



知識情報演習III (前半第3回)

関 洋平

筑波大学 図書館情報メディア系

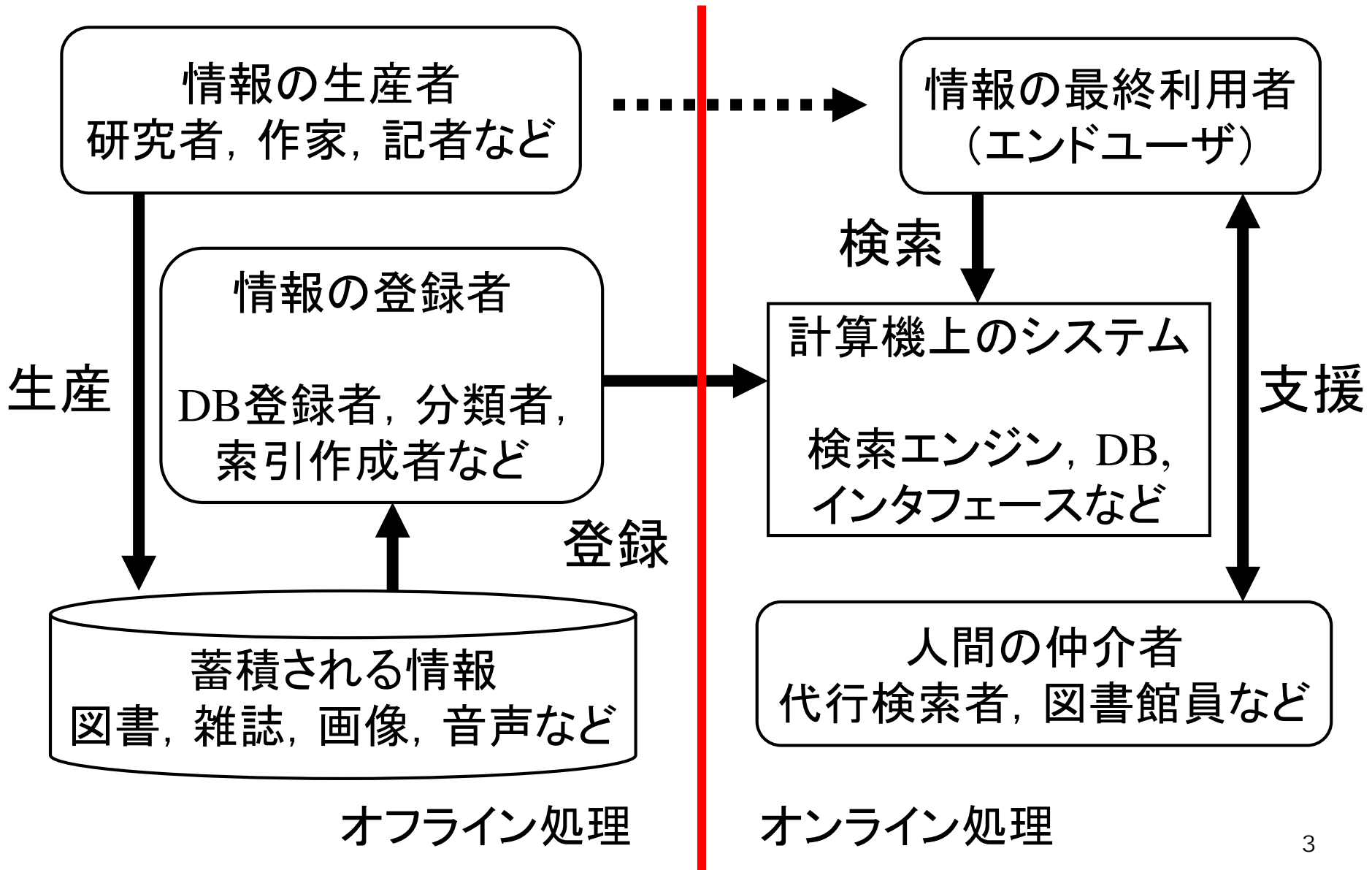
yohei@slis.tsukuba.ac.jp



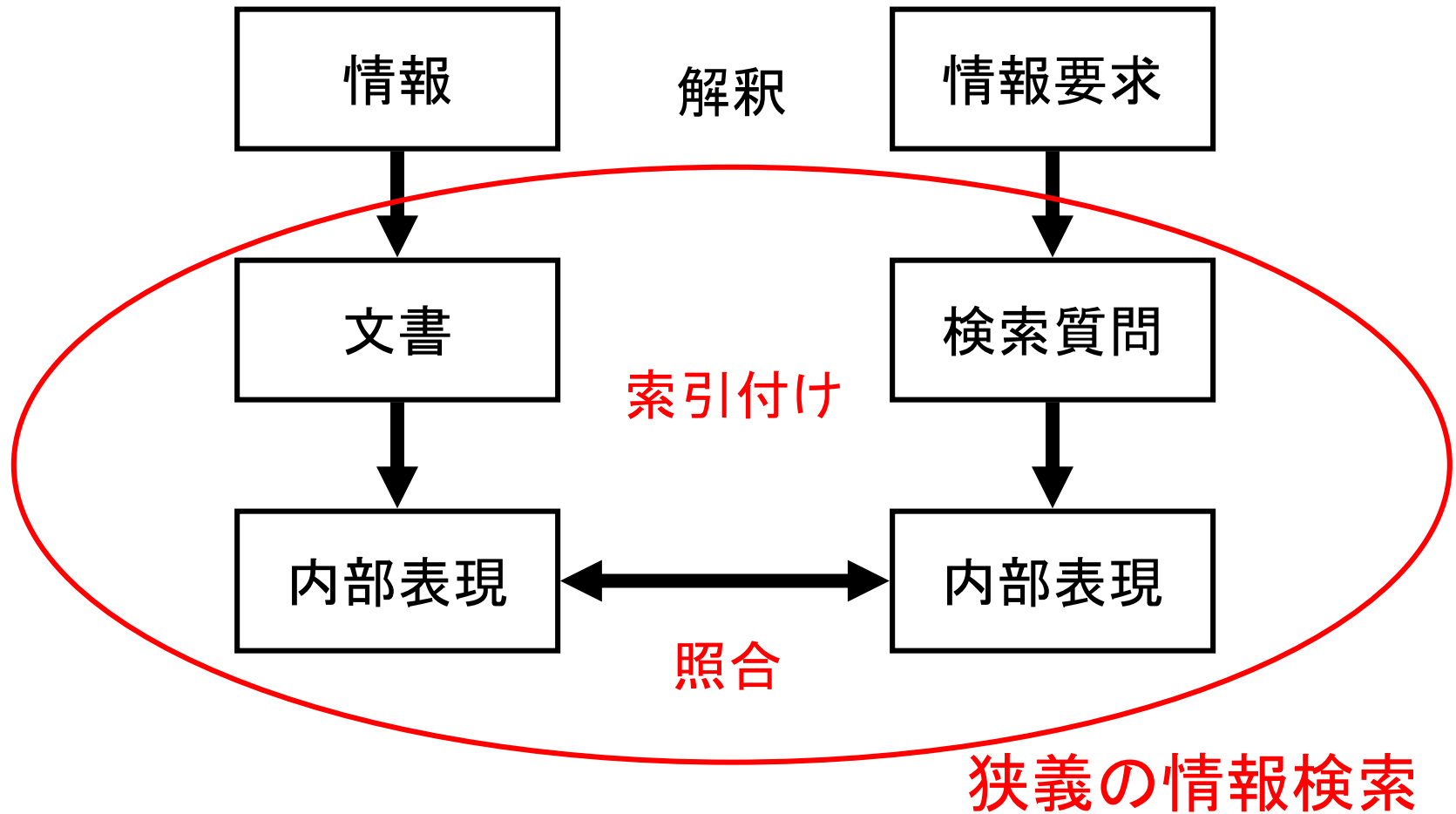
内容

1. 前回の復習
2. 索引語の重み付け: TF.IDF
3. 索引付けの実装

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

(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

ホデレ
賞
(
2008
年度
)
の
受賞
者
が
決まり
まし
た
。

原形

ホデレ
賞
(
2008
年度
)
の
受賞
者
が
決まる
まし
た
。

品詞

未知語
名詞
記号
数字
助数詞
記号
助詞
名詞
接尾辞
助詞
動詞
助動詞
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



