本当にその検索は自分が想像している検索になってますか？

~~*2020/02/18 JJUGナイトセミナー*~~

Jun Ohtani
@johtan

自己紹介

- フリーランスエンジニア(仮)
- *Apache Solr*入門(第2版まで)やデータ分析基盤構築入門の著者の一人
- *KibanaのAnalyze API UI plugin*の作者

今日話すこと

- 検索エンジンの仕組み
  - *Lucene/Elasticsearch*の仕組み
- 検索システムに出てくる要素とか

話さないこと

- 個別の検索エンジン、検索サービスの使い方
- 検索システムの運用
- 画像検索、音声検索

アジェンダ

- 検索とは?
- *Grep*と全文検索の違い
- *Lucene/Elasticsearch*の仕組み
- よりよい検索にするためには？

「Java」で検索したときに出てきてほしいものは？

どっちが出てきてほしいですか？

- あなたと*Java*、今すぐダウンロード
- *JavaScript*という言葉は狭義には*Mozilla*が仕様を策定し実装しているスクリプト言語を指す。

「*Exception*」で検索したときに出てきてほしいものは？

どれが出てきてほしいですか？

- *java.lang.Exception*
- *java.lang.IllegalArgumentException*
- *java.io.FileNotFoundException*

検索とは

検索とは、データの集合の中から目的のデータを探し出すことである。
また、目的のデータが存在しないことの確認行為も含まれる。

Wikipediaより https://ja.wikipedia.org/wiki/

どっちが出てきてほしいですか？

- あなたと*Java*、今すぐダウンロード
- *JavaScript*という言葉は狭義には*Mozilla*が仕様を策定し実装しているスクリプト言語を指す。

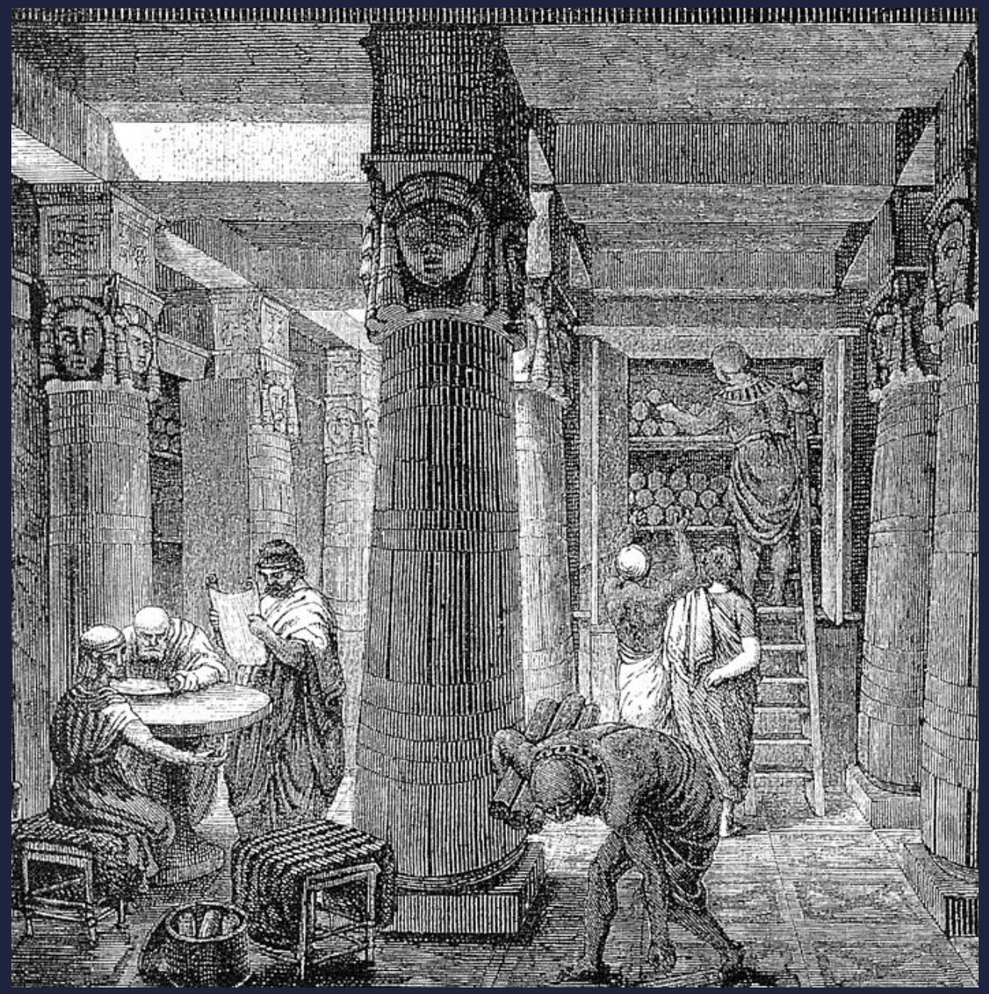
どれが出てきてほしいですか？

- *java.lang.Exception*
- *IllegalArgumentException*
- *FileNotFoundException*

検索ユースケース

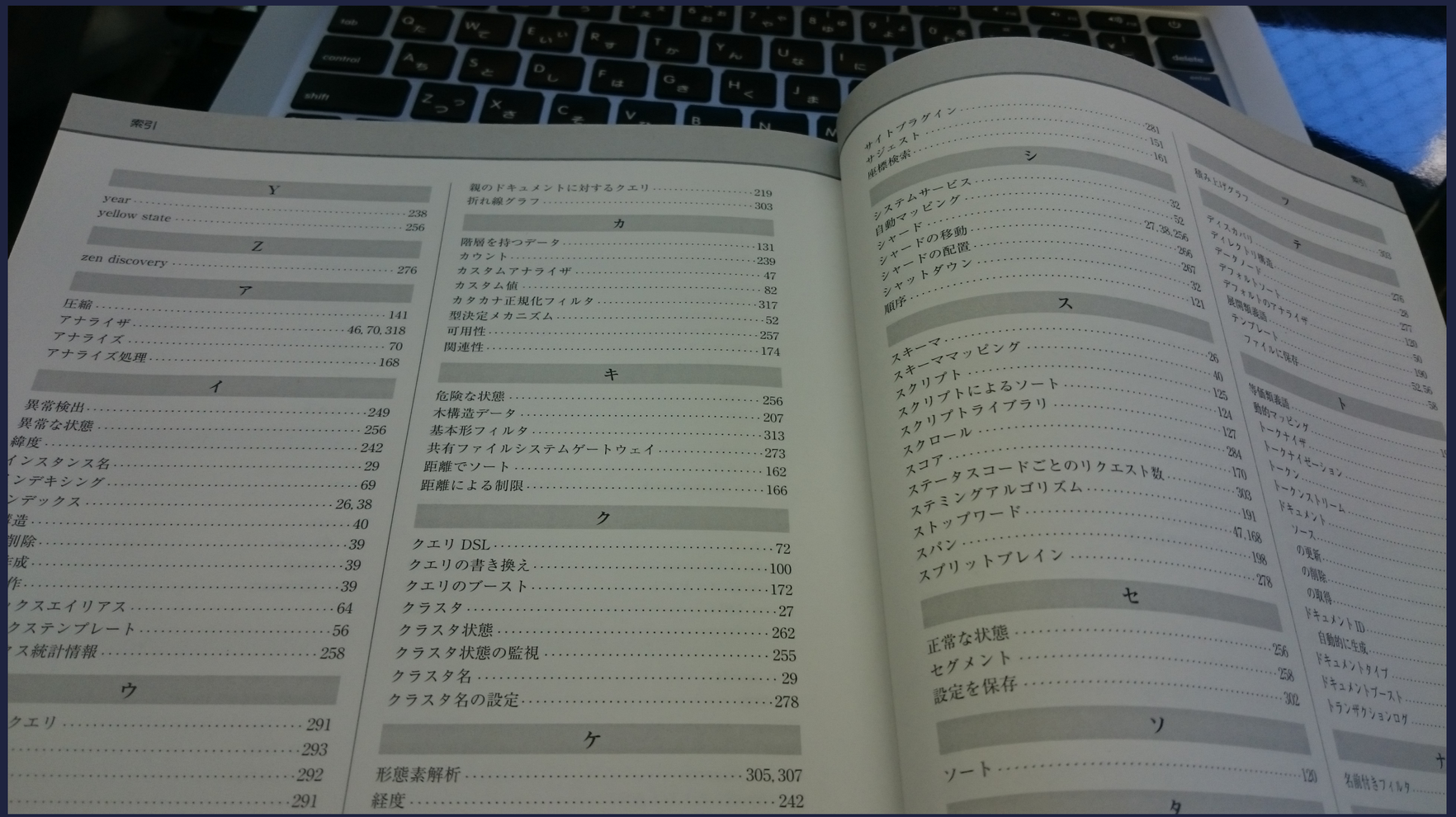
- 図書館
- 本
- *Grep*
- *Web*サービス*(Google,Yahoo!,Bing*など*な*

