

鮮度と精度を重視した 全文検索エンジンSenna



<http://qwik.jp/senna/>

(有)未来検索ブラジル
末永 匡

システム制御情報学会セミナー2007
「ウェブを守る, ウェブを活用する」にて発表



はじめに

- 自己紹介
 - 末永 匡と申します
 - (有) 未来検索ブラジル所属
 - 全文検索エンジンSennaの開発
 - Webサービスの開発



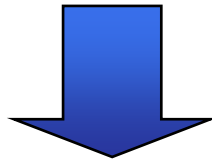
背景

- いわゆるCGMの台頭
 - 2ちゃんねる
 - mixi
 - Wikipedia
 - etc...
- これらに共通する特徴は...



CGMに対応した検索エンジン

- CGMに共通する特徴
 - 頻繁な更新・常時更新
 - 巨大な文書量



対応した検索エンジンが求められる



頻繁な更新・常時更新

- コンテンツが頻繁かつ常に更新される
- 更新されたものをすぐに検索したい！というニーズ
 - 2ちゃんねるにおける地震関係の掲示板
 - 地震直後に検索したい！
- 文書の追加更新が低速だったり、不可能である検索エンジンは不向き

更新が高速なエンジンが求められる



巨大な文書量

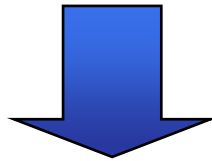
- 文書量がとっても多い！
 - 頻繁な更新・常時更新の結果として
 - ユーザが、自分が見たいコンテンツにたどり着くことが困難
 - コンテンツ選別手段として検索のニーズが高まる
- 「京都」で検索して「東京都」が検索対象となるような検索エンジンは不向き

精度が高い検索エンジンが求められる



CGMに対応するためには？

- CGMに共通する特徴
 - 頻繁な更新・常時更新
 - 巨大な文書量



鮮度・精度の高い検索エンジン



構成

- **全文検索の基礎について**
- **全文検索エンジンSennaについて**
- **2ちゃんねる検索について**



全文検索の基礎について

- 用語説明
- 検索エンジンにおける基本性能の尺度
- 検索速度・更新速度のトレードオフ
- 適合率・再現率のトレードオフ
- フレーズ検索でのトレードオフ



用語説明

- 全文検索
 - 全文検索エンジン
 - インデックス
-
- 以上3つの用語について説明



全文検索

- 全文検索：文書に含まれるテキスト全体を対象とする検索のこと。
 - Yahoo! / Google
 - デスクトップ検索



全文検索エンジン・インデックス

- 全文検索エンジン
 - ある文書集合を与えたときに、その文書集合を対象とした全文検索を行うシステム
- インデックス
 - 全文検索を行うために必要な補助情報が格納されているもの
 - 全文検索エンジンは、一般的にインデックスを作成し、そのインデックスに基づいて検索を行う



検索エンジンにおける基本性能の尺度

- 検索速度
 - 更新速度
 - 文書容量
 - 適合率
 - 再現率
-
- 文書容量には軽くしか触れません
 - 1台のマシンでどれだけの文書を検索できるか



検索速度と更新速度

- **検索速度**
 - 検索応答時間
 - 検索スループット
 - 主にユーザの待ち時間に影響する
- **更新速度**
 - 文書の新規作成時における処理速度
 - 既存文書の更新・削除時における処理速度
 - 主に検索結果の鮮度に影響する