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

(5) 索引ファイルの編成



内容

1. 前回の復習
2. **索引語の重み付け: TF.IDF**
3. 索引付けの実装

索引語の重み付け

- ある文書の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
- 完全一致（ブーリアンモデル）では不要

索引語頻度

- Term Frequency (TF)
- ある文書によく出現する索引語は、その文書の特徴付けるといふ仮説に基づいている
- $tf(t, d)$
文書 d における索引語 t の出現頻度
- 索引語を「ターム」とも呼ぶ（単語とは限らない）
- TFは文書と索引語が与えられて決まる尺度

TFの例

犬 ... 犬犬
犬 ... ネコ ...
ネコ ... 犬

文書A

$$tf(\text{犬}, A) = 5$$

$$tf(\text{ネコ}, A) = 2$$

犬

文書B

$$tf(\text{犬}, B) = 1$$

逆文書頻度

- Inverse Document Frequency (IDF)
- 多くの文書に出現する索引語は、特定の文書を弁別する能力が低い
- 少数の文書にしか現れない索引語を重視

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

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

- 索引語だけで決まる尺度 (TFとの違いに注意)

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~~

$idf(\text{動物}) = 1$ ←

$idf(\text{犬}) = 1.32$

$idf(\text{ネコ}) = 2.32$

$idf(\text{ロボット}) = 3.32$

- 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

Perlにおけるハッシュ

- 配列と違って文字列をキーとして使える
- 1つのキーで値を特定できるデータ

例： 索引語 dog の IDF が 2.5

$\$idf\{\text{'dog'}\} = 2.5;$

- 複数のキーで値を特定できるデータ

例： 索引語 dog の文書D001における TF が 10

$\$tf\{\text{'dog'}\}\{\text{'D001'}\} = 10;$

キーが1つの場合

$\$idf\{key\}$

%idf

key	value
dog	2.5
cat	1.6
⋮	⋮
year	3.3
⋮	⋮

%idf =

('dog' => 2.5,
'cat' => 1.6,
'year' => 3.3);

$\$idf\{'dog'\} = 2.5;$

$\$idf\{'cat'\} = 1.6;$

$\$idf\{'year'\} = 3.3;$

キーが複数の場合

%tf

key	value
dog	●
cat	●
⋮	⋮
year	●
⋮	⋮

%{\$tf{'dog'}}
というハッシュ

$\$tf\{key\}\{key2\}$

key2	value
D001	10
D002	3
⋮	⋮

$\$tf\{'dog'\}\{'D002'\} = 3;$

ハッシュの名前

%{\$tf{'cat'}}


key2	Value
D002	14
⋮	⋮

%{\$tf{'year'}}

ハッシュの内容を出力するプログラムの例

キーが1つ


```
foreach $term (sort keys %idf) {  
    print "$term $idf{$term}¥n";  
}
```



```
dog 2.5  
cat 1.6  
...  
year 3.3  
...
```

キーが2つ

```
foreach $x (sort keys % {$tf{'dog'}}) {  
    print "$x $tf{'dog'} {$x}¥n";  
}
```



```
D001 10  
D002 3  
...
```

演習1

- 演習のページにある
tf_idf.pdf の内容を入力して実行せよ
 - ✓ コピーペーストできないPDFファイルなので、
全て自分で入力すること
 - ✓ 印刷はできます
- 次に、重み $tf(t,d) \times idf(t)$ を計算して
出力するように修正せよ
 - ✓ 実際には、最後の方に何行か追加すればよい



演習作業のまとめ

- Perl入門1の例題を全てやる → 確認
- Perl入門2の例題を全てやる → 確認
- 第3回目の演習1をやる → 確認
- extract.pl から idf.plの作成に挑戦



