知識情報演習Ⅲ（前半第5回）

辻　慶太

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

索引付けの手順概要（復習）

(1) 索引語候補の抽出　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

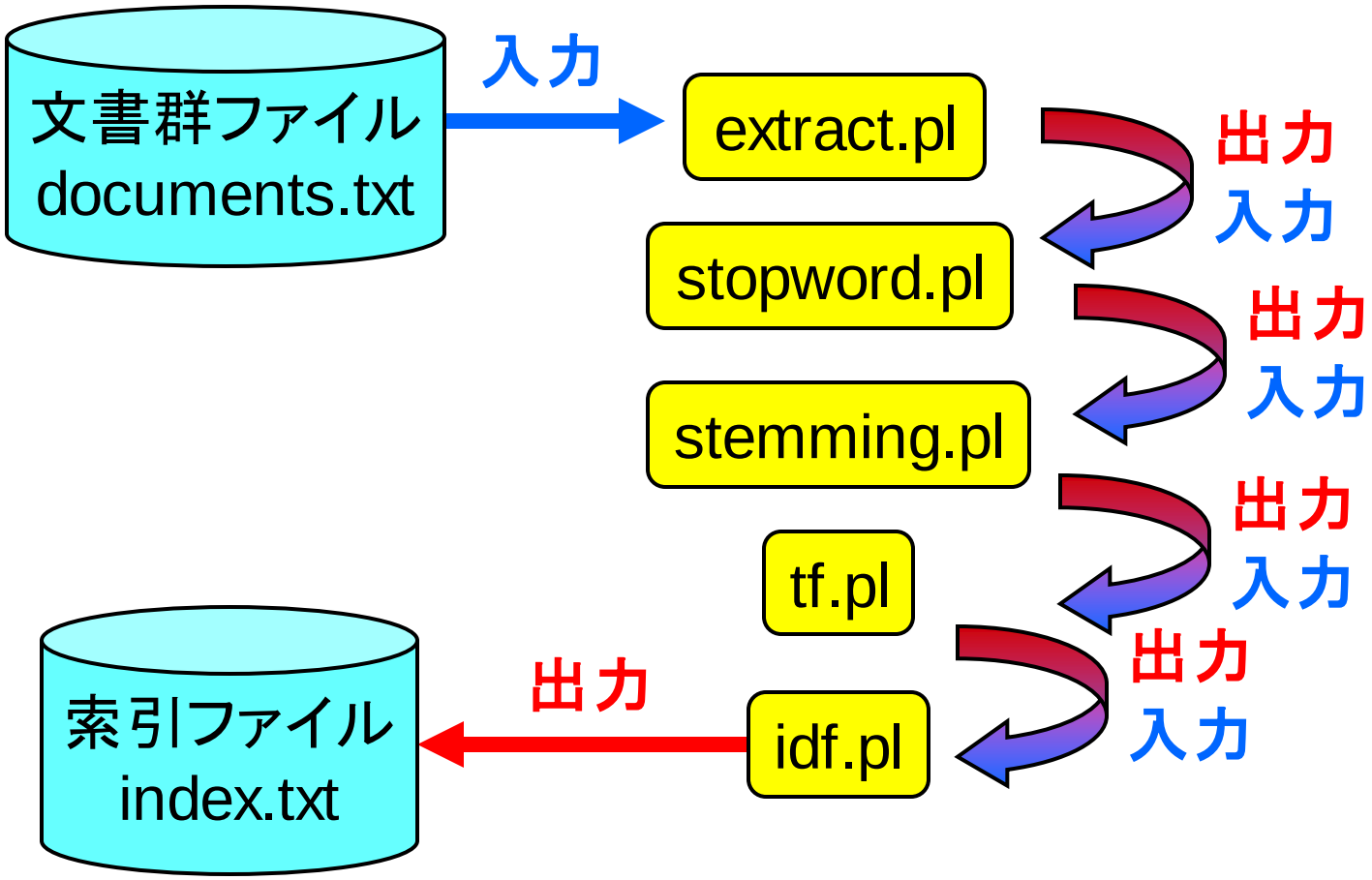
(4) 索引語の重み付け　tf.pl　idf.pl

　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

索引付けのプログラム（復習）



作って頂きたい6つ目のプログラム
retrieval.pl

- 語を指定すると，索引ファイルを見て，該当する文書を出力するプログラム

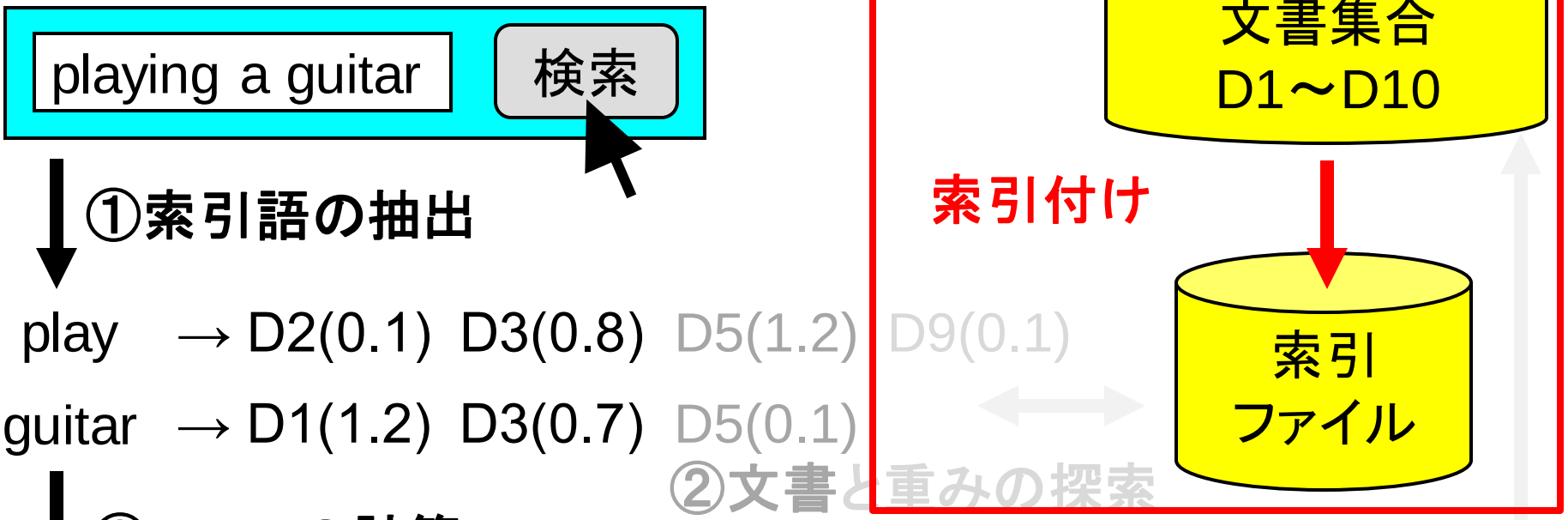
