MySQL・PostgreSQLだけで作る 高速あいまい全文検索システム

須藤功平

株式会社クリアコード

db tech showcase Tokyo 2018
2018-09-20

全文検索システム

- 大量の文書から
- 指定されたキーワードを使って
- 高速に必要な文書を
- 見つけるシステム

dbts2017

MySQL・PostgreSQLだけで作る 高速でリッチな全文検索システム

須藤功平　　*株式会社クリアコード*

db tech showcase Tokyo 2017
2017-09-07

https://slide.rabbit-shocker.org/authors/kou/db-tech-showcase-tokyo-2017/

リッチな全文検索システム

- キーワードハイライト
- 周辺テキスト表示
- 入力補完・同義語展開
- 関連文書の表示
- 構造化データ対応（例：オフィス文書）

もっとリッチな全文検索システム

- 大量の文書から
- **あいまいな情報を使って**
- 高速に必要な文書を
- 見つけるシステム

あいまいな情報

相手が人だから

人

- 文書内の表記が揺れる
  - 例：「焼き肉」「焼きにく」
  - 検索対象
- 提供情報が間違っている
  - 例：「テノクロジー」
  - 検索クエリー

もっとリッチな全文検索システム

- 大量の文書から
- **あいまいな情報を使って**
- 高速に必要な文書を
- 見つけるシステム

全文検索エンジン

普通の全文検索エンジン

- 高速検索
- あいまい検索
- 独自の使い方

開発しやすいシステム

SQLで使える

使いやすい全文検索エンジン

- 高速検索
- あいまい検索
- SQLで使える

SQLで使える全文検索エンジン

- Mroonga
  - MySQL・MariaDB用
  - http://mroonga.org/ja/docs/install.html
- PGroonga
  - PostgreSQL用
  - https://pgroonga.github.io/ja/install/

あいまい検索：表記ゆれ1

- 焼肉：全部漢字
- 焼き肉：送り仮名
- 焼きにく：まぜまぜ

あいまい検索：表記ゆれ1対策

ヨミガナ検索

ヨミガナ検索

- ヨミガナで検索
  - 「焼肉」→「ヤキニク」
  - 「焼きにく」→「ヤキニク」
  - どちらも同じになる
  - 人名（高と髙とか）にも使える
- ヨミガナ情報の取得が必要
  - MeCabで自動化可能

MeCabでヨミガナ化

```
% echo 焼肉と焼きにく | mecab | cut -d, -f8
ヤキニク
ト
ヤキニク
EOS
```

ヨミガナ検索：Mroonga：テーブル定義

```
CREATE TABLE menus (
  name varchar(255),     -- 検索対象
  FULLTEXT INDEX (name) -- ヨミガナ検索対応インデックス
    COMMENT
      'tokenizer "TokenMecab(\'use_reading\', true)"'
) ENGINE=Mroonga DEFAULT CHARSET=utf8mb4;
```

ヨミガナ検索：Mroonga：データ例

```
INSERT INTO menus
  VALUES ('焼肉定食'),
         ('焼きにく定食');
```

ヨミガナ検索：Mroonga：検索

```
SELECT name FROM menus
  WHERE MATCH (name)
    AGAINST ('*D+ 焼きにく' IN BOOLEAN MODE);
-- 焼肉定食
-- 焼きにく定食
```

ヨミガナ検索：PGroonga：テーブル定義

```
CREATE TABLE menus (
  name text -- 検索対象
);
```

ヨミガナ検索：PGroonga：インデックス定義

```
CREATE INDEX menus_search ON menus
  USING PGroonga (name)
  WITH (tokenizer='TokenMecab("use_reading", true)');
```

ヨミガナ検索：PGroonga：データ例

```
INSERT INTO menus
  VALUES ('焼肉定食'),
         ('焼きにく定食');
```

ヨミガナ検索：PGroonga：検索

```
SELECT name FROM menus
  WHERE name &@~ '焼きにく';
-- 焼肉定食
-- 焼きにく定食
```

あいまい検索：表記ゆれ2

「ぼたん鍋」
と
「猪鍋」
（別名）

あいまい検索：表記ゆれ2対策

同義語展開

同義語展開

- 実行前にクエリーを変換
  - 「ぼたん鍋」→「ぼたん鍋 OR 猪鍋」
  - どちらもヒット
- 変換ルールは事前に用意
  - ある程度自動生成可能
  - 例：NEologdやWikipediaを利用