内容

1. 前回の復習
2. 索引語の重み付け: TF.IDF
3. 索引付けの実装

索引付けプログラムの実装：方針

- 索引付けの段階ごとにプログラムを作る
- 小さなプログラムを複数作ることによって、実装を段階的に行う
 - ✓ 大きなプログラムを作ると、中間データの保存が煩雑になる
 - ✓ うまく動かない場合に問題の所在が分かりづらい
- 複数のプログラムを連結させる方法
 - ✓ 方法1： 中間ファイルを作る
 - ✓ 方法2： パイプライン処理を行う

索引付けの手順概要

- (1) 索引語の抽出

extract.pl

文字バイグラム, 単語, フレーズなど

- (2) 不要語の削除

stopword.pl

- (3) 接辞処理

stemming.pl

- (4) 索引語の重み付け

tf.pl

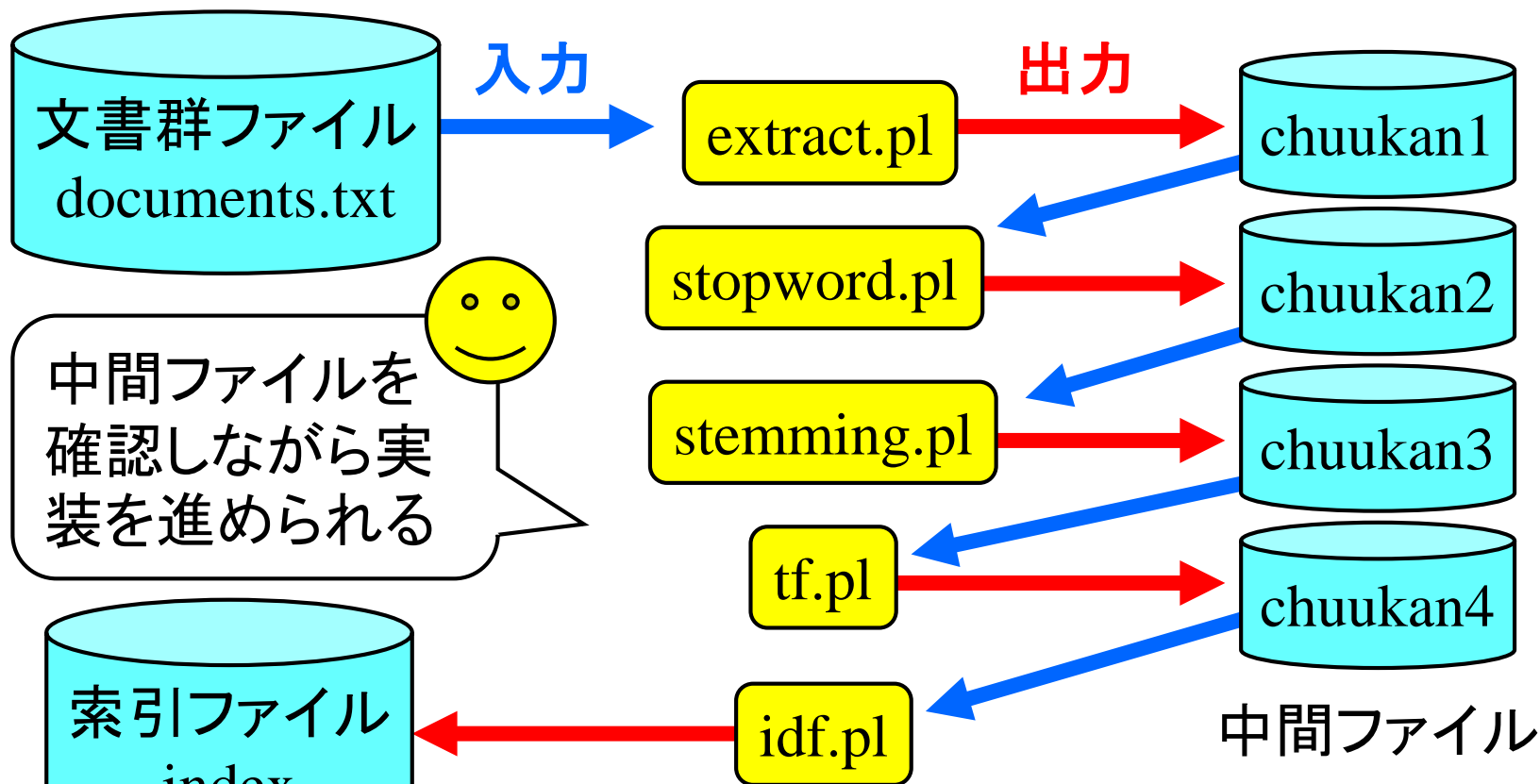
idf.pl

検索手法（検索モデル）によっては不要

例：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

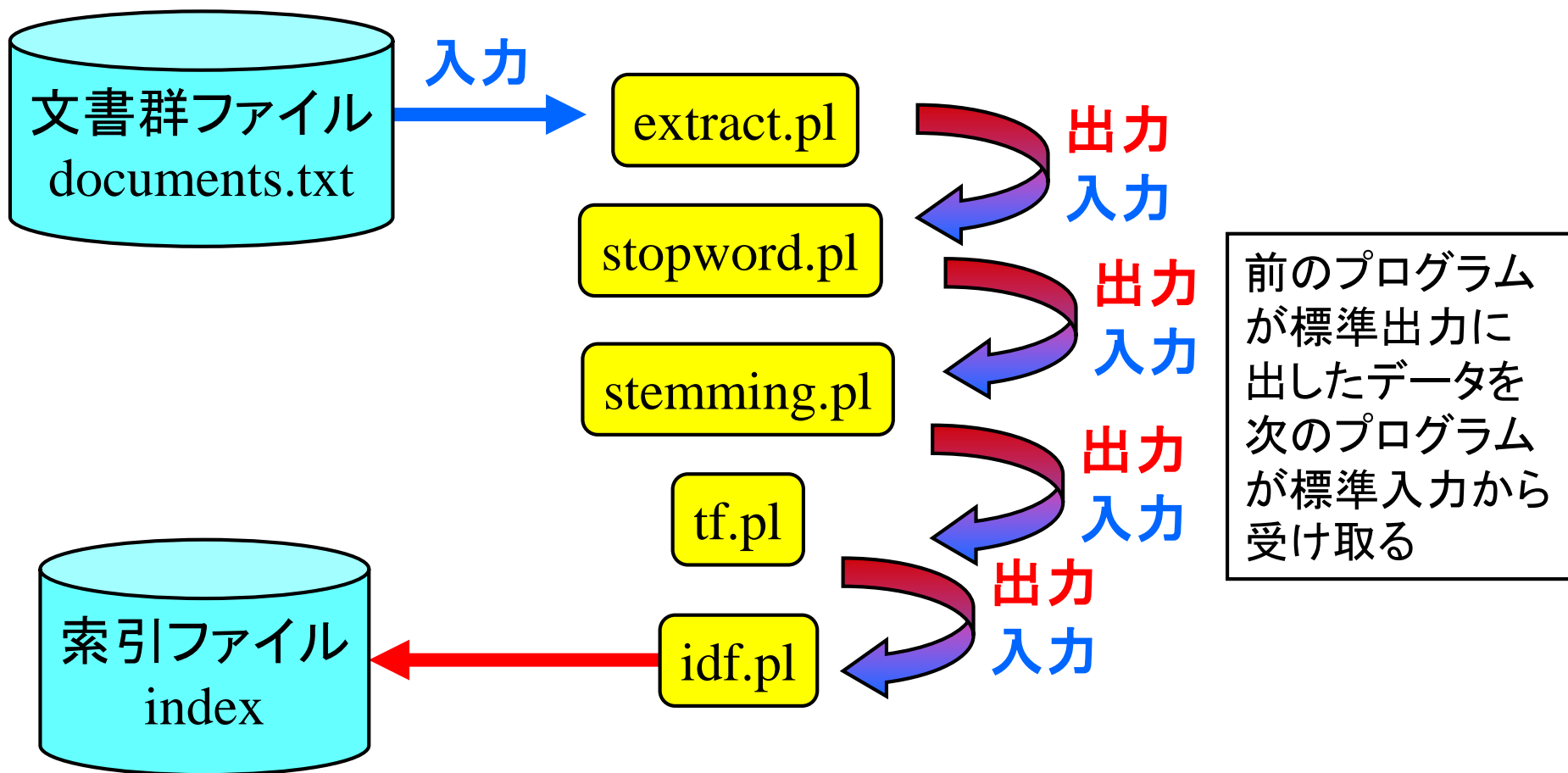
連結方法1: 中間ファイルを作る



本来不要なファイルがたくさんできる

```
% perl extract.pl documents.txt > chuukan1
% perl stopwords.pl chuukan1 > chuukan2
% perl stemming.pl chuukan2 > chuukan3
% perl tf.pl chuukan3 > chuukan4
% perl idf.pl chuukan4 > index
```

連結方法2: パイプライン処理を行う



複数のコマンドを縦棒でつなぐ(改行せずに1行で書く)

```
% perl extract.pl documents.txt | perl stopwords.pl |  