  - 索引付けと同様に以下の前処理が必要
    (1) 索引語の抽出
    (2) 不要語の削除
    (3) 接辞処理

    } 索引付けのプログラムをそのまま使う

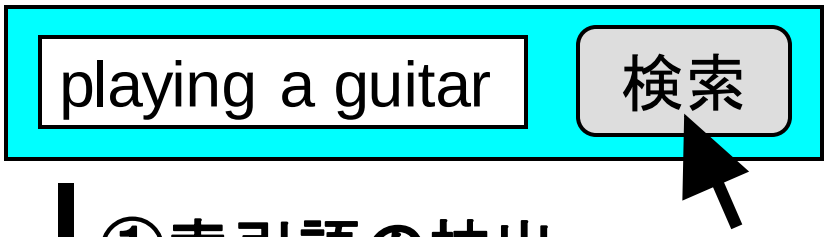
  - その後で文書ごとにスコアを計算し，スコアの降順に整列して出力する

retrieval.pl

処理の図解


playing a guitar
検索
①索引語の抽出
play → D2(0.1) D3(0.8) D5(1.2) D9(0.1)
guitar → D1(1.2) D3(0.7) D5(0.1)
②文書と重みの探索
③スコアの計算
D1 = 1.2
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
文書集合
D1～D10
索引付け
索引
ファイル
個別の文書を
読む場合

