

# Redmine検索の 未来像

須藤功平

株式会社クリアコード

redmine.tokyo第17回  
2019-11-02





# 全文検索プラグイン

[https://github.com/clear-code/redmine\\_full\\_text\\_search](https://github.com/clear-code/redmine_full_text_search)



# 従来の検索システム

- SQLの標準機能で実現
  - 比較条件+LIKE
- 例：
  - プロジェクト絞り込み：等価条件
  - コメント検索：LIKE



# 未来の検索システム

- 全文検索エンジンで実現
  - 全文検索だけでなく比較条件もすべて
- 例：
  - プロジェクト絞り込み：等価条件
  - コメント検索：全文検索



# 従来の検索システムの課題

# 速度とソート



# 速度

- コメント増加→検索時間増加
- 検索への期待
  - **すばやく**必要な情報を見つけない
- 遅いと期待に応えられない



# ソート

- 更新時刻のみ
- 検索への期待
  - すばやく**必要な**情報を見つけない
- 最新の情報 ≠ 必要な情報
- 必要な順にソートしないと期待に応えられない



# 未来の検索システム

- 速度
  - コメント増加→検索時間超微増
  - **すばやく**必要な情報を見つけられる
- ソート
  - 更新時刻+適合度
  - すばやく**必要な**情報を見つけられる





## 高速さと高精度



# 未来の検索システムのさらなる改善点

- 検索対象を追加
- クリックで絞り込めるUI
- 表現の揺らぎに対応
- 定量的な検索機能の評価基盤



# 検索対象を追加

- メタデータ
  - ステータス・トラッカー・ユーザー・...
- 添付ファイル・リポジトリ内のファイル
  - PDF・オフィス文書からもテキスト抽出
  - アーカイブも展開して再帰的にテキスト抽出



# メタデータで検索

ホーム マイページ プロジェクト 管理 ヘルプ ログイン中 | [new](#) | [工数](#) | [個人設定](#) | [ログアウト](#)

ClearCode 検索:  ClearCode

概要 活動 チケット 作業時間 Wiki リポジトリ インポート 設定

検索

[→ オプション](#)

0 0.0%

すべて (3083) チケット (2776) ニュース (0) 文書 (0) 更新履歴 (24) Wikiページ (206) メッセージ (0) プロジェクト (0) 変更 (73)

並び替え対象

- ★ スコア
- 🕒 更新日時
- 並び順
- 👤 昇順
- 👤 降順

チケットのステータス

- 終了 (2744)
- 却下 (20)
- 実行中 (7)
- 解決 (4)
- 新規 (3)

トラッカー

- タスク (1712)
- ログ (1052)

ユーザー

- Sutou Kouhei (676)
- Horimoto Yasuhiro (588)
- Hayashi Kentaro (574)
- 南 樹一郎 (277)
- 結城 洋志 (248)

1 misc:print/groonga-support/groonga.pdf#8371b01ccffba898ff7b7516edf241eb9e49fb592 (206)

ストア付き検索エンジン [groonga](#) と [Graln](#) が 2 は、オープンソースのカラムストア機能付き全文検索エンジンです。 [groonga](#) を使うことにより、高性能タスクを更新できる [groonga](#) が使われています。また、日付や放送時間、出演者などのメタ情報によるファセット検索も、 [groonga](#) の機能が活かされています。 [groonga](#) が活用されています。膨大な量の店舗情報の中から指定座標に近い店舗をリストアップするなどの高度な検索も、 [groonga](#) では非常に

2013/02/26 11:26

2 Wiki: Groonga-command-filterを使ってGroongaのダンプから不具合の再現に必要なテーブル・カラムのデータのみを抽出するには (206)

タを用いた作業を効率よくすめられるようにしましょう。 [Groonga command filter](#) を使って [Groonga](#) のダンプから不具合の再現に必要なテーブル・カラムのデ

2010/10/17 15:48

3 misc:doc/groonga-misc/groonga-daily-report.md#616c2c1975a5d48897a31198da6fe4c569de477f (201)

2014/03/05 14:53

4 misc:print/groonga-support/groonga-outline.pdf#8371b01ccffba898ff7b7516edf241eb9e49fb592 (201)