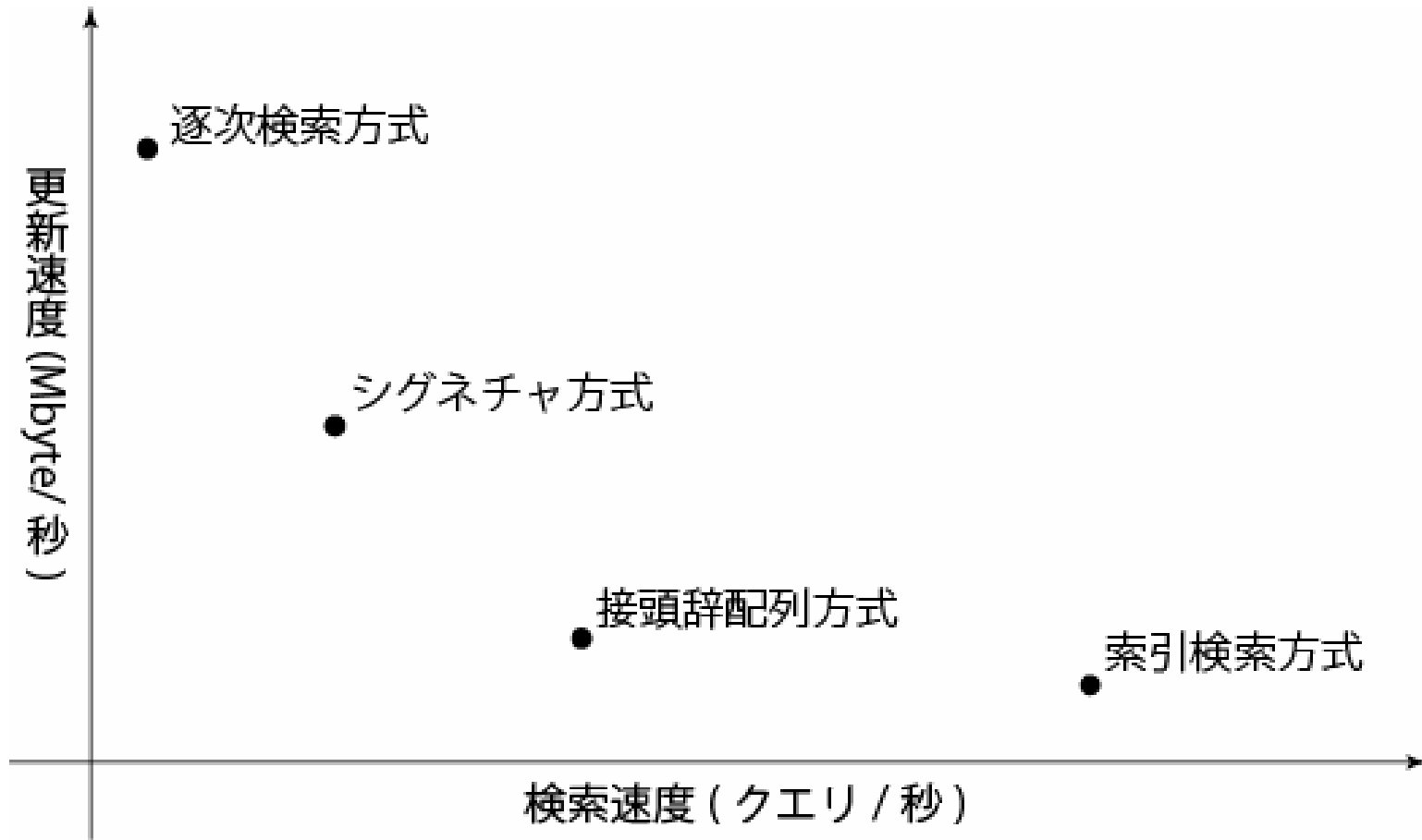


検索エンジンのアルゴリズム

- 逐次検索方式(grep)
- シグネチャ方式
- 接尾辞配列方式
- 索引検索方式



各種方式の特徴



検索速度と更新速度はトレードオフ関係



逐次検索方式

- 対象文書を逐次スキャンする
 - grepと同様
 - 検索時に文書本体のみあればよい
 - インデックスなどの余分なデータが必要ない
- 更新速度：高
 - インデックス等を作成しないため
- 検索速度：低
 - データを逐次スキャンするため



索引検索方式

- ある単語が、どの文書にあるかを保持
 - 「転置インデックス」というデータ構造
 - (例) 書籍巻末にある用語索引

検索対象語	文書番号列
インド	1, 10, 12
インドア	5, 10, 11
インドネシア	10

- 更新速度：低
- 検索速度：高



まとめ：検索速度と更新速度のトレードオフ

- 検索速度と更新速度にはトレードオフ関係がある
 - 逐次検索方式
 - 検索速度：低
 - 更新速度：高
 - 索引検索方式
 - 検索速度：高
 - 更新速度：低
- 今後、索引検索方式（転置インデックスの利用）を前提としてお話しします