処理の図解



①索引語の抽出

play → D2(0.1) D3(0.8) D5(1.2) D9(0.1)

guitar → D1(1.2) D3(0.7) D5(0.1)

②文書と重みの探索

③スコアの計算

D1 = 1.2
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

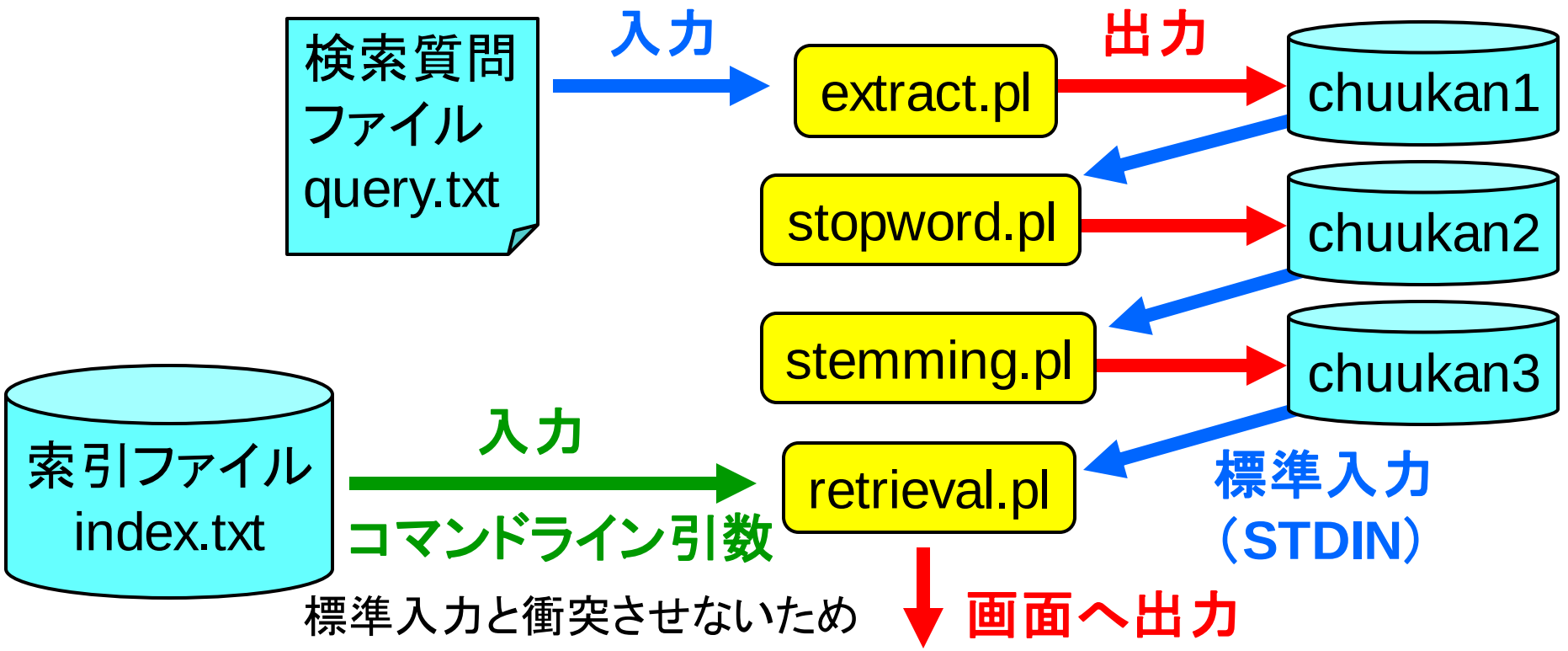
検索質問ファイルの形式

- 文書群ファイル（documents.txt）と形式を揃えることで，索引付けのプログラムを流用できるようにする
- 授業用ページにある query.txt を使うとよい

```
<QUERY>
<NUM>Q001</NUM>
<TEXT>
playing a guitar
</TEXT>
</QUERY>
```

