全文検索の基本的なしくみ

@myokoym

全文検索エンジンGroonga勉強会@札幌

2015-06-14

全文検索の流れ

- 文書を登録
- キーワードで検索

登録の流れ

- ノーマライズ
- トークナイズ
- トークンをキーとして文書IDをインデックスに登録

検索の流れ

- ノーマライズ
- トークナイズ
- トークンでインデックスを検索
- 文書IDを取得

用語

- ノーマライズ
- トークナイズ
- トークン

ノーマライズ

- 正規化
  - 大文字小文字、全角半角などを揃える
  - ノーマライザーによって挙動が変わる

トークナイズ

- 文書やキーワードをトークンに分割する
- トークナイザーによって分け方が異なる
  - 大きく分けるとN-gramと形態素解析の2種類

N-gram

- 文字数で分割する
  - bi-gramは2文字ごと
  - tri-gramは3文字ごと

形態素解析

- 品詞を判別して分割する
- 別途辞書が必要

トークン

- インデックスのキー
- Groongaで採用している転置インデックス方式では、トークンごとに文書リストが作成される

トークナイズの例

東京都府中市

bi-gram

東京/京都/都府/府中/中市/市

tri-gram

東京都/京都府/都府中/府中市/中市/市

形態素解析

東京/都/府中/市

全文検索の流れ

- 文書を登録
- キーワードで検索

全文検索の流れ

- **文書を登録**
- キーワードで検索

登録の流れ

- ノーマライズ
- トークナイズ
- トークンをキーとして文書IDをインデックスに登録

サンプル文書

文書ID	内容
1	カレー食べた
2	カレ-食べたい
3	ｶﾚｰ好き

ノーマライズ

- 1．カレー食べた
- 2．カレー食べたい
- 3．カレー好き

トークナイズ

- MeCabの例
  - 1．カレー/食べ/た
  - 2．カレー/食べ/たい
  - 3．カレー/好き

インデックスに登録

キー	文書IDリスト
カレー	[1 2 3]
食べ	[1 2]
た	[1]
たい	[2]
好き	[3]

全文検索の流れ

- 文書を登録
- キーワードで検索

検索の流れ

- ノーマライズ
- トークナイズ
- トークンでインデックスを検索
- 文書IDを取得

サンプルキーワード

- 食べた

ノーマライズ

- 食べた
  - ノーマライズ対象なし

トークナイズ

- 食べ/た

インデックス（再掲）

キー	文書IDリスト
カレー	[1 2 3]
食べ	[1 2]
た	[1]
たい	[2]
好き	[3]

トークンでインデックスを検索

- 食べ -> [1 2]
- た -> [1]

文書IDを取得

- [1]
  - 両方含まれるもの
  - 設定によって出現位置も調べる
    - 隣り合っているかどうかなど

ヒット1件

カレー食べた

ポイント

- 登録と検索で同じノーマライザーとトークナイザーを使う必要がある
  - それはなぜか？

参考

- Groongaの可変型Ngramトークナイザーについて - Naoya Murakami - Rabbit Slide Show
  - http://slide.rabbit-shocker.org/authors/naoa/groonga-tokenizer-talks-naoa/