検索エンジンにおける基本性能の尺度

- 検索速度
- 更新速度
- 文書容量
- 適合率
- 再現率

それぞれの要素同士がトレードオフ関係

検索ノイズ・検索漏れ

- 検索エンジンを使う目的
 - 適合する文書を見つけたい！
- 検索ノイズ
 - (例) 「インド」 → 「インドネシア」
「京都」 → 「東京都」
「先生」 → 「この先生きのこるため」
- 検索漏れ
 - (例) 「打ち合わせ資料」 → × 「打合せ資料」
- 精度を表す尺度が必要



適合率・再現率とは

- 適合率・再現率
 - 検索エンジンの精度をあらわす尺度
- 適合率
 - 「検索ノイズ」に対する尺度
- 再現率
 - 「検索漏れ」に対する尺度



適合率

- 適合率とは
 - 検索された文書の中で、
検索しなかった文書(=適合する文書)の割合
 - 簡単にいうと、1-ノイズ率
 - 高適合率 = ノイズが少ない = 精度が高い
 - (例) 「インド」で検索して40件の文書がhit
 - 「インド」に関する文書 30件
 - 「インドネシア」に関する文書 10件
 - 適合率 = $30 / 40 = 0.75$



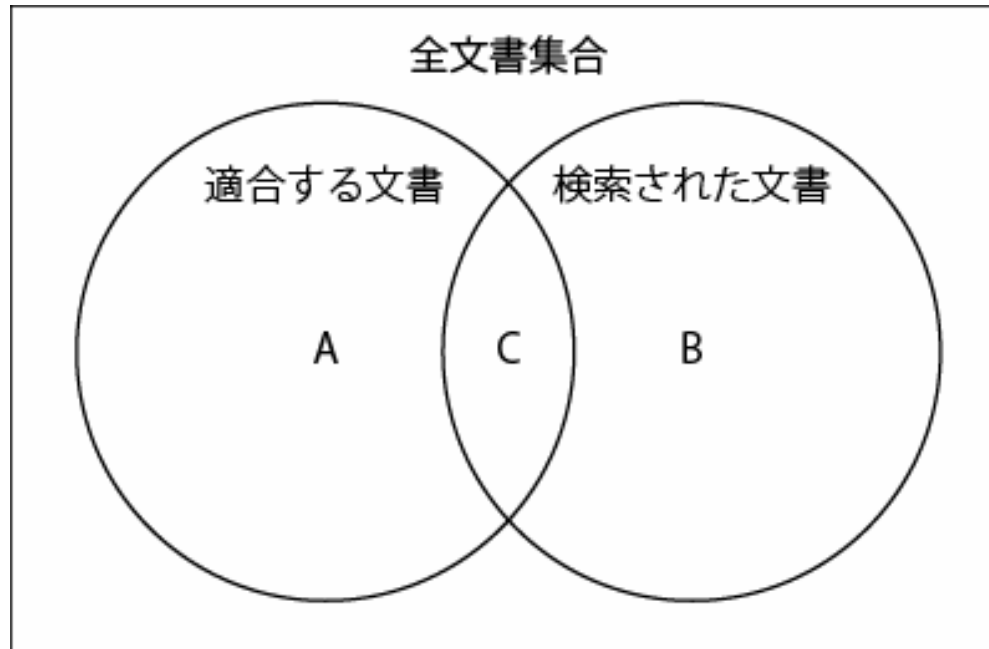
再現率

- 再現率とは
 - 適合する文書の中で、実際に検索された文書の割合
 - 簡単にいうと、1-検索漏れ率
 - 高再現率 = 検索漏れが少ない
 - (例) 「打ち合わせ資料」で検索して30件hit
 - 「打ち合わせ資料」を含み検索された文書 30件
 - 「打合せ資料」を含み検索された文書 10件
 - 再現率 = $30 / (30 + 10) = 0.75$



適合率・再現率

- 適合率 : C / B
- 再現率 : C / A
- どちらも高いほうがよい
 - 検索ノイズも検索漏れも少ないほうがよい



適合率・再現率のトレードオフ

- 適合率のみを高めたい
 - 適合しない可能性が少しでもある文書を、検索結果から除外
 - 再現率の低下 (= 検索漏れの発生)
- 再現率のみを高めたい
 - 適合する可能性が少しでもある文書を、検索結果として採用
 - 適合率の低下 (= 検索ノイズの発生)
- 両者はトレードオフ関係