<QUERY> 1つの検索質問
<NUM> 検索質問番号
<TEXT> 検索質問の本文

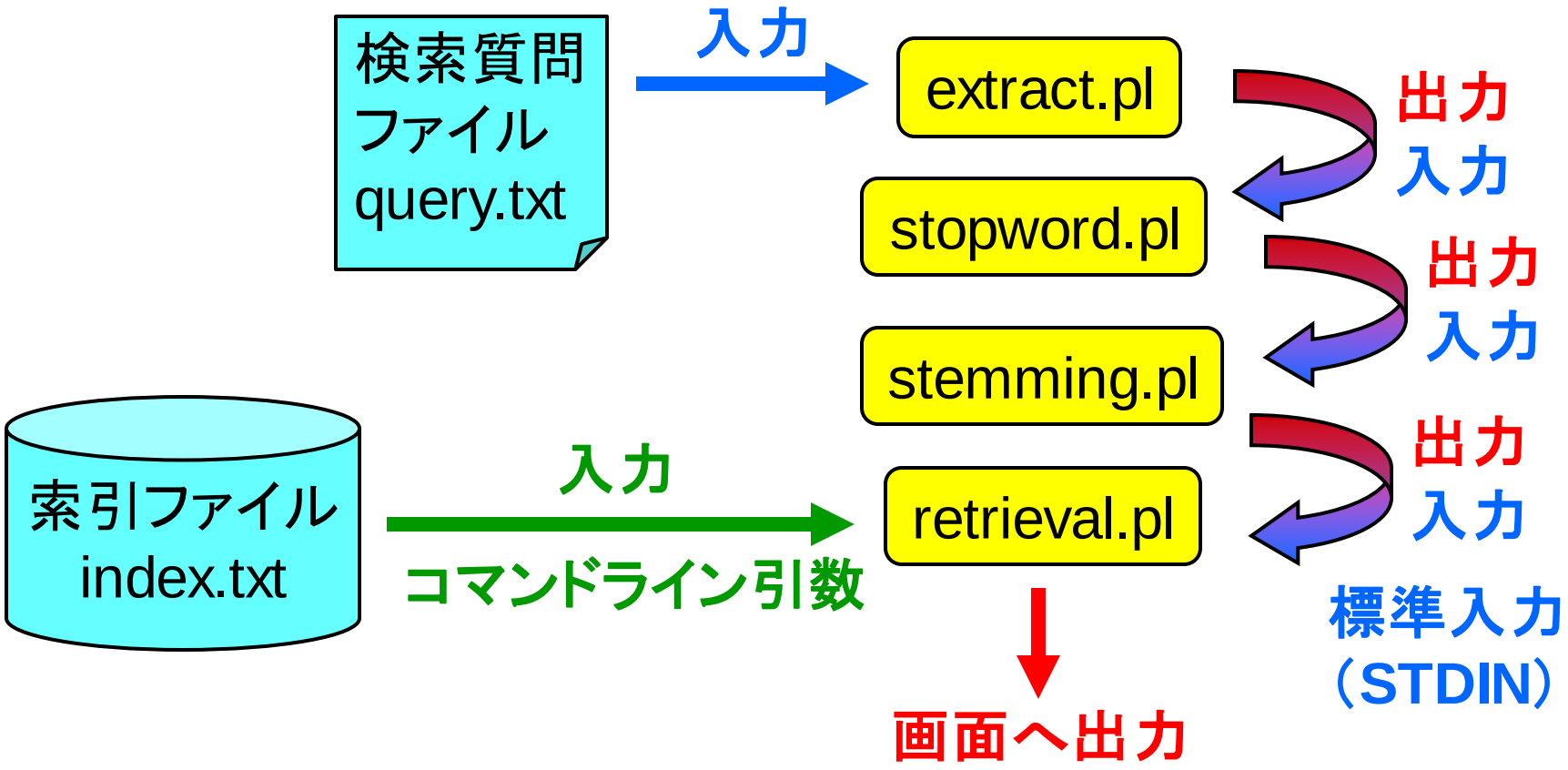
連結方法1：中間ファイルを作る




```
% perl extract.pl query.txt > chuukan1
% perl stopword.pl chuukan1 > chuukan2
% perl stemming.pl chuukan2 > chuukan3
% perl retrieval.pl  index.txt  < chuukan3
```