同義語展開：Mroonga：テーブル定義

```
CREATE TABLE synonyms (
  term varchar(255),    -- 展開対象の語
  synonym varchar(255), -- 同義語
  INDEX (term)          -- 高速化と精度向上
    COMMENT 'normalizer "NormalizerNFKC100"'
) ENGINE=Mroonga DEFAULT CHARSET=utf8mb4;
```

同義語展開：Mroonga：データ例

```
INSERT INTO synonyms
-- 「ぼたん鍋」を「ぼたん鍋 OR 猪鍋」に展開
  VALUES ('ぼたん鍋', 'ぼたん鍋'),
         ('ぼたん鍋', '猪鍋'),
-- 「猪鍋」を「猪鍋 OR ぼたん鍋」に展開
         ('猪鍋', '猪鍋'),
         ('猪鍋', 'ぼたん鍋');
```

同義語展開：Mroonga：確認方法

```
SELECT mroonga_query_expand(
  'synonyms',          -- テーブル名
  'term',              -- 展開対象のカラム名
  'synonym',           -- 対応する同義語のカラム名
  'ランチ ぼたん鍋' -- クエリー
);
-- 'ランチ ((ぼたん鍋) OR (猪鍋))'
```

同義語展開：Mroonga：検索方法

```
SELECT title FROM entries
  WHERE
    MATCH (title)
    -- '*D+ ランチ OR ((ぼたん鍋) OR (猪鍋))'になる
    AGAINST (mroonga_query_expand('synonyms',
                                  'term',
                                  'synonym',
                                  '*D+ ランチ ぼたん鍋')
             IN BOOLEAN MODE);
```

同義語展開：PGroonga：テーブル定義

```
CREATE TABLE synonyms (
  -- 展開対象の語
  term text,
  -- 同義語のリスト
  -- term自身も含める
  -- 含めない場合はtermが検索禁止語になる
  terms text[]
);
```

同義語展開：PGroonga：データ例

```
INSERT INTO synonyms
  VALUES ('ぼたん鍋', -- 「ぼたん鍋」を展開
          ARRAY['ぼたん鍋', '猪鍋']),
         ('猪鍋', -- 「猪鍋」を展開
          ARRAY['猪鍋', 'ぼたん鍋']);
```

同義語展開：PGroonga：インデックス定義

```
CREATE INDEX synonym_search ON synonyms
  USING PGroonga
  -- ...text_term_search...
  -- termで完全一致検索をするため
  (term pgroonga_text_term_search_ops_v2);
```

同義語展開：PGroonga：確認方法

```
SELECT pgroonga_query_expand(
  'synonyms', -- テーブル名
  'term', -- 展開対象のカラム名
  'terms', -- 対応する同義語配列のカラム名
  'ランチ ぼたん鍋' -- クエリー
);
-- 'ランチ ((ぼたん鍋) OR (猪鍋))'
```

同義語展開：PGroonga：検索方法

```
SELECT title FROM entries
  WHERE
 -- title &@~ ランチ ((ぼたん鍋) OR (猪鍋))'になる
    title &@~
      pgroonga_query_expand('synonyms',
                            'term',
                            'terms',
                            'ランチ ぼたん鍋');
```

あいまい検索：表記ゆれ3

- 090-1234-5678：ハイフン入り
- (090)1234-5678：カッコとハイフン入り
- 09012345678：区切りなし
- 090 1234 5678：空白区切り
- （090）１２３４ー5678：全角文字入り

あいまい検索：表記ゆれ3対策

電話番号検索

電話番号検索

- 文字を正規化
  - 全角→半角
  - ハイフンっぽい文字→ハイフン
  - 長音っぽい文字→ハイフン
- 記号・空白を無視して検索
  - 元クエリー：(090)1234 5678
  - 実クエリー：09012345678

電話番号検索：Mroonga：テーブル定義

```
CREATE TABLE people (
  tel varchar(255), -- 検索対象
  FULLTEXT INDEX (tel) COMMENT
    -- 電話番号検索対応インデックス
    'normalizer "NormalizerNFKC100(
      \'unify_hyphen_and_prolonged_sound_mark\', true)",
     tokenizer "TokenNgram(\'loose_symbol\', true,
                           \'loose_blank\', true)"'
) ENGINE=Mroonga DEFAULT CHARSET=utf8mb4;
```

電話番号検索：Mroonga：データ例

```
INSERT INTO people
  VALUES ('090-1234-5678'),
         ('(090)1234-5678'),
         ('09012345678'),
         ('090 1234 5678'),
         (' (090) １２３４ー5678');
```