適合率・再現率へのニーズ

- 高い適合率が求められるケース
 - 大規模な文書集合を対象とした検索
 - (例) Webサーチエンジン
 - 「ノイズが多くて欲しい情報が探せない」
- 高い再現率が求められるケース
 - (例) 特許文書検索
 - 「調査漏れがあると特許侵害リスクを負う」

要求によってどちらを優先するかを選択



転置インデックスの形式と精度

- 転置インデックス（おさらい）
 - ある単語が、どの文書にあるかを保持
 - (例) 書籍巻末にある用語索引
- 単語の選び方で精度が変わる
 - 形態素
 - 形態素：言語の中で意味を持つ最小単位
 - 適合率が高まる
 - N-gram
 - N-gram：任意のN文字幅の部分文字列
 - 再現率が高まる



形態素を単語として採用

- 形態素単位での転置インデックス
 - 形態素に一致しない語が検索されない
 - 適合率：高
 - Googleも形態素を単語として採用

単語	文書番号列
インド	1, 10, 12
インドネシア	5, 10, 11
京都	1, 2, 3, 6, 10
東京都	1, 2, 4, 5, 11
先生	10



N-gramを単語として採用

- N文字の部分文字列単位でインデックス
 - N文字の部分文字列が一致すれば検索される
 - 再現率：高

部分文字列	文書番号列
イン	1, 3, 5, 9
ンド	1, 3, 5, 9
ドネ	3, 9
ネシ	3, 9
シア	3, 9



まとめ：適合率と再現率のトレードオフ

- 適合率と再現率にはトレードオフ関係がある
 - 形態素を単語とした転置インデックス
 - 適合率：高
 - 再現率：低
 - N-gramを単語とした転置インデックス
 - 適合率：低
 - 再現率：高



フレーズ検索

- ある単語列が指定された順番に出現する文書を検索したいという検索のこと
- (例) 「システム」「制御」「情報」が連続して出現する文書のみを検索したい
 - OK：システム制御情報学会セミナー
 - NG：計測自動制御学会 システム・情報部門
- フレーズ検索の実現方法が複数存在
 - 手法によって、基本性能の尺度におけるトレードオフ関係が存在



フレーズ検索(2)

- 例えば、以下のような文書があると仮定
- **文書番号10**
 - 「システム 制御 情報 学会 セミナー」
- **文書番号12**
 - 「計測 自動 制御 学会 システム 情報 部門」

検索対象語	文書番号列
システム	10, 11, 12
制御	10, 11, 12
情報	5, 10, 12



フレーズ検索(3)

検索対象語	文書番号列
システム	10, 11, 12
制御	10, 11, 12
情報	5, 10, 12

- それぞれの単語(システム, 制御, 情報)を全て含むレコードを検索する
 - 適合しない文書番号12が検索結果に入る
 - 適合率：低



フレーズ検索(4)

検索対象語	文書番号列
システム	10, 11, 12
制御	10, 11, 12
情報	5, 10, 12

文書データ

- それぞれの単語(システム, 制御, 情報)を全て含むレコードを検索し、検索結果のうち「システム」「制御」「情報」が連続して出現するもののみを検査する
 - 適合率：高
 - 検索速度：低



完全転置インデックス

- 文書番号のみならず、文書内での検索対象語の出現位置も保持

検索対象語	(文書番号, 出現位置)列
システム	(10, 1), (11, 4), (12, 5)
制御	(10, 2), (11, 1), (12, 3)
情報	(5, 3), (10, 3), (12, 6)

- 高速なフレーズ検索が可能
 - 文書データとの照合が不要となるため



完全転置インデックスの欠点

- 文書番号のみならず、文書内での検索対象語の出現位置も保持
 - 適合率：高
 - 検索速度：高
- 多くの情報をインデックスに持つ
 - 更新速度：低
 - 文書容量：低

トレードオフ関係が生じている



まとめ：フレーズ検索

- フレーズ検索
 - 転置インデックスのみ
 - 適合率 : 低 検索速度 : 高
 - 転置インデックス + 全件検査
 - 適合率 : 高 検索速度 : 低
 - 完全転置インデックス
 - 適合率 : 高 検索速度 : 高
 - 更新速度 : 低 文書容量 : 低



