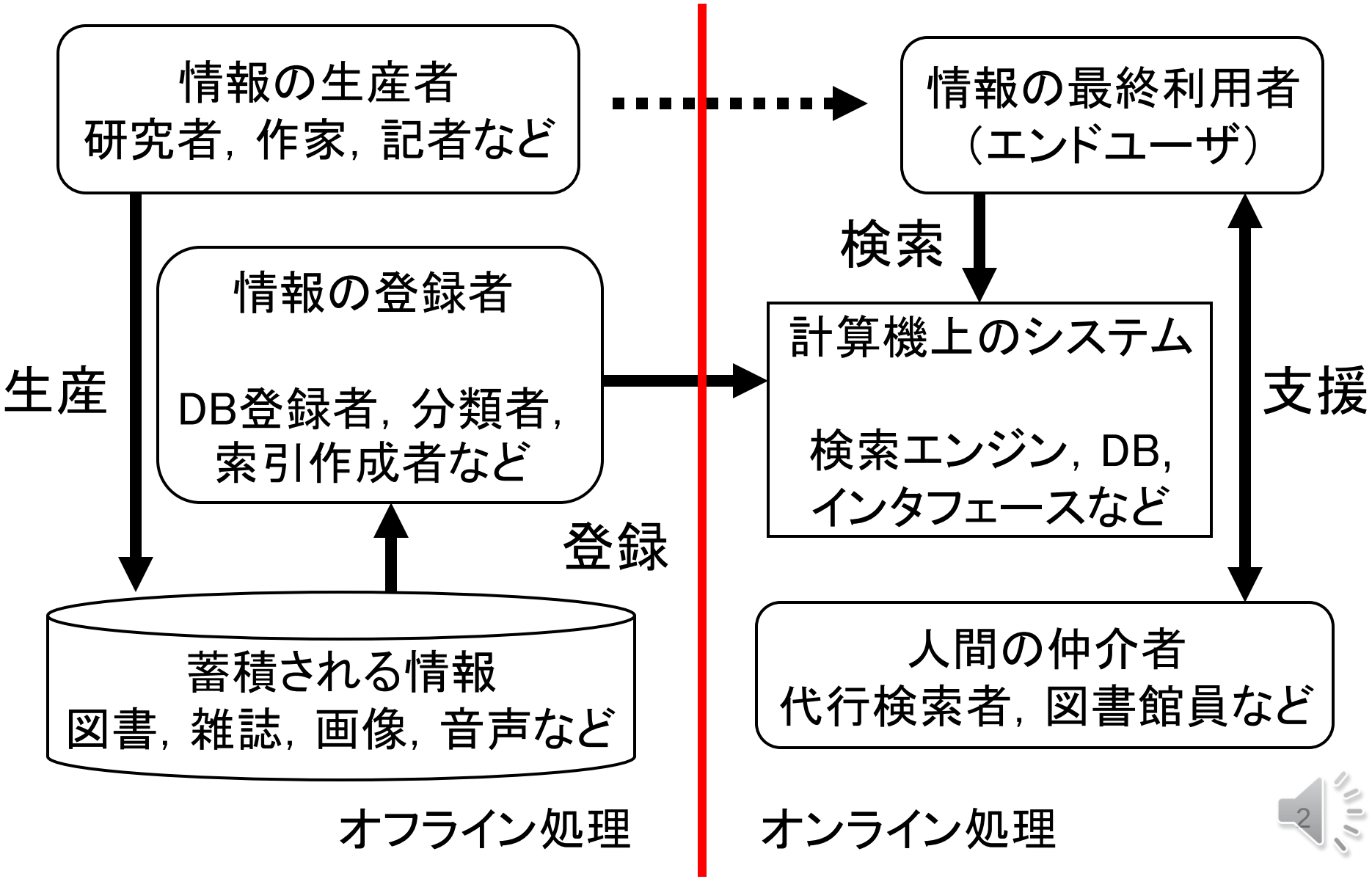
知識情報演習Ⅲ（前半第2回）

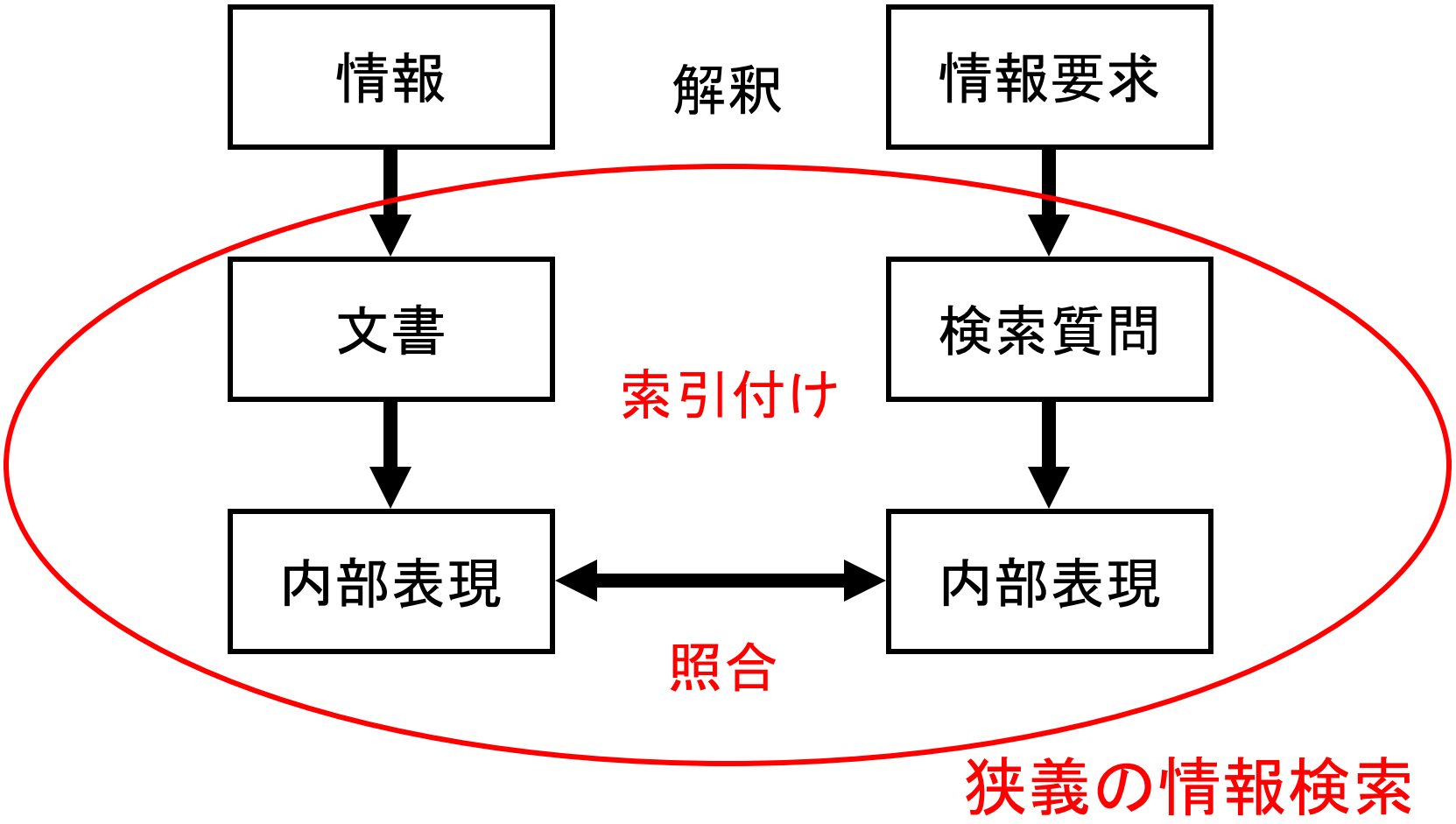
辻　慶太

http://slis.sakura.ne.jp/cje3

情報検索システムの世界観



情報検索の基本モデル

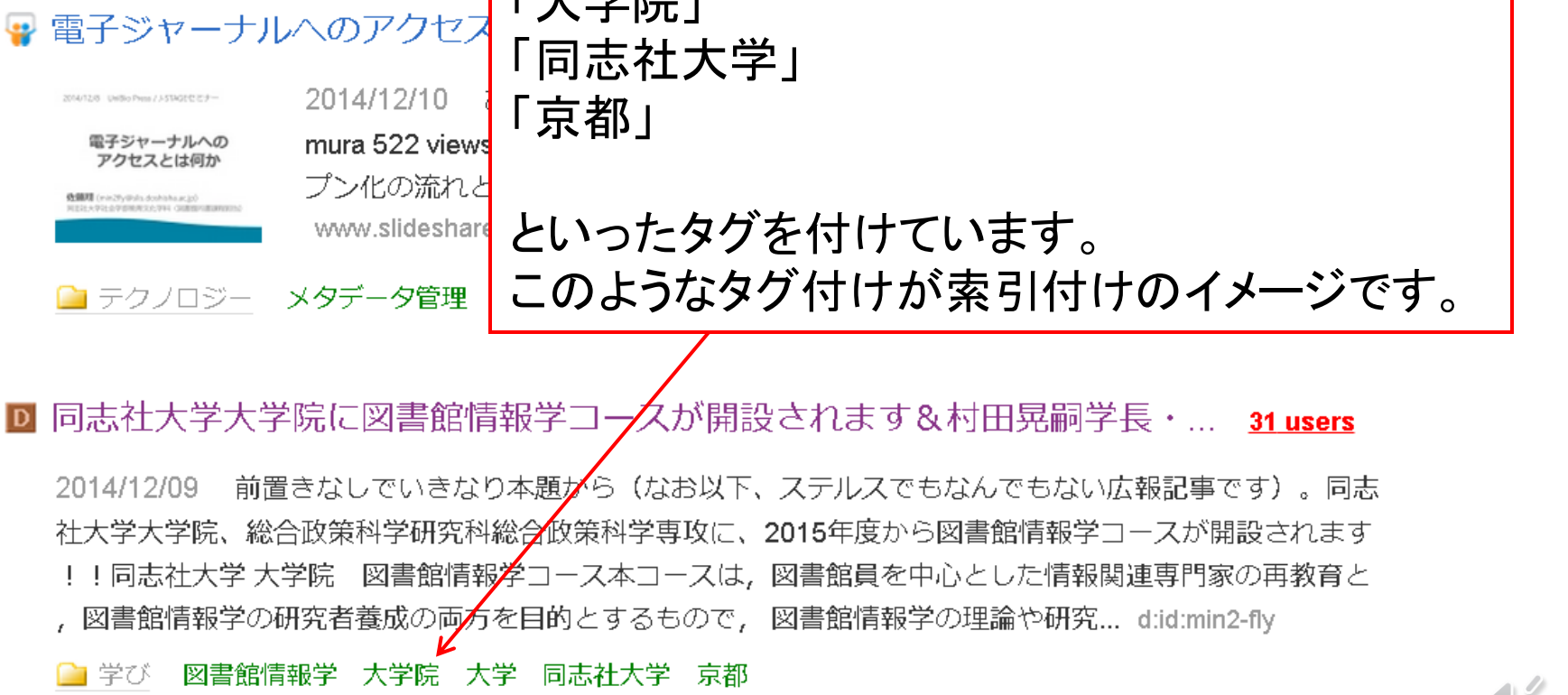


※索引付け？　→ブックマークでタグを付けるようなイメージ

「同志社大学大学院に図書館情報学コースが開設されます」というページに対して，この人は：

「図書館情報学」
「大学院」
「同志社大学」
「京都」

といったタグを付けています。
このようなタグ付けが索引付けのイメージです。



照合

- 文書の索引語と，検索質問の索引語を比較し，一致するものや似たものを特定すること。

→「図書館」というキーワードで検索してくる人がいたら，
「図書館」という索引語が付与された文書がないか探す。

- 2つのモデル（方法）に大別することができる。
  - 完全一致（exact match）