コマンドライン引数　　標準入力（STDIN）

連結方法2：パイプライン処理を行う



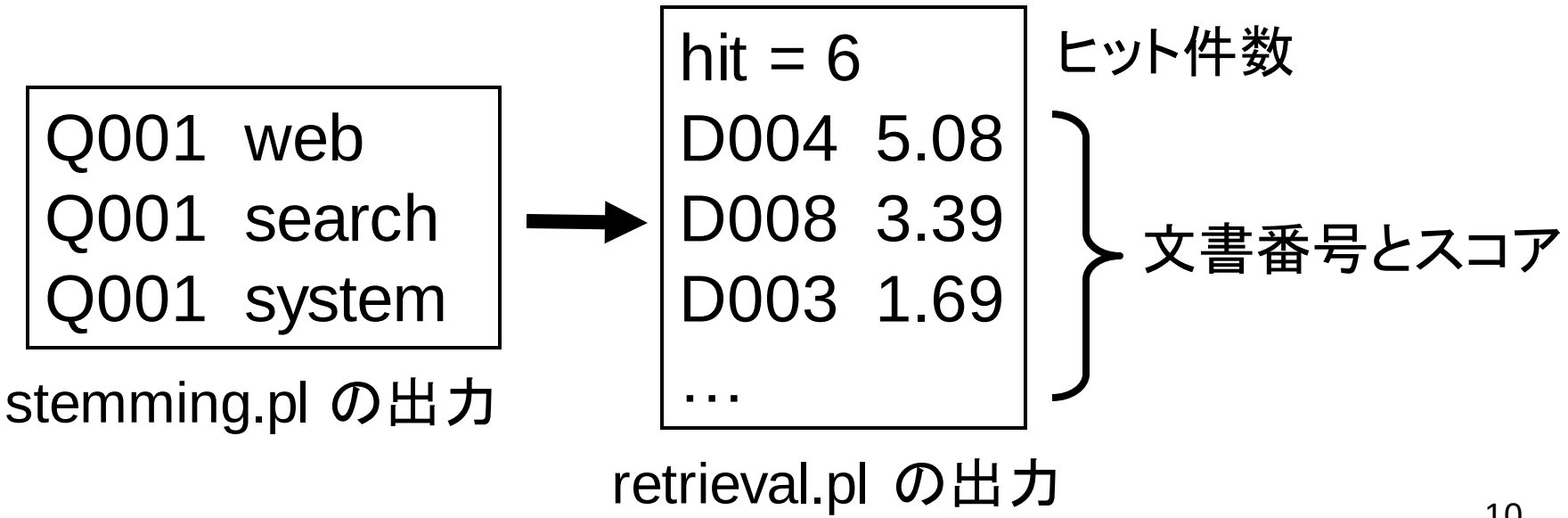

```
% perl extract.pl query.txt | perl stopword.pl |
perl stemming.pl | perl retrieval.pl  index.txt
```