まとめ：全文検索の基礎について

- 検索エンジンにおける基本性能の尺度
 - 検索速度, 更新速度, 文書容量, 適合率, 再現率
- 上記尺度間はトレードオフ関係
- どの尺度を重視するかによって、検索エンジンの設計が決まる
 - 鮮度と精度を重視：
更新速度と適合率を重視



構成

- 全文検索の基礎について
- **全文検索エンジンSennaについて**
- 2ちゃんねる検索について



Senna

- (有)未来検索ブラジルを中心に開発されている全文検索エンジン
- オープンソースライセンス(LGPL)
 - 無償で商用利用可



オープンソースの検索エンジン

- オープンソース検索エンジンの台頭
 - Apache Lucene
 - Wikipedia(英語版)
 - Hyper Estraier
 - mixi
 - Senna
 - 2ちゃんねる検索
- 全て索引検索方式を利用

今回はSennaの特徴をご紹介します



Sennaの三大特徴

- 高速
 - 高精度
 - 高柔軟性
-
- 順に説明していきます



高速

- 検索が高速
 - 完全転置インデックスを採用
 - 高速な検索・フレーズ検索ができる
- 更新が高速
 - 既存インデックスへの追加・変更が高速
 - 鮮度の高い情報をインデックス化できる
- 高速化には実装上の工夫が不可欠



高速化の工夫

- 転置インデックスに適したバッファ機構
 - インデックスの一部をメモリにキャッシュ
 - I/O負荷の減少
- 検索時インデックスの排他制御が不要
 - 高い並列性
- インデックス自動再配置 (=デフラグ)
 - インデックスの不要な領域を再配置
 - 更新・削除による性能劣化を防ぐ

高速な更新・検索を志向した実装



高精度

- Sennaは精度重視の設計
 - 速度は、精度を阻害しない範囲で追求
- 具体的には
 - 1. 適合率
 - 2. 再現率の順番で重視
- 大規模なコンテンツを対象とした検索では、適合率が重要となるため



適合率と再現率の両立

- 適合率と再現率はトレードオフの関係
- どちらも高いほうがよい
- Sennaでは両尺度を高める方式を採用



Sennaの工夫

- 適合率と再現率を両立する独自方式
 - 形態素の部分一致検索が可能なデータ構造
 - 「京都」で検索して「東京都」を含む文書が検索可能
 - 具体的には次のスライドで説明



Sennaの転置インデックス

- 通常の転置インデックス

- 検索対象語の完全一致

検索対象語	文書番号列
東京都	1, 10, 12
京都	2, 5, 11
大阪府	10

- Senna

- 検索対象語の部分一致検索が可能

適合率が落ちてしまうのでは...?



Sennaの部分一致検索

- 適合率が落ちるとなぜうれしくないか？
 - 検索結果にノイズが多くて
欲しい検索結果が見つからない
- Sennaの考え方
 - 検索漏れが多い場合のみに限って
部分一致検索をすればよい
 - 部分一致の検索結果が目立たなければよい
- それぞれに対応した処理が選択可能



部分一致検索(1)

- Sennaでは、検索結果がn件以下だった場合のみ部分一致検索を行うことが可能
 - 検索結果が少ない場合、検索漏れの存在が疑われる
 - 部分一致検索の実行によって、検索漏れの救出
 - (例) 「インド」で検索して0件ヒットだから、「インドネシア」を含む文書も検索する

適合率を阻害せずに再現率を向上



部分一致検索(2)

- 検索結果ごとに、スコアという値を計算
 - 文書がどれだけ適合しているかを表す
- 通常、検索結果はスコア順で並べる
- Sennaでは、部分一致による検索結果のスコアを相対的に下げることが可能
 - 検索されるが、検索結果の後のほうに表示
 - ユーザは検索結果の最初のほうしか見ない

主観的な適合率を保ちつつ再現率を向上



適合率をさらに高めるために

- Sennaは適合率を重視した検索エンジン
 - 部分一致検索を用いた適合率の向上を説明
- さらなる工夫を行っている
 - 検索文書の階層化：これから説明



文書単位での検索の問題点(1)