  - 最良一致（best match）

→「図書館」という索引語が付与された文書だけを出力する。

→「図書館」という索引語が付与されていなくても，何となく図書館に関する文書と判断できるならば出力する。

完全一致

- ブーリアンモデルが代表的
  - 古典的なキーワード検索
- 論理演算子（AND，OR，NOT）で式を構成
  - 例: 中華料理 AND レシピ NOT スープ
- 論理式に一致する文書だけが検索される
- ただし，厳密なNOTではないことが多い
  - 絞込み情報としての利用が中心
  - 例：NOT 犬 →「犬」を含まない文書が全て出るわけではない

照合

- 文書の索引語と，検索質問の索引語を比較し，一致するものや似たものを特定すること。

  →「図書館」というキーワードで検索してくる人がいたら，「図書館」という索引語が付与された文書がないか探す。

- 2つのモデル（方法）に大別することができる。
  - 完全一致（exact match）
  - 最良一致（best match）

→「図書館」という索引語が付与された文書だけを出力する。

→「図書館」という索引語が付与されていなくても，何となく図書館に関する文書と判断できるならば出力する。

最良一致の代表的なモデル

- ベクトル空間モデル
  システムの例：SMART
- 確率型モデル
  システムの例：OKAPI

- どちらのモデルも1970年代に提案され，現在も改良が重ねられている
  – 両モデルの検索精度に大きな違いはない

最良一致の代表的なモデル

- ベクトル空間モデル
  システムの例：SMART
- 確率型モデル
  システムの例：OKAPI

→ Gerald Salton が提案。

- どちらのモデルも1970年代に提案され，現在も改良が重ねられている
  – 両モデルの検索精度に大きな違いはない

最良一致の代表的なモデル

- ベクトル空間モデル
  システムの例：SMART
- 確率型モデル
  システムの例：OKAPI

→ Stephen Robertson が提案。
OKAPI BM25 の“BM”は
文字通り“Best Match”（最良一致）の略。

- どちらのモデルも1970年代に提案され，現在も改良が重ねられている
  – 両モデルの検索精度に大きな違いはない

索引付けの手順概要

(1) 索引語の抽出

文字バイグラム，単語，フレーズなど

(2) 不要語の削除

(3) 接辞処理

(4) 索引語の重み付け

検索手法（検索モデル）によっては不要

例えば，論理式によるブーリアンモデルでは不要

(5) 索引ファイルの編成

「図書館システム」からバイグラムを
切り出すと「図書」「書館」「館シ」「シス」…

索引付けの手順概要

(1) 索引語の抽出

　文字バイグラム，単語，フレーズなど

(2) 不要語の削除

(3) 接辞処理

(4) 索引語の重み付け

　検索手法（検索モデル）によっては不要

　例えば，論理式によるブーリアンモデルでは不要

(5) 索引ファイルの編成

不要語（stopword）

- 検索の役に立たない語（they, might など）
- 不要語辞書を用意しておくことが多い
  - 高頻度語：「研究」など
  - 機能語：「前置詞（of）」など
- 語の分類
  - 内容語：名詞，動詞，形容詞など
  - 機能語：助詞，助動詞，冠詞，前置詞など