perl stemming.pl | perl tf.pl | perl idf.pl > index
```


文書群ファイルの形式

<DOC>

<NUM>D001</NUM>

<TEXT>

He is a student. ...

Students are ... student ...

She is not a student. ...

</TEXT>

</DOC>

<DOC>

<NUM>D002</NUM>

<TEXT>

Two dogs are ... The dog is ...

</TEXT>

</DOC>

...

<DOC> 1つの文書

<NUM> 文書番号

<TEXT> 本文

英文の文書を対象とする

演習のページにある

documents.txt を使うとよい

必要に応じて小さい

(または大きい)ファイルを
自分で作成してもよい

extract.pl の仕様

- 文書群ファイルを入力し、空白を区切りとして索引語を抽出
- 索引語を小文字に統一
- 索引語の末尾に付いたカンマとピリオドを削除
- 以下の形式で出力

```
D001 he  
D001 is  
D001 a  
D001 student
```

...

```
D002 two  
D002 dogs
```

1行に「文書番号 索引語」
文書番号と索引語は
半角スペース1つで区切る

stopword.pl の仕様

- extract.pl の出力を入力し，不要語を削除
- 不要語のリスト（自分で適宜追加してよい）

a, an, and, in, of, the

×

```
D001 he
D001 is
D001 a
D001 student
...
D002 two
D002 dogs
```



```
D001 he
D001 is
D001 student
...
D002 two
D002 dogs
```

stemming.pl の仕様

