「集合知プログラミング」

第4章　検索とランキング(2)

06t4071f　林華

問い合わせ

- 単一の単語の場合→一度一つ検索できる

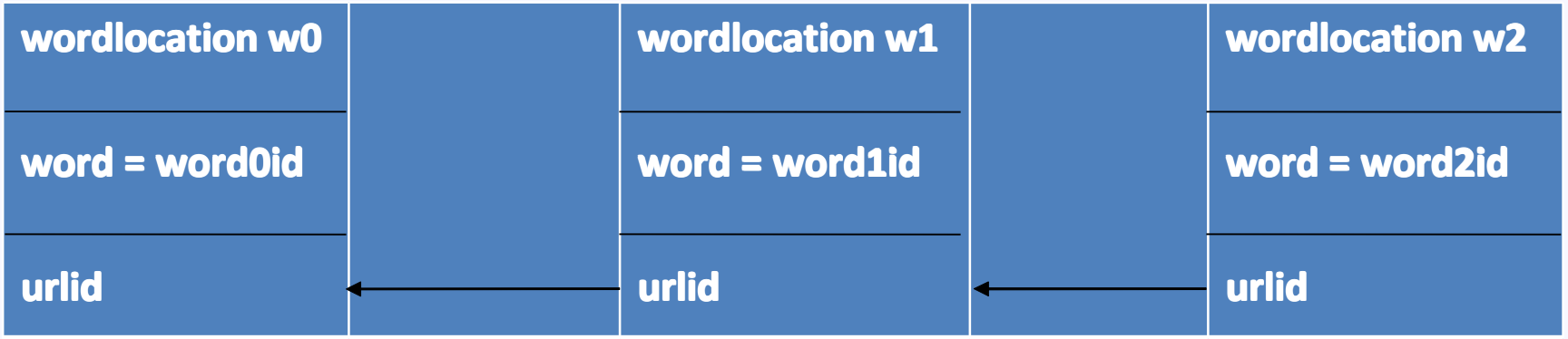
ところが 

- 複数の単語の場合→機能の拡張必要がある

複数単語のクエリへの処理

- クエリを作るための文字列を用意
- 単語分割
- 単語のIDを取得
- クエリに全ての単語を含むページを抽出

例



設定例：

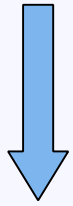
```
select w0.urlid, w0.location, w1.location
from wordlocation w0, wordlocation w1
where w0.urlid=w1.urlid
and w0.wordid=10
and w1.worid=17
```

出力例：

```
>>reload(searchengine)
>>e=searchengine.searcher('searchindex.db')
>>e.getmatchrows('functional programming')
([1, 327, 23), (1, 327, 162), (1, 327, 243), (1, 327, 261),
(1, 327, 269), (1, 327, 436), (1, 327, 953), ・・・
```

上例の不具合点とその対策

- 単純にクロールされた順であった
- 検索したい内容は結果と関連性が低い

対策 

- クエリによるページ内容のスコア算出
- ページ内容以外の情報のスコア算出

内容ベースの順位付けアプローチ

- 単語の出現頻度

  ドキュメント内、クエリ中単語の出現頻度

- ドキュメント中での位置

  ドキュメントの主題はドキュメントの最初の部分に出現する可能性が高い

- 単語間の距離

  クエリ中に複数の単語が含まれる場合、それらは近くにくっついた状態でドキュメント中に出現する可能性が高い

単語頻度に基づくスコアリング

検索語出現する回数多いほど、そのページと検索語の関連性が高い

1.ページ毎にクエリにあるアイテムを数える
2.その数をスコアとして正規化
  – 正規化：①最善のスコアを1とし、基準とする
    ②残りのスコアは基準との比例をとる
3.正規化されたスコアの降順に出力

ドキュメント中での位置に基づく

検索語と関連性高いページであれば、その単語はページの最初部分に出現する

1.クエリにある単語の位置を合計
2.最小位置合計のページを1とする
3.各ページを正規化
4.結果を降順に出力

単語間の距離に基づく

ページ中でクエリ中の単語同士が近いほど、関連性が高い

1.単語は一つしかない場合、全て出力
2.単語同士中に一番近いのをそのページの値
3.各ページの値を正規化
4.降順に出力

改善：三つ方法の組合せ

1.クエリ中の単語を全て含むページの抽出
2.各ページの単語間の最小距離値計算
3.ステップ2結果の上位1000ページ抽出
4.ステップ3結果の単語のドキュメント位置計算
5.距離値×位置値を計算、昇順ソート
6.ステップ5結果の上位100ページを抽出
7.単語の頻度を計算、降順ソート
8.正規化、出力