索引付けの手順概要

(1) 索引語の抽出

　文字バイグラム，単語，フレーズなど

(2) 不要語の削除

(3) 接辞処理

(4) 索引語の重み付け

　検索手法（検索モデル）によっては不要

　例えば，論理式によるブーリアンモデルでは不要

(5) 索引ファイルの編成

接辞処理（stemming）

- 活用形を原形に戻し，索引語の表記を統一
  - 質問と文書における表記の違いを吸収
- いくつかの手法がある
  - 辞書の利用
  - 語尾の自動削除

“libraries”という表記で検索してきた人に対しては“library”という表記で索引付けされている文献も出力したい。

- 自動削除の場合は，必ずしも言語学的に意味のある単位ではない点に注意

  例：facility（単数形），facilities（複数形）
  - どちらも facilit になるかもしれない

索引付けの手順概要

(1) 索引語の抽出

　文字バイグラム，単語，フレーズなど

(2) 不要語の削除

(3) 接辞処理

(4) 索引語の重み付け

　検索手法（検索モデル）によっては不要

　例えば，論理式によるブーリアンモデルでは不要

(5) 索引ファイルの編成

ホデレ賞（2008年度）の受賞者が決まりました。

形態素	原形	品詞
ホデレ	ホデレ	未知語
賞	賞	名詞
（	（	記号
2008	2008	数字
年度	年度	助数詞
）	）	記号
の	の	助詞
受賞	受賞	名詞
者	者	接尾辞
が	が	助詞
決まり	決まる	動詞
まし	ます	助動詞
た	た	助動詞
。	。	記号

手順（1）～（3）の例

上の例文に対する形態素解析結果

赤字部分を索引語として抽出する

索引付けの手順概要

(1) 索引語の抽出

　文字バイグラム，単語，フレーズなど

(2) 不要語の削除

(3) 接辞処理

(4) 索引語の重み付け

　検索手法（検索モデル）によっては不要

　例えば，論理式によるブーリアンモデルでは不要

(5) 索引ファイルの編成

索引語の重み付け

- ある文書を特徴付ける索引語には高い重みを与える
- 伝統的な手法に TF.IDF法がある
  - TF: 索引語頻度
  - IDF: 逆文書頻度

  } これから詳細を説明

- 完全一致（ブーリアンモデル）では不要

  → ブーリアンモデルでは索引語に「あるかないか」だけ考える。「どれくらいあるか」は考えない。

TF: 索引語頻度

- Term Frequency（TF） → ここで言うTermとは索引語を表す

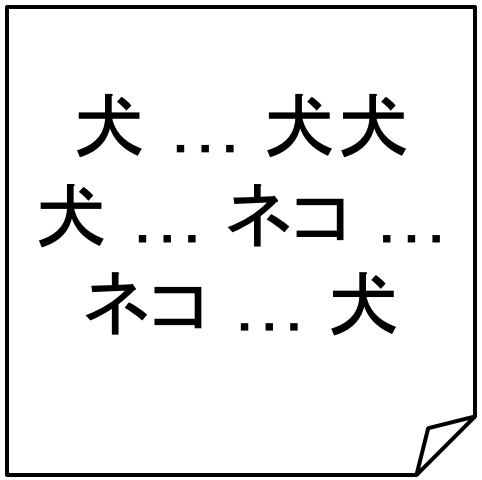
- $tf(t,d)$ と表す。

  文書 d における索引語 t の出現頻度

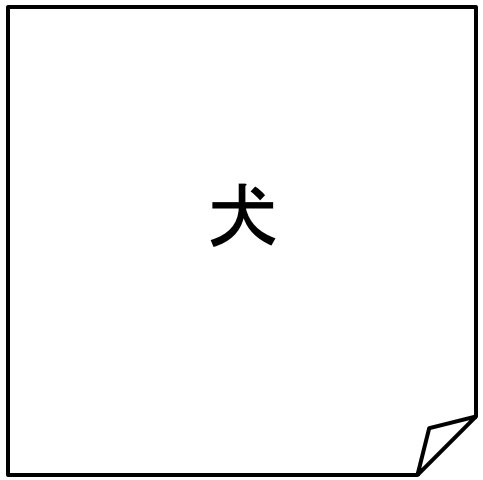
  → なぜ用いるか？

  → ある文書によく出現する索引語は，その文書をよく特徴付けるだろうという仮説に基づく

TFの例



文書A



文書B

$$tf(\text{犬}, A) = 5$$

$$tf(\text{ネコ}, A) = 2$$

$$tf(\text{犬}, B) = 1$$

IDF: 逆文書頻度

- Inverse Document Frequency（IDF）
- 少数の文書にしか現れない索引語を重視する

$$idf(t) = \log \frac{N}{df(t)} + 1$$

N: コレクション中の文書総数

df(*t*): 索引語 *t* が出現する文書数

→ なぜ用いるか？

→ TFだけでは問題がある。TFが高い語は多くの文書に出現する為，特定の文書を弁別する能力が低い → 例えば「は」「が」などはTFが非常に高いがほとんどどの文書にも現れる為，文書の特徴は表さない（弁別性に欠ける）。