- stopwords.pl の出力を入力し，接辞処理を行う
- 接辞処理の規則（自分で適宜追加してよい）
 - ✓ 複数形への対応（末尾の s や es を削除）
 - ✓ 過去形への対応（末尾の ed を削除）

副作用が起きても
気にしない

D001 he
D001 is
D001 student
...
D002 two
D002 dogs



D001 he
D001 i
D001 student
...
D002 two
D002 dog

tf.pl の仕様

- stemming.pl の出力を入力し、
文書ごとに索引語の頻度（TF）をかぞえる
- 文書総数をかぞえてファイルの先頭行に出力する

```
D001 he
D001 i
D001 student
...
D001 student
D001 student
D002 dog
...
D002 dog
D003 dog
```



文書の総数
(IDFの計算に必要)

```
10
D001 he 1
D001 i 1
D001 student 4
...
D002 dog 2
D003 dog 1
```

idf.pl の仕様

- tf.pl の出力を入力し，索引語のIDFを計算する
- $TF \times IDF$ によって索引語の重みを計算する
- 文書の総数は出力しない

```
10
D001 he 1
D001 i 1
D001 student 4
...
D002 dog 2
D003 dog 1
```



```
D001 he 1 2.6 2.6
D001 i 1 1 1
D001 student 4 3.3 13.2
...
D002 dog 2 2.2 4.4
...
```

索引ファイルが完成

文書番号 索引語 TF IDF 重み



演習作業のまとめ

- Perl入門1の例題を全てやる → 確認
- Perl入門2の例題を全てやる → 確認
- 第3回目の演習1をやる → 確認
- extract.pl から idf.plの作成に挑戦