電話番号検索：Mroonga：検索

```
SELECT tel FROM people
  WHERE MATCH (tel)
    AGAINST ('*D+ 090−1234５６78' IN BOOLEAN MODE);
-- 090-1234-5678
-- (090)1234-5678
-- 09012345678
-- 090 1234 5678
--  (090) １２３４−5678
```

電話番号検索：PGroonga：テーブル定義

```
CREATE TABLE people (
  tel text -- 検索対象
);
```

電話番号検索：PGroonga：インデックス定義

```
CREATE INDEX people_search ON people
  USING PGroonga (tel)
  WITH (normalizer='
    NormalizerNFKC100("unify_hyphen_and_prolonged_sound_mark", true)',
        tokenizer='TokenNgram("loose_symbol", true,
                              "loose_blank", true)');
```

電話番号検索：PGroonga：データ例

```
INSERT INTO people
  VALUES ('090-1234-5678'),
         ('(090)1234-5678'),
         ('09012345678'),
         ('090 1234 5678'),
         (' (090) １２３４ー5678');
```

電話番号検索：PGroonga：検索

```
SELECT tel FROM people
  WHERE tel &@~ '090ー1234５６78';
-- 090-1234-5678
-- (090)1234-5678
-- 09012345678
-- 090 1234 5678
--  (090) １２３４ー5678
```

あいまい検索：表記ゆれ4

sèvre-et-maine

- セーヴェル エ メーヌ
  「ーヴェ」・空白区切り
- セブルエメーヌ
  「ブ」・区切りなし
- セーブル・エ・メーヌ
  「ーブ」・中点区切り
- セーヴル　エメーヌ
  「ーヴ」・片方だけ空白区切り

あいまい検索：表記ゆれ4対策

ワイン名検索

ワイン名検索

- 文字を正規化
  - ヴ・ヴェ→ブ
  - ハイフン・長音っぽい文字→ハイフン
  - 中点っぽい文字→中点
- 記号・空白を無視して検索
  - 元クエリー：セーヴェル・エメーヌ
  - 実クエリー：セブルエメヌ

ワイン名検索：Mroonga：テーブル定義

```
CREATE TABLE wines (
  name varchar(255), -- 検索対象
  FULLTEXT INDEX (name) COMMENT
    -- ワイン名検索対応インデックス
    'normalizer "NormalizerNFKC100(
      \'unify_katakana_bu_sound\', true,
      \'unify_hyphen_and_prolonged_sound_mark\', true,
      \'unify_middle_dot\', true)",
     tokenizer "TokenNgram(\'loose_symbol\', true,
                           \'loose_blank\', true)"'
) ENGINE=Mroonga DEFAULT CHARSET=utf8mb4;
```

ワイン名検索：Mroonga：データ例

```
INSERT INTO wines
  VALUES ('セーヴェル エ メーヌ'),
         ('セブルエメーヌ'),
         ('セーブル・エ・メーヌ'),
         ('セーヴル　エメーヌ');
```

ワイン名検索：Mroonga：検索

```
SELECT name FROM wines
  WHERE MATCH (name)
    AGAINST ('*D+ セーヴェルエメーヌ' IN BOOLEAN MODE);
-- セーヴェル エ メーヌ
-- セブルエメーヌ
-- セーブル・エ・メーヌ
-- セーヴル　エメーヌ
```

ワイン名検索：PGroonga：テーブル定義

```
CREATE TABLE wines (
  name text -- 検索対象
);
```

ワイン名検索：PGroonga：インデックス定義

```
CREATE INDEX wines_search ON wines
  USING PGroonga (name)
  WITH (normalizer='NormalizerNFKC100(
    "unify_katakana_bu_sound", true,
    "unify_hyphen_and_prolonged_sound_mark", true,
    "unify_middle_dot", true)',
        tokenizer='TokenNgram("loose_symbol", true,
                              "loose_blank", true)');
```

ワイン名検索：PGroonga：データ例

```
INSERT INTO wines
  VALUES ('セーヴェル エ メーヌ'),
         ('セブルエメーヌ'),
         ('セーブル・エ・メーヌ'),
         ('セーヴル　エメーヌ');
```

ワイン名検索：PGroonga：検索

```
SELECT name FROM wines
  WHERE name &@~ 'セーヴェルエメーヌ';
-- セーヴェル エ メーヌ
-- セブルエメーヌ
-- セーブル・エ・メーヌ
-- セーヴル　エメーヌ
```