標準入力（STDIN）

コマンドライン引数

retrieval.pl の仕様

- stemming.pl の出力と索引ファイル（index.txt）を入力し，検索された文書をスコアの降順に整列して出力する
- ヒット件数（検索された文書数）を先頭の行に出力する



index.txt の開き方①エラー処理

- ユーザがプログラムの使い方を誤った場合への対処が重要 → **エラー処理**

```
open(IN, $file)  ||  die ”$file: $!”;
```

ファイルのオープン

または

エラーメッセージを出して終了

ファイルのオープンに失敗したらエラーメッセージを出して終了

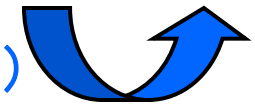
```
# 索引ファイル index.txt をコマンドラインで指定する
if (@ARGV != 1) {  # ファイルが指定されていない場合は
    print STDERR "Usage: $0 <index file>¥n"; # エラーを出して
   exit;                                          # 強制終了
};
```

index.txt の開き方②実際に開く

```
# 索引ファイル index.txt を読む
($index_file) = @ARGV;
open(IN, $index_file) || die "$index_file: $!";
while ($line = <IN>) {
    chomp($line);
    ...
}
close(IN);
```

```
% perl extract.pl query.txt | perl stopword.pl |
perl stemming.pl | perl retrieval.pl  index.txt
```

標準入力（STDIN）



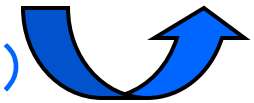
コマンドライン引数

query.txt 方面からの（パイプラインからの）情報の得方

```
# 検索質問のファイルを標準入力（STDIN）から読む
@qterm = ();
while ($query = <STDIN>) {
    chomp($query);
    ...
```

% perl extract.pl query.txt | perl stopword.pl | perl stemming.pl | perl retrieval.pl index.txt

標準入力（STDIN）



コマンドライン引数

レポート課題

- 索引付けとオンライン検索のプログラムを作成する
  - extract.pl, stopword.pl, stemming.pl, tf.pl, idf.pl, retrieval.pl
- プログラムに行番号を付けて，それぞれの行について別紙で説明する
  - 説明は，プログラム1行につき1文程度でよい

プログラムと説明の例（extract.pl）

```
1 while ($line = <>) {
2     chomp($line);
3     if ($line =~ /<NUM>(.+)<¥/NUM>/) {
4         $docid = $1;
5     }
6 ……
```

プログラムと説明を別のページに印刷する

1 ファイルの内容がなくなるまで1行ずつ読み込む
2 ファイル行の末尾にある改行を削除する
3 ファイル行に <NUM> と </NUM> があれば
4 <NUM> と </NUM> の間にある文字列を$docidに代入する
5 3行目の if 文による条件分岐の終了

レポートの提出方法

- 形式
  - 表紙：科目名，受講クラス（火・水），学籍番号，氏名
  - 本文：6つのプログラムとそれらの説明
  - 注意：
    - 両面印刷し，左上をステープラで止める
    - プログラムと説明が見開きのページになるように
    - どのプログラムか分かるように
    - 可読性が良くなるように努めること
    - 同一・酷似レポートはどちらも0点とする
- 提出場所
  - 学務係のレポートボックス
  - 受講クラスごとにボックスがあるので間違えないように（間違えた場合は採点対象から漏れる場合があります）
- 締切（厳守）
  - 1月11日（水）17:00

成績評価

- 後半5回分の配点（50点）
  - 出席（30%）
    - 3点×5回＝15点
  - レポート（70%）
    - 「プログラム」と「説明」の組で各5点　5点×6＝30点
    - レポートの体裁や可読性など　5点
- 前半5回分と総合して100点満点で計算し，A～Dを判定する
- 完成したプログラムの数が多いほど評価は高い
- 全てのプログラムが完成しなくても及第点に及ぶ可能性はあるので，あきらめないように

ここでの数値は目安であり，実際の評価では多少変動することがある