逆文書頻度（つづき）N=100の場合

逆数を取ることで
df(t)が小さいほど
大きな値にする

対数を取ることで変化分
をなだらかにする

1を足して，重みを
正数にする

df(t)	N/df(t)	log(N/df(t))	log(N/df(t))+1
1	100	6.64	7.64
2	50	5.64	6.64
5	20	4.32	5.32
10	10	3.32	4.32
100	1	0	1

IDFの例

動物 ネコ	動物 犬 犬	動物 犬 ネコ	動物 犬 ロボット	動物 動物 犬

$N = 5$

df　動物=5，犬=4，ネコ=2，ロボット=1

~~動物=6，犬=5~~

idf(動物) = 1

idf(犬) = 1.32

idf(ネコ) = 2.32

idf(ロボット) = 3.32

- idfの最小値
- 「動物」では全文書が検索されてしまい，弁別性が低い

TF.IDF法による重みの計算

- 簡単な計算方法

$$w(t,d) = tf(t,d) \times idf(t)$$

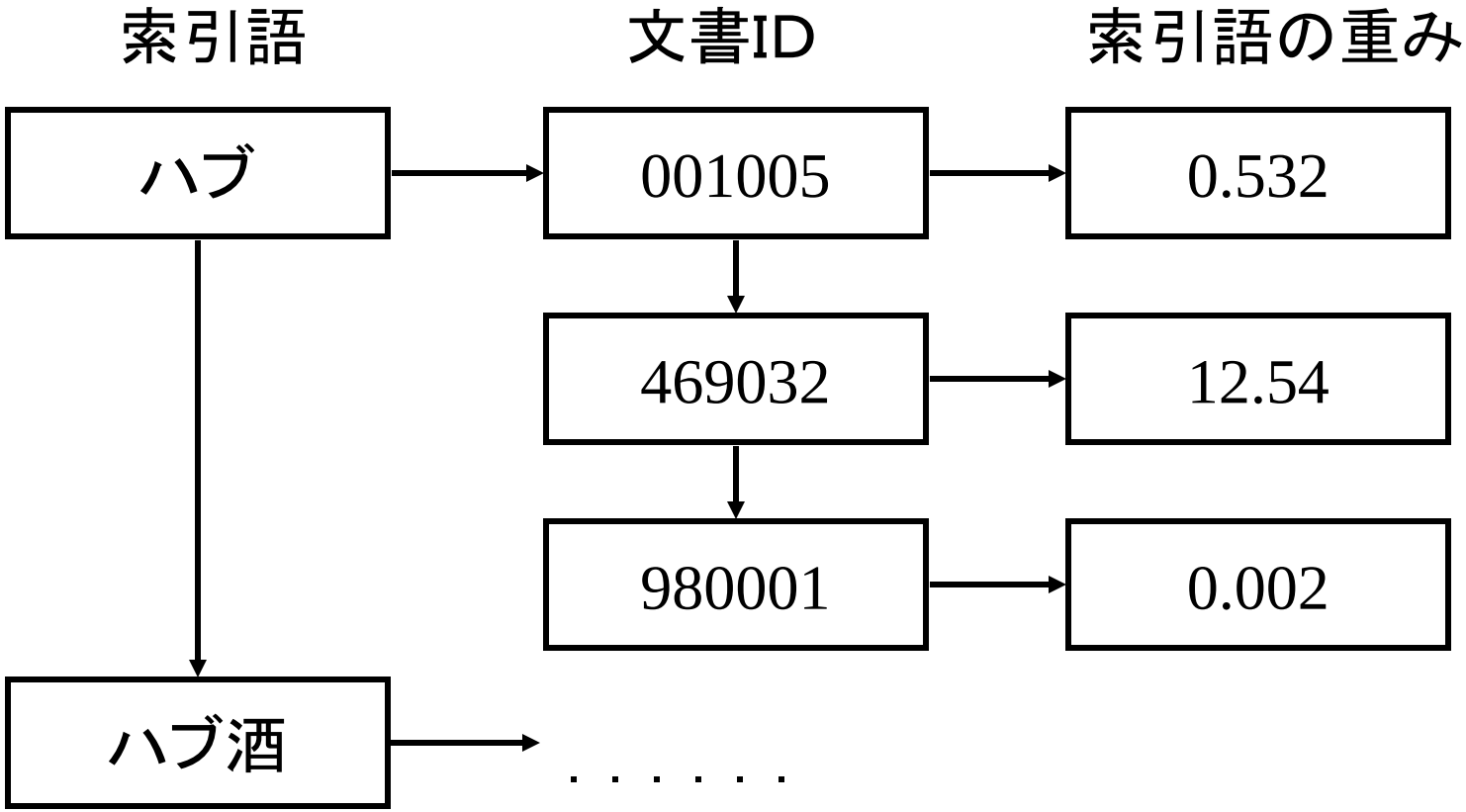
文書 d における索引語 t の重み

- 以下のような行列で表現できる

$w(t_2,d_3)$の値

	d_1	d_2	d_3	d_4	d_5
t_1					
t_2					
t_3					
t_4					

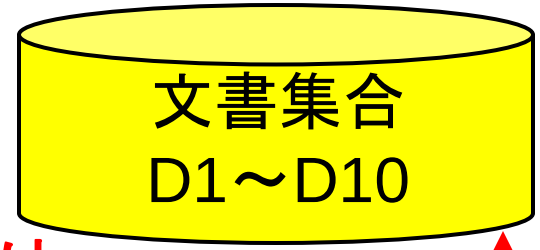
転置ファイルの例