2013/02/26 11:26

5 misc:print/groonga-support/groonga.si#371b01ccffba898ff7b7516edf241eb9e49fb592 (201)

2013/02/26 11:26

6 Wiki: LaravelでPostgreSQLとPGroongaを使って日本語全文検索を実現する方法 (111)

7 [PGroonga](#) LaravelでPostgreSQLとPGroongaを使って日本語全文検索を実現する方法 [# はじめに](#) ※この記事は、Laravelを使った開発の経験がある人を対象と分にあるでしょう。以前、 [Ruby on Rails](#) で作ったアプリケーションから [PGroonga](#) を使って日本語全文検索機能を実現する方法 [\[https://www.clear-code.com/blog/2014/06/26/\]](#) そのPHP版として、Laravelで作ったブログ風のアプリケーションに [PGroonga](#) [\[https://groonga.org/\]](#) を使った日本語全文検索機能を組み込む方法をご紹介します

2017/06/26 15:41

8 Wiki: GroongaをRubyから使う方法 (111)

9 [Groonga](#) \* [ActiveGroonga](#) \* [groonga-client](#) \* [groonga-client-rails](#) \* [groonga-client-model](#) \* [groonga/groonga](#) \* [fluent-plugin-groonga](#) たぶんこんだけある。それぞれの説明人が選べるようになるよ。 [# \[Groonga\] \[Ruby\] GroongaをRubyから使う方法 \[Groonga\] \[https://groonga.org/ja/\]](#) はクリアコードが開発に参加している



# クリックで絞り込めるUI

- ユーザー：なにを探しているかわからない
  - 検索システムとの対話の中で見つける
- 対話：検索結果+次のクエリーを提案
  - ユーザーと検索システムと一緒に答えを探す
- 提案されたクエリーで絞り込み



# 表現の揺らぎに対応

- 表現の揺らぎ：
  - 同義語：ネジとビス
  - 異表記：ネジと螺子
- クエリー拡張：
  - 検索前に自動でクエリーを拡張
  - ネジ→ネジ OR ビス OR 螺子

# クエリー拡張：Groonga→Groonga OR PGroonga

  
→ オプション

0.00s

並び替え対象

★ スコア  
🕒 更新日時

並び順

📈 昇順  
📉 降順

チケットのステータス

- 終了 (2744)
- 却下 (20)
- 進行中 (7)
- 解決 (4)
- 新規 (3)

トラッカー

- タスク (1712)
- ログ (1052)

ユーザー

- Sutou Kouhei (678)
- Horimoto Yasuhiro (600)

すべて (3083) チケット (2778) ニュース (0) 文書 (0) 更新履歴 (24) Wikiページ (208) メ

[misc:print/groonga-support/groonga.pdf@371b01ccffba8698ffb7516edf241eb9e49fb592](#) (204)  
ストア付き検索エンジン groonga<< グルンガ >>は、オープンソースのカラムストア機能付き全文検索エンジンを更新できる groonga が使われています。また、日付や放送時間帯、出演者などのメタ情報において、 groonga が活用されています。膨大な量の店舗情報の中から指定座標に近い店舗をリスト  
2013/02/28 11:28

[Wiki: Groonga-command-filterを使ってGroongaのダンプから不具合の再現に必要なテーブルタを用意する作業を効率よくすめられるようにしましょう。 groonga-command-filterを使ってGroongaのダ](#)  
2019/10/17 18:48

[misc:doc/groonga-misc/groonga-daily-report.md@616c2c1975a5d48997a31198da6fe4c569de4](#)  
2014/03/06 14:53

[misc:print/groonga-support/groonga-outline.pdf@371b01ccffba8698ffb7516edf241eb9e49f](#)  
2013/02/28 11:28

[misc:print/groonga-support/groonga.ai@371b01ccffba8698ffb7516edf241eb9e49fb592](#) (201)  
2013/02/28 11:28