- (例)Webサーチエンジンで「淀屋橋 ラーメン」が検索された
 - ユーザは「淀屋橋のラーメン情報」を知りたい
- 「淀屋橋」と「ラーメン」が入っているページを検索結果としてよいか？



文書単位での検索の問題点(2)

- このようなブログは適合している？
 - 10月11日
今日淀屋橋に行った。...
 - 10月17日
自宅でインスタントラーメン食べた。...
- 同じページに「淀屋橋」も「ラーメン」も出現しているが、淀屋橋のラーメン情報はない

文書単位での検索では適合率が低い



段落・パラグラフ情報の保持

- 段落単位での検索が可能
 - 段落：文章中の任意の一部
 - (例) ブログのエントリごとの検索
- スコアの重み付けが可能
 - パラグラフ：段落中の任意の一部
 - パラグラフ単位で重み付けしたスコアを保持
 - (例) ブログのエントリごとの見出しに
単語が出現していたら高いスコアを付与

高精度な検索の実現



まとめ：高精度

- Sennaは精度重視の設計
 - 形態素を単語として採用した転置インデックス
 - 適合率を重視しつつ、再現率も高める
- Sennaは段落検索・スコアの重み付けが可能
 - 内容がいくつかに分割できる文書に対しても適合率の高い検索が可能



高柔軟性

- 多くの利用形態をサポート
- 柔軟な問い合わせ言語をサポート



Sennaの利用形態

- Sennaの利用形態
 - Sennaを部品として他のDBMSに組み込んだ形態
 - MySQL + Senna = Tritonn
 - PostgreSQL + Senna = Ludia
 - Sennaを単体で利用する形態
 - 単体のサーバ
 - ライブラリとしての利用



利用形態に応じた問い合わせ言語

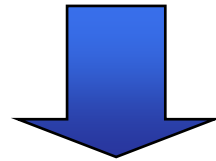
- DBMS組み込み
 - SQLを用いた全文検索問い合わせが可能
- Senna単体
 - SennaQLという問い合わせ言語
- SQL/SennaQLとも柔軟な問い合わせが可能

どうして柔軟な問い合わせが必要？



書誌情報の重要性

- 書誌情報：文書に付随した各種情報のこと
- 実用上の要求
 - 文書作成日で並べ替えて表示したい
 - 非公開フラグが立っている文書を除きたい
 - タイトルに検索語が登場するものを、内容のみに検索語が登場するものより優先的に表示したい



問い合わせ言語に柔軟性が必要



利用形態別での特徴紹介

- 利用形態ごとに特徴を紹介
 - Tritonn = MySQL + Senna
 - Ludia = PostgreSQL + Senna
 - SennaQL



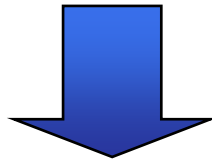
Tritonn : MySQL + Senna

- MySQLが文書を管理
 - SQLを用いた問い合わせ
(例) `SELECT * FROM table1
WHERE MATCH(col1)
AGAINST('クエリ');`
 - SQLの高い記述力 = 高柔軟性
 - 複雑な問い合わせが可能
 - 他カラムでの絞り込み・並び替え・グループ化
 - (例) あるユーザが、閲覧権限のある文書のみに対して全文検索を行い、その結果を日付順に並べたい



オリジナルの全文検索の問題点

- 日本語など、分かち書きしない言語が検索できない
- フレーズ検索が遅い
- インデックス更新が極端に遅い
- 全文検索と、他のインデックスとを組み合わせる検索できない



Sennaで解決



MySQLとSennaの比較表

実験データ : Wikipedia 英語版 458,713レコード 1088MB

	MySQL オリジナル	MySQL+ Senna
インデックスサイズ	109 MB	1028 MB
フレーズ検索 (‘united states’)	44.91 sec	0.40 sec
既存のレコードに インデックス付与	1,474 sec	1,808 sec
インデックス付与後に レコードを追加	28,182 sec	1,839 sec
order by 主キー	20.33 sec	0.89 sec
where 全文検索 and 主キー > 20万	6.55 sec	0.32 sec