表記ゆれ：まとめ

- ヨミガナ検索
  - 漢字・送り仮名の違いを吸収
- 同義語展開：別名をカバー
- 電話番号検索
  - 半角全角・記号の有無・記号の違いを吸収
- ワイン名検索
  - 外来語のカタカナ表記の違いを吸収

表記ゆれ：参考情報

文字の正規化方法

正規化：かなの同一視

- unify_kana
- ひらがなとカタカナを区別しない
- ↓は同じ
  - あいうえお
  - アイウエオ

正規化：濁点の同一視

- unify_sound_mark
- 濁点・半濁点の有無を区別しない
- ↓は同じ
  - はひふへほ
  - ばびぶべぼ
  - ぱぴぷぺぽ

正規化：大文字・小文字の同一視

- unify_kana_case
- 大文字・小文字を区別しない
- ↓は同じ
  - やゆよ
  - ゃゅょ

正規化：ハイフンっぽい文字の同一視

- unify_hyphen
- ハイフンっぽい文字をハイフンへ
  - ハイフン：U+002D
- ハイフンっぽい文字：
  - ‐‑‒–—―-⁻₋−

正規化：長音記号っぽい文字の同一視

- unify_prolonged_sound_mark
- 長音記号っぽい文字を長音記号へ
  - 長音記号：U+30FC
- 長音記号っぽい文字：
  - ー﹘—―‐‒–

正規化：ハイフン・長音記号っぽい文字

- unify_hyphen_and_prolonged_sound_mark
- ハイフン・長音記号っぽい文字を
  ハイフン（U+002D）へ
- ハイフンっぽい文字：
  - ‐‑‒–—―⁃－
- 長音記号っぽい文字：
  - ー﹘—㇐ㅡ－

正規化：中点っぽい文字の同一視

- unify_middle_dot
- 中点っぽい文字を中点へ
  - 中点：U+00B7
- 中点っぽい文字
  - ･････ ･･

正規化：ヴァ→バ

- unify_katakana_v_sounds
- ヴァ行をバ行へ
- ↓は同じ
  - ヴァヴィヴヴェヴォ
  - バビブベボ

正規化：ヴァ行→ブ

- unify_katakana_bu_sound
- ヴァ行をブへ
- ↓は同じ
  - ヴァヴィヴヴェヴォ
  - ブブブブブ

正規化：MySQL 8.0

- 日本語用COLLATIONを追加
  - utf8mb4_ja_0900_as_cs
  - COLLATION：文字の順序のルール
  - 順序なので等価比較機能もある
- 最新Mroongaは対応済み

正規化：PostgreSQL 10

- ICUベースのCOLLATION対応
  - ICU：Unicode処理ライブラリー
  - COLLATION：文字の順序のルール
  - 順序なので等価比較機能もある

あいまい検索：typo

テノクロジー

あいまい検索：typo対策

fuzzy検索

fuzzy検索

- 似ている文字列を検索
  - 似ている＝編集距離が小さい
- インデックスを使って検索

編集距離

- Aを何回編集するとBになるか
- 編集：
  - 挿入・削除・置換
  - 置換を禁止するケースもある
- 編集回数が距離

編集距離例：置換あり

- A:テノクロジー
- 置換：ク↔ノ
- B:テクノロジー

編集距離：1

編集距離例：置換なし

- A:テノクロジー
- 削除：ク：テノロジー
- 挿入：ノ：テクノロジー
- B:テクノロジー

編集距離：2

fuzzy検索：Mroonga：テーブル定義

```
CREATE TABLE tags (
  name varchar(255),     -- 検索対象
  FULLTEXT INDEX (name) -- fuzzy検索対応インデックス
    COMMENT 'tokenizer "none"'
) ENGINE=Mroonga DEFAULT CHARSET=utf8mb4;
```

fuzzy検索：Mroonga：データ例

```
INSERT INTO tags
  VALUES ('テクノロジー'),
         ('テクニカル');
```

fuzzy検索：Mroonga：検索

```
SELECT name,
       MATCH(name) AGAINST(...↓と同じ内容...) AS score
  FROM tags
  WHERE MATCH (name)
    AGAINST (CONCAT('*SS fuzzy_search(name, ',
                    mroonga_escape('テノクロジー' AS script),
                    ', ',
                    '{"with_transposition": true,
                      "max_distance": 4}')')
             IN BOOLEAN MODE);
-- テクノロジー | 4
-- テクニカル   | 1
```