[Wiki: LaravelでPostgreSQLとPGroongaを使って日本語全文検索を実現する方法](#) (111)  
# [Groonga] LaravelでPostgreSQLとPGroongaを使って日本語全文検索を実現する方法 ## はじめに ※ご自分にあるでしょう。以前、[Ruby on Railsで作ったアプリケーションからPGroongaを使って日本語全文] そのPHP版として、Laravelで作ったブログ風のアプリケーションに[PGroonga] (https://pgroonga.github.io)   
2017/06/26 15:41



# 表現の揺らぎのメンテナンス

- 手動はツライ。。。
  - 誰がやるの？
  - いつやるの？
- 自動化しないと陳腐化





自動化中



# 自動化方法

- 既知の情報を収集
  - 例：Wikipediaのリダイレクト情報
  - 例：NEologdの情報
- Redmine内の情報から学習



# 既知の情報を収集

- メリット：楽
  - スクリプトを準備する予定
- デメリット：
  - 一般的な言い回ししか集まらない
  - ドメイン固有の言い回しが無い



# Redmine内の情報から学習

- メリット：
  - ドメイン固有の言い回しに対応可
- デメリット：
  - 難しい



# 学習：検証中

1. テキスト情報を抽出
  - 全文検索用に抽出した情報を活用
  - 自然言語っぽいものだけ選別して行に分割
2. SentencePieceでトークナイズ
  - MeCabだとドメイン固有の未知語に対応できない
3. BERT/fastTextでベクトル化
4. 類似単語を揺らぎとして抽出



# 学習：現状（伸びしろしかない）

## クエリー展開一覧

[+ 新しいクエリー展開](#)

フィルタ

groonga 連絡します



## クエリー展開結果

((Groonga) OR (PGroonga)) ((をお伝えします) OR (提案します) OR (支援します) OR (確認します) OR (調整します) OR (連絡します))

展開元 ^	展開先	作成日	更新日			
込むこと	を学ぶこと	2019/08/20 15:06	2019/08/20 15:37	表示	編集	削除
込むこと	込むこと	2019/08/20 15:06	2019/08/20 15:37	表示	編集	削除
追加して	を削除して	2019/08/20 15:06	2019/08/20 15:37	表示	編集	削除
追加して	統一して	2019/08/20 15:06	2019/08/20 15:37	表示	編集	削除

### 管理

- プロジェクト
- ユーザー
- グループ
- ロールと権限
- トラッカー
- チケットのステータス
- ワークフロー
- カスタムフィールド
- 選択肢の値
- 設定
- LDAP認証
- クエリー展開一覧
- プラグイン



# 定量的な検索機能の評価基盤

- 定性的な評価（例：インタビュー）
  - 体感で速くなった！
  - 体感で見つかりやすくなった！
- 定量的な評価（例：ログ分析）
  - 利用ユーザーが○人
  - ユーザーが必要な情報を取得できた割合が○%

ログを追加し定量的な評価が可能に



# 定量的な評価基盤：現状

- Railsのログに記録
- スクリプトでログを解析してレポート





# 定量的な評価基盤：未来

- RDBMSに記録
- 管理画面でリアルタイムにレポート

未来の検索システムの  
近い未来



# さらなる高速化

- 常に1秒未満を目指す
- 検索システムとの対話ストレスをなくす
- インクリメンタル検索も現実的に
  - パカパカしてうるさいかもしれない
  - Googleはこの機能を捨てた
  - 定量的な評価をしながら取捨選択する予定



# 入力補完

- 入力中にクエリーを提案
  - Googleはこの機能にシフトした
- 課題：良質な補完候補の用意
  - Wikipediaなど一般的なデータを活用
  - Redmine内のデータから機械学習で生成
  - 極秘データの扱いに注意が必要



# テキスト抽出可能ファイルの追加

- .msg (Outlookのファイル)
- SVGファイル
- CADファイル
- ...



## 全文検索プラグイン

[https://github.com/clear-code/redmine\\_full\\_text\\_search](https://github.com/clear-code/redmine_full_text_search)



# 一緒に未来を作ろう！

- 仕事として一緒に開発
  - あなた：お金と要望とデータを提供
  - クリアコード：開発・コンサルティング
- 成果は自由なソフトウェアとして公開

連絡先

<https://www.clear-code.com/contact/>