Ludia : Pg + Senna

- **Ludia**

<http://www.nttdata.co.jp/services/ludia/index.html>

- (株)NTTデータが開発

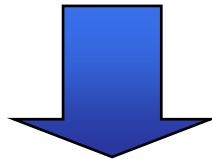
- SQLを用いた問い合わせ

(例) `SELECT * FROM table1
WHERE col1 @@ 'クエリ';`



GINの問題点

- PostgreSQL 8.2で全文検索が可能に
 - GINインデックス
- MySQLと同様の欠点
 - 日本語など、分かち書きしない言語が検索できない
 - フレーズ検索が遅い



Sennaで解決



SennaQL

- Sennaを単体で利用する場合の問い合わせ言語
 - Schemeベース
- Sennaの基本性能を生かすよう設計
 - DBMS組み込みの場合、組み込み元のアプリケーションとの連結部分でボトルネックが発生するケースが存在する
- 詳細な説明は今回省略いたします



以上がSennaの三大特徴です

- **高速**
 - **高精度**
 - **高柔軟性**
-
- **是非覚えてネ！**



Sennaが持つその他の機能

- UTF-8対応
- 純粋なN-gramインデックス作成機能
 - 高い再現率
- 関連文書検索
 - クエリに指定された文書と、内容が類似する文書を検索
- 近傍検索
 - 指定された複数の単語が、近傍に現れる文書を検索



導入実績

- タワーレコード商品検索
<http://search.tower.jp>

TOWER SEARCH
beta

- はてなキーワード検索
<http://search.hatena.ne.jp/keyword?mode=top>



- その他、続々導入中!!!!!!!!!!!!!!



今後の開発予定

- 分散版Senna (コードネーム Dicty)
 - スケールアウト可能に
 - SennaQLを用いたアプリケーションの場合
アプリケーションの改修が不要
 - 有償
- 基本性能の向上
 - メモリ管理機構の搭載
 - トランザクション対応



まとめ：全文検索エンジンSennaについて

- オープンソースの検索エンジン
- 三大特徴
 - 高速、よって、高鮮度
 - 高精度
 - 高柔軟性
- 鮮度と精度が高い実用的な検索エンジン
- 有償の分散対応バージョンを開発中



構成

- 全文検索の基礎について
- 全文検索エンジンSennaについて
- **2ちゃんねる検索について**



2ちゃんねる検索

- **2ちゃんねる/PINKちゃんねるを検索**
 - スレッドタイトル/本文 など
- **書き込まれたら1分以内に検索可能**
 - **鮮度の高い検索**
 - 2chをクローリングして即時インデックス反映
- **バックエンドにSennaを利用**
 - **精度の高い検索**
 - フロント9台 : バックエンド8台
 - ページビュー 9,000万/月 超



有料検索のねらい

- **有料検索サービス**
 - 広告主からの収益ではなく、ユーザからの直接収益を主とする
- **検索精度は収益源に左右されがち**
 - 検索インターフェースのデザイン
 - 検索結果のランキング
- **ユーザ視点での検索精度の向上を目指す**

ユーザからの課金による検索精度の確保



モリタポ

- 森田検索ポイント
 - ユーザからの利用料金徴収手段
 - 小額決済向けポイントサービス
 - 1モリタポ = 0.1円
 - 匿名性を重視
 - メールアドレスをユーザ識別手段とする
 - APIを通じて外部サービスでも利用可能

小額決済を利用したユーザ課金検索の実現



まとめ：2ちゃんねる検索

- 即時クローリング・更新
 - 鮮度を確保
- Sennaの採用
 - 精度を確保
- モリタポという小額決済の導入
 - ユーザ視点での精度を確保

鮮度と精度の高い検索サービスの実現



構成

- 全文検索の基礎について
 - 全文検索エンジンSennaについて
 - 2ちゃんねる検索について
- 以上3点についてお話をさせていただきました。



まとめ：全体

- CGMの台頭
 - 鮮度・精度の高い検索システムが求められる
- 鮮度・精度の高い検索システム
 - 検索エンジンの基本性能として
 - Senna
 - 高速・高精度・高柔軟性
 - 収益構造を含めたサービス全体として
 - 2ちゃんねる検索
 - モリタポ



発表終了

- ご清聴ありがとうございました
- Sennaに関する詳細な情報は、
<http://qwik.jp/senna/>
をご覧ください
- 是非是非ご利用ください!!!