fuzzy検索：PGroonga：テーブル定義

```
CREATE TABLE tags (
  name text -- 検索対象
);
```

fuzzy検索：PGroonga：インデックス定義

```
CREATE INDEX tags_search ON tags
  USING PGroonga (name)
  WITH (tokenizer='');
```

fuzzy検索：PGroonga：データ例

```
INSERT INTO tags
  VALUES ('テクノロジー'),
         ('テクニカル');
```

fuzzy検索：PGroonga：検索

```
SELECT name, pgroonga_score(tableoid, ctid)
  FROM tags
  WHERE name &`
    ('fuzzy_search(name, ' ||
                  pgroonga_escape('テノクロジー') || ',
                  {"with_transposition": true,
                   "max_distance": 4})');
-- テクノロジー | 4
-- テクニカル   | 1
```

まとめ：あいまいな情報

- 人が用意した情報はあいまい
  - クエリーも検索対象も
- あいまいでも必要な文書を見つける
  - 人よりも機械ががんばる

まとめ：あいまいな検索

- 全文検索エンジンを活用して実現
  - あいまい検索機能を提供しているはず

まとめ：全文検索エンジン

- 普通の全文検索エンジン
  - 独自の使い方
- Mroonga・PGroonga
  - SQLで使える→開発しやすい

まとめ：ヨミガナ検索

- 漢字・送り仮名の違いを吸収
  - 焼肉・焼き肉・焼きにく
  - 高橋・髙橋
- MeCabで自動化できる
  - 辞書により失敗することはある
- 同義語には対応できない
  - 同義語展開と併用

まとめ：同義語展開

- 別名に対応：「ぼたん鍋」と「猪鍋」
- なにを同義語とするかが難しい
  - システム依存度が高い
  - ある程度は自動化できる
  - 手動でのメンテナンスも必要

まとめ：電話番号検索

- 半角全角・記号有無・記号違いを吸収
-  注意 
  - どんな検索対象でもゆるくてよいわけではない
  - 誤ヒットも増えてしまう
  - 電話番号ならここまでゆるくてもOKというだけ

まとめ：ワイン名検索

- 外来語のカタカナ表記の違いを吸収
-  注意 
  - どんな検索対象でもゆるくてよいわけではない
  - 誤ヒットも増えてしまう
  - ワイン名ならここまでゆるくてもOKというだけ

補足：ゆるくするなら重みも考慮

- 検索対象を限定できないがゆるくしたい
- ゆるくない検索と組み合わせて重み調整
  - ゆるくない方がゆるい方より重要

重み調整例：Mroonga

```
SELECT
  MATCH(...) AGAINST(...ゆるくない...) * 10 +
  MATCH(...) AGAINST(...ゆるい...) AS score
  ...;
```

重み調整例：PGroonga

```
SELECT pgroonga_score(tableoid, ctid)
  WHERE
    ... &@~ ('...ゆるくない...',
             ARRAY[10],
             'pgroonga_index')::pgroonga_full_text_search_condition OR
    ... &@~ '...ゆるい....';
```

まとめ：fuzzy検索

- typoしても本来のキーワードを推測
- 活用方法：
  - ヒットしなかったときの「もしかして」の実装
  - 入力補完候補

参考情報：リッチな全文検索システム

MySQL・PostgreSQLだけで作る
高速でリッチな全文検索システム

須藤功平　　株式会社クリアコード

db tech showcase Tokyo 2017
2017-09-07

https://slide.rabbit-shocker.org/authors/kou/db-tech-showcase-tokyo-2017/

扱わなかったあいまい検索1

近傍検索

近傍検索

- 指定したキーワード間に
  違う単語が含まれていてもマッチ
- 「みそラーメン」で検索：
  - 「みそバターラーメン」：マッチ

扱わなかったあいまい検索2

quorumマッチ

quorumマッチ

- 閾値以上の要素がマッチしたらマッチ
- 閾値2と
  「MySQL MariaDB Percona」で検索：
  - 「MySQLとMariaDBの比較」：マッチ
  - 「MySQLとPostgreSQLの比較」：マッチしない

扱わなかった話題

- 運用について
  - 障害対策・レプリケーション
- チューニング

サポートサービス紹介

- 導入支援（設計支援・性能検証・移行支援・…）
- 開発支援（サンプルコード提供・問い合わせ対応・…）
- 運用支援（障害対応・チューニング支援・…）

問い合わせ先：

https://www.clear-code.com/contact/?type=groonga