$  と、転置索引用の連想配列  $I$  をそれぞれ用意
2. 索引付け対象の文書集合から1件を取り出す
3. 取り出した文書について、各索引語の出現頻度を  $S$  に集計する ... 次ページ図の①
4. 集計した結果  $S$  の内容を  $I$  に追加する ... 次ページ図の②
5.  $S$  の内容を空にする
6. すべての文書の処理が終わるまで、上記2～5の手順を繰り返す

**連想配列**: 添字として数値ではなく文字列などを使用できる配列  
(キーと値の組を格納, Pythonではデータ型「辞書」が相当)

# 転置索引の作成手順

※集計用の連想配列  $S$  と転置索引用の連想配列  $I$  の2つの連想配列を用いる



① 文書ごとに単語の出現頻度の集計を行い,  $S$  に格納

②  $S$  の内容を  $I$  に格納

③ 文書の数だけ①, ②を繰り返す

# 索引を用いた検索(1)

情報	1:2,2:1
検索	1:1
システム	1:1
イベント	2:2
会場	2:1,3:1
サッカー	3:1
試合	3:1

- 転置索引を用いることで、利用者が入力した問合せに対して検索を行うことができる

- 問合せ「会場」を入力



文書2, 文書3を検索結果として返す

- 問合せ「情報 AND イベント」を入力



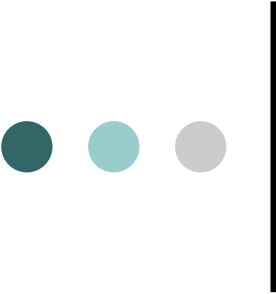
文書2を検索結果として返す

- 問合せ「情報 OR 会場」を入力



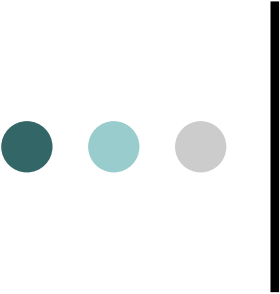
文書1, 文書2, 文書3を検索結果として返す

ブーリアン  
モデルに  
よる検索  
(第5回で  
説明)



## 索引を用いた検索(2)

- 通常の検索システムでは、索引語の数は**数万種類**以上になる場合が多い
  - 扱う文書の数が増えるほど、使われる単語の種類が増えるため
- **連想配列**を用いることで、索引語が含まれる文書を高速に見つけることが可能
  - 連想配列は、内部的にはハッシュテーブルや平衡二分探索木で実装され、キーに対応する値を高速に参照できる



# 索引付けのまとめ

- 検索を高速に行うために、文書集合に対してあらかじめ行っておく重要な処理
  - 語の切り出し, 不要語の除去, 接辞処理, 重み付け, 転置索引の作成, etc.
- 索引語の重み付けには**TF·IDF**などの尺度が用いられる
- 索引のデータ構造としては**転置索引**が用いられる