検索ユースケース - 図書館



検索ユースケース - 図書館

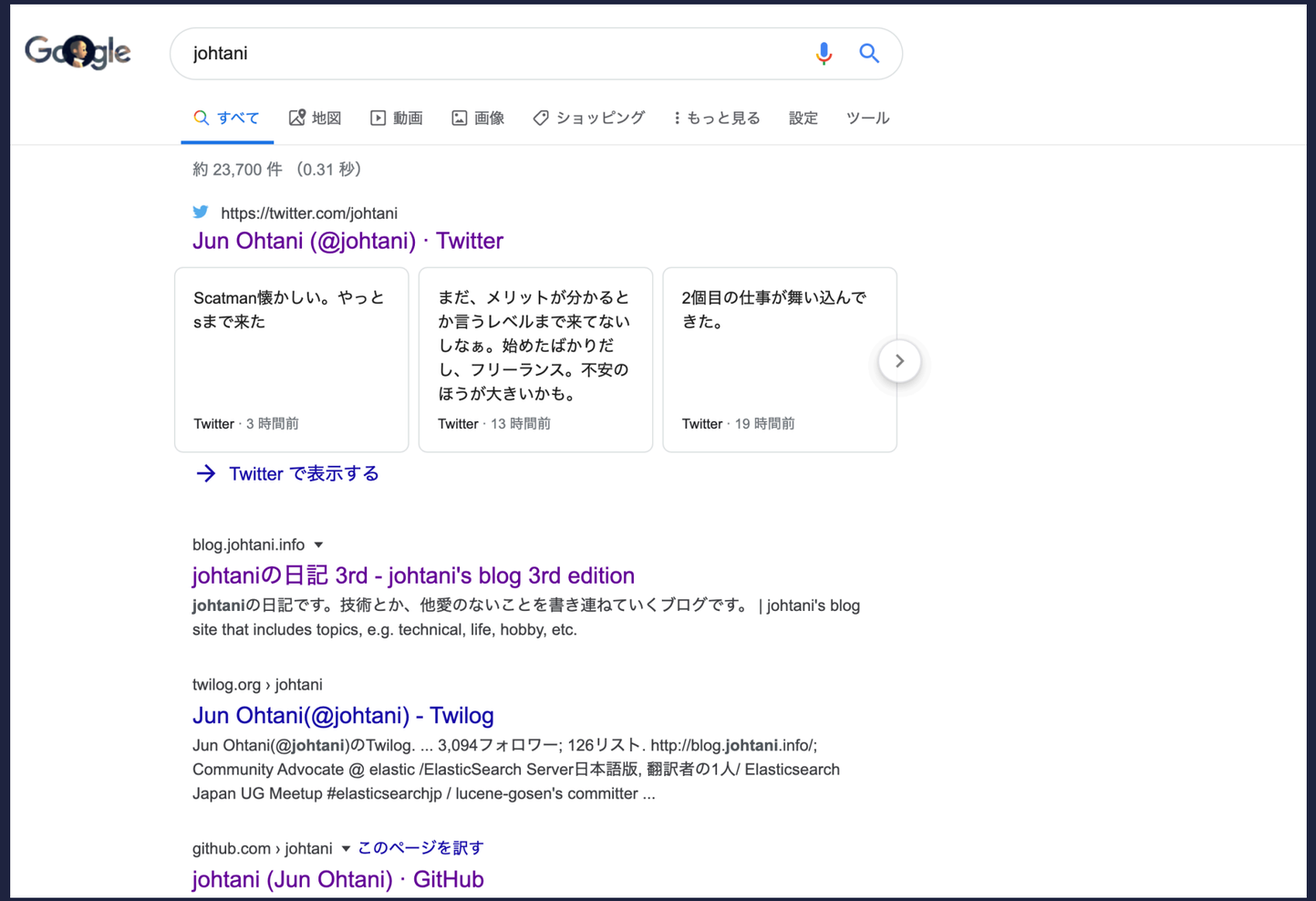
検索ユースケース - 本


索引
Y
year 238
yellow state 256
Z
zen discovery 276
ア
圧縮 141
アナライザ 46, 70, 318
アナライズ 70
アナライズ処理 168
イ
異常検出 249
異常な状態 256
緯度 242
インスタンス名 29
インデキシング 69
インデックス 26, 38
作成 40
削除 39
作成 39
作 39
クエイリアス 64
クステンプレート 56
ス統計情報 258
ウ
クエリ 291
293
292
291
親のドキュメントに対するクエリ 219
折れ線グラフ 303
カ
階層を持つデータ 131
カウント 239
カスタムアナライザ 47
カスタム値 82
カタカナ正規化フィルタ 317
型決定メカニズム 52
可用性 257
関連性 174
キ
危険な状態 256
木構造データ 207
基本形フィルタ 313
共有ファイルシステムゲートウェイ 273
距離でソート 162
距離による制限 166
ク
クエリ DSL 72
クエリの書き換え 100
クエリのブースト 172
クラスタ 27
クラスタ状態 262
クラスタ状態の監視 255
クラスタ名 29
クラスタ名の設定 278
ケ
形態素解析 305, 307
経度 242
サイトプラグイン 281
サジェスト 151
座標検索 161
シ
システムサービス 32
自動マッピング 52
シャード 27, 38, 256
シャードの移動 256
シャードの配置 267
シャットダウン 32
順序 121
ス
スキーマ 26
スキーママッピング 40
スクリプト 125
スクリプトによるソート 124
スクリプトライブラリ 127
スクロール 264
スコア 170
ステータスコードごとのリクエスト数 303
ステミングアルゴリズム 191
ストップワード 47, 168
スパン 198
スプリットブレイン 278
セ
正常な状態 256
セグメント 258
設定を保存 302
ソ
ソート 120
積み上げグラフ
ディスカバリ
ディレクトリ構造
データノード
デフォルトソート
デフォルトのアナライザ
座標検索
テンプレート
ファイルに保存
ト
動的マッピング
トークナイザ
トークナイゼーション
トークン
トークンストリーム
ドキュメント
ソース
の更新
の削除
の取得
ドキュメントID
自動的に生成
ドキュメントタイプ
ドキュメントブースト
トランザクションログ
名前付きフィルタ

検索ユースケース - Grep

```
johtani@johtanis-macbook-pro-16 ~/projects/blog/hugo/blog_generator/content/post master ± grep java */*
2013/2013-06-19-introduction-kibana3.md:$ git clone https://github.com/elasticsearch/kibana.git kibana-javascript
2013/2013-06-19-introduction-kibana3.md:$ cp -R kibana-javascript /var/www/html
2013/2013-06-19-introduction-kibana3.md:ElasticSeachサーバとKibana3のApacheのサーバが別のサーバの場合やElasticSearchサーバのポートが異なる場合はkibana-javascript/config.jsファイルの編集が必要になります。
2013/2013-06-19-introduction-kibana3.md:```javascript
2013/2013-06-19-introduction-kibana3.md:http://hogehoge/kibana-javascript/
2013/2013-07-04-schemaless-example.md:java -Dsolr.solr.home=example-schemaless/solr -jar start.jar
2013/2013-07-04-schemaless-example.md:        <str name="valueClass">java.lang.Boolean</str>
2013/2013-07-04-schemaless-example.md:        <str name="valueClass">java.util.Date</str>
2013/2013-07-04-schemaless-example.md:        <str name="valueClass">java.lang.Long</str>
2013/2013-07-04-schemaless-example.md:        <str name="valueClass">java.lang.Integer</str>
2013/2013-07-04-schemaless-example.md:        <str name="valueClass">java.lang.Number</str>
2013/2013-08-02-morphlines-loadsolr.md:なんか調べることになってたので、関係しそうなMorphlinesの[LoadSolr](https://github.com/cloudera/cdk/blob/master/cdk-morphlines/cdk-morphlines-solr-core/src/main/java/com/cloudera/cdk/morphline/solr/LoadSolrBuilder.java)コマンドを調べてみました。
2013/2013-08-02-morphlines-loadsolr.md:別途、[sanitizeUnknownSolrFilds](https://github.com/cloudera/cdk/blob/master/cdk-morphlines/cdk-morphlines-solr-core/src/main/java/com/cloudera/cdk/morphline/solr/SanitizeUnknownSolrFieldsBuilder.java)というコマンドが用意されていて、Solrのスキーマにないものはこのコマンドを使って、無視するフィールド名に変えたり、雑多なデータを入れるためのフィールド名にするといった処理ができるようです。このコマンド内部で、Solrのスキーマ設定を元に、Solrのフィールドに合致する物があるかをチェックして処理しています。
2013/2013-08-02-morphlines-loadsolr.md:Solrへの登録処理自体はLoadSolrクラス内部でDocumentLoaderというクラスのload()メソッドを呼び出しているだけでした。ということで、[DocumentBuilder](https://github.com/cloudera/cdk/blob/master/cdk-morphlines/cdk-morphlines-solr-core/src/main/java/com/cloudera/cdk/morphline/solr/DocumentLoader.java)クラスを少し調査。
2013/2013-08-02-morphlines-loadsolr.md:2. なければ、[SolrServerDocumentLoader](https://github.com/cloudera/cdk/blob/b6f98cff4a027af04f97fdec9abf729785d74cf5/cdk-morphlines/cdk-morphlines-solr-core/src/main/java/com/cloudera/cdk/morphline/solr/SolrServerDocumentLoader.java)をnewしたものを利用
2013/2013-08-02-morphlines-loadsolr.md:Solrへの接続情報とか設定ファイルとかSolrCloud用のZooKeeperとかは[SolrLocatorクラス](https://github.com/cloudera/cdk/blob/master/cdk-morphlines/cdk-morphlines-solr-core/src/main/java/com/cloudera/cdk/morphline/solr/SolrLocator.java)に設定される内容が利用されます。
2013/2013-08-02-morphlines-loadsolr.md:1.のパターンは、どうやら、[Cloudera SearchのMapReduceIndexerToolのクラス](https://github.com/cloudera/search/blob/master/search-mr/src/main/java/org/apache/solr/hadoop/morphline/MorphlineMapper.java)にあるMyDocumentLoaderかなぁと。
2013/2013-08-02-morphlines-loadsolr.md:内部処理は、HadoopのContext.writeメソッドにでSolrInputDocument（＝MorphlinesのRecord）を書きだして、ReducerでSolrOutputFormatでインデックス作成の流れかなと。たぶん、[MorphlineMapRunner](https://github.com/cloudera/search/blob/master/search-mr/src/main/java/org/apache/solr/hadoop/morphline/MorphlineMapRunner.java)あたり
```

検索ユースケース - Webサービス



検索の種類

- ディレクトリ型検索
- キーワード検索
- 音声検索
- 画像検索

*Grep*と全文検索の違い

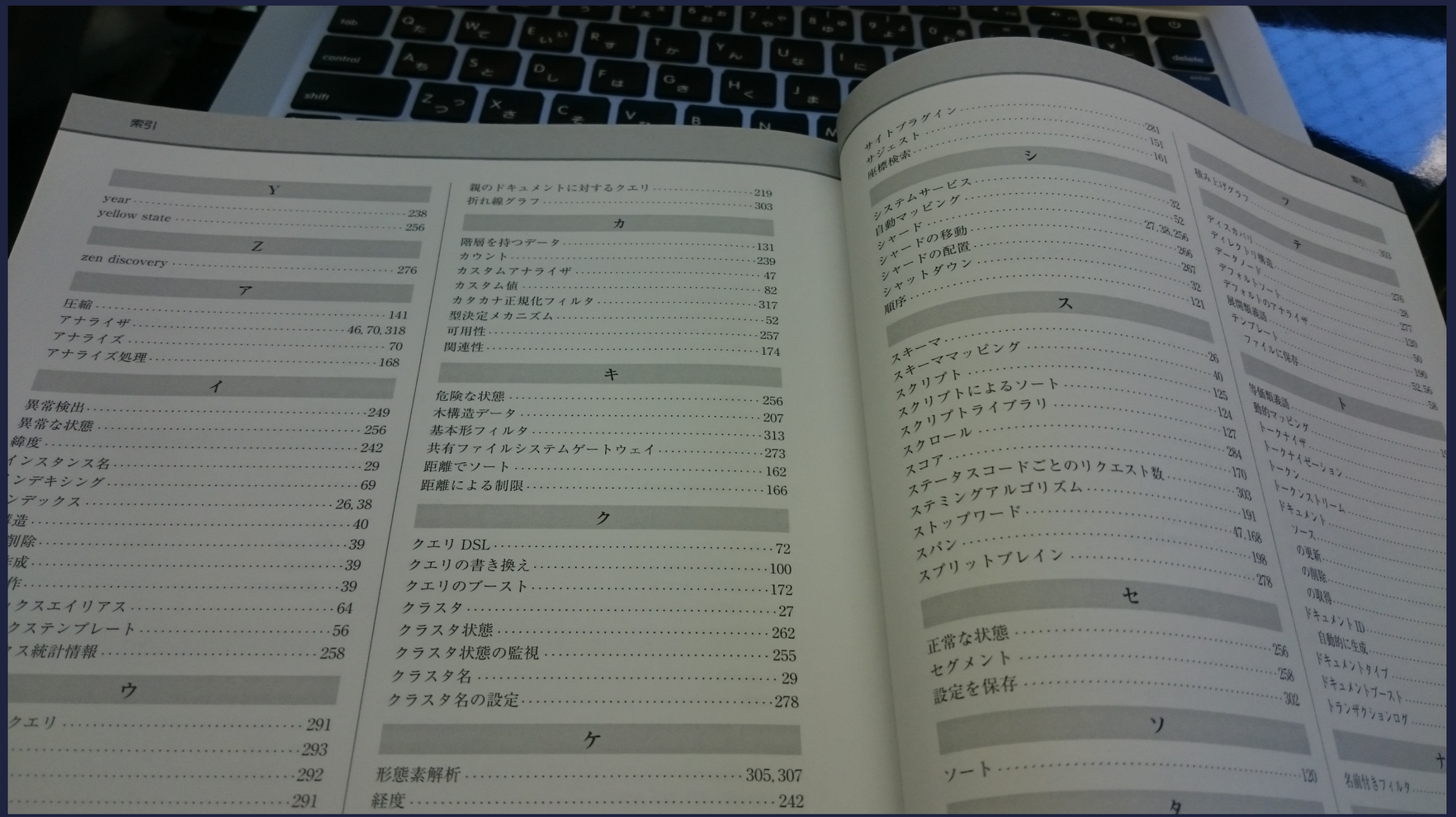
- *Grep* - 部分一致(文字一致)
- 全文検索 - 転置インデックスを元に検索(詳しくは後述)

Grepと全文検索の違い - Grep

```
.md:$ git clone https://github.com/elasticsearch/kibana.g
.md:$ cp -R kibana-javascript /var/www/html
.md:ElasticSeachサーバとKibana3のApacheのサーバが別のサー

.md:```javascript
.md:http://hogehoge/kibana-javascript/
d:java -Dsolr.solr.home=example-schemaless/solr -jar star
d:         <str name="valueClass">java.lang.Boolean</str>
d:         <str name="valueClass">java.util.Date</str>
d:         <str name="valueClass">java.lang.Long</str>
d:         <str name="valueClass">java.lang.Integer</str>
d:         <str name="valueClass">java.lang.Number</str>
md:なんか調べることになってたので、関係しそうなMorphlines
/cdk/morphline/solr/LoadSolrBuilder.java)コマンドを調べて
```

Grepと全文検索の違い - 全文検索



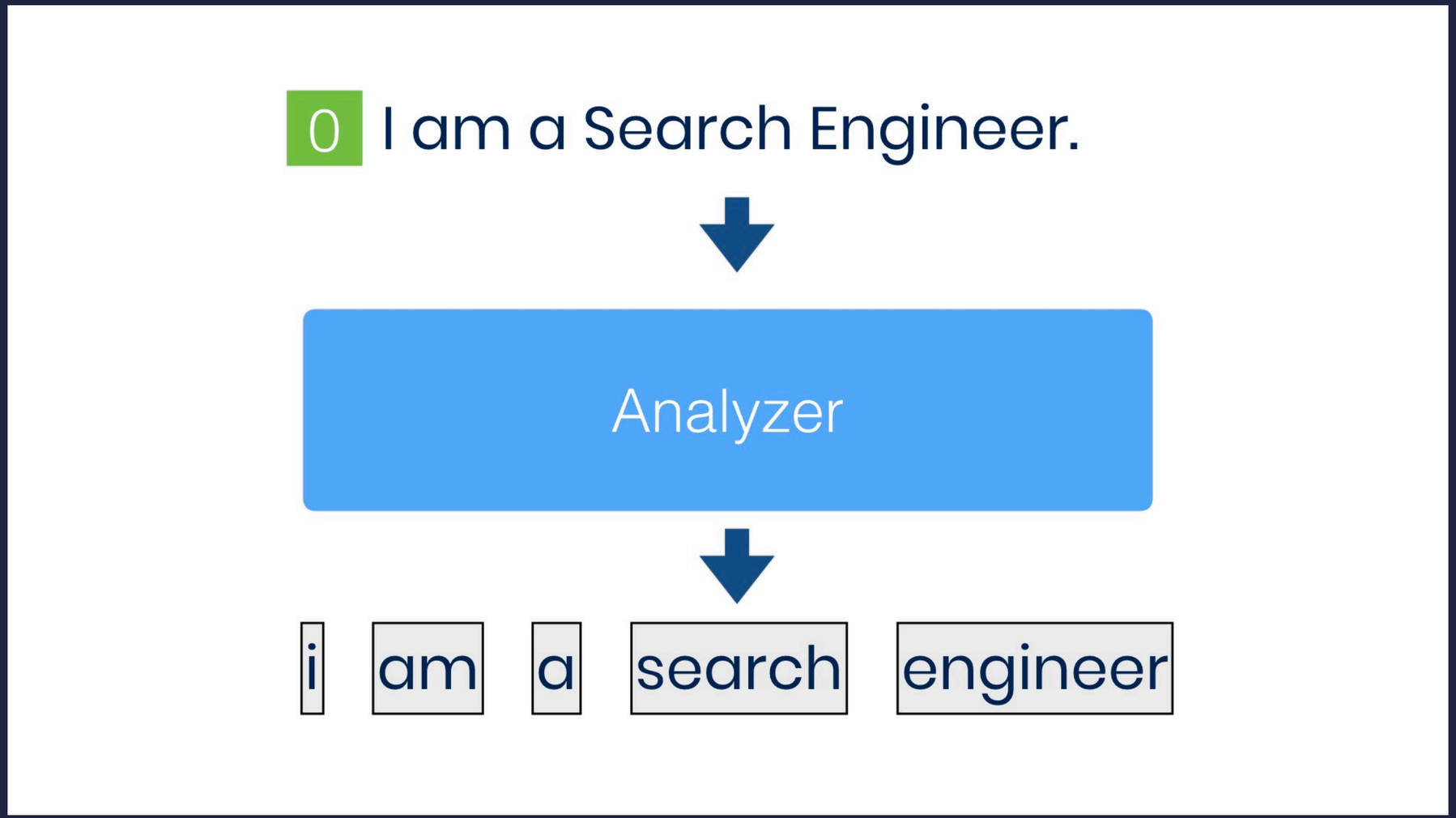
Lucene/Elasticsearchの仕組み - Index

1. ドキュメントの文字列を単語に分割
   - *Text Analysis*
2. 分割された単語で転置インデックス構築
   - *Indexer / Inverted Index*