オンライン処理

① 検索質問から索引語(検索語)を抽出する
② 各索引語について索引から以下を取得する
  - その索引語を含む文書の集合
  - その索引語の重みw(t,d)
③ 各文書のスコアを計算する
  - その文書が含む検索語の重みを総和する
④ スコアに基づいて文書を整列(ソート)する

オンライン処理の図解

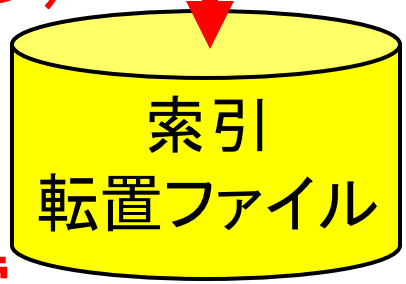


索引付け
（オフライン）

①索引語の抽出

犬　→ D2(0.1) D3(0.8) D5(1.2) D9(0.1)

ロボット → D1(1.3) D3(0.7) D5(0.1)



②文書と重みの探索

③スコアの計算

D1 = 1.3
D2 = 0.1
D3 = 0.8 + 0.7 = 1.5
D5 = 1.2 + 0.1 = 1.3
D9 = 0.1

④文書の整列

1. D3
2. D5
3. D1
4. D2
5. D9

個別の文書を
読む場合

演習：「Perl入門」が終了した人

- 授業ページに置いた documents.txt を読み込んで，各単語 t の各文書 d における重み w(t,d) を計算するプログラムを作成せよ。
  - ここで d とは <TEXT>タグと</TEXT>で囲まれた8つの英語テキスト
- 入力や出力の形式は各自で決めてよい。
- まずは各単語が各英語テキストそれぞれに何回出現しているか数える（即ち，tf(t,d)を算出する）プログラムを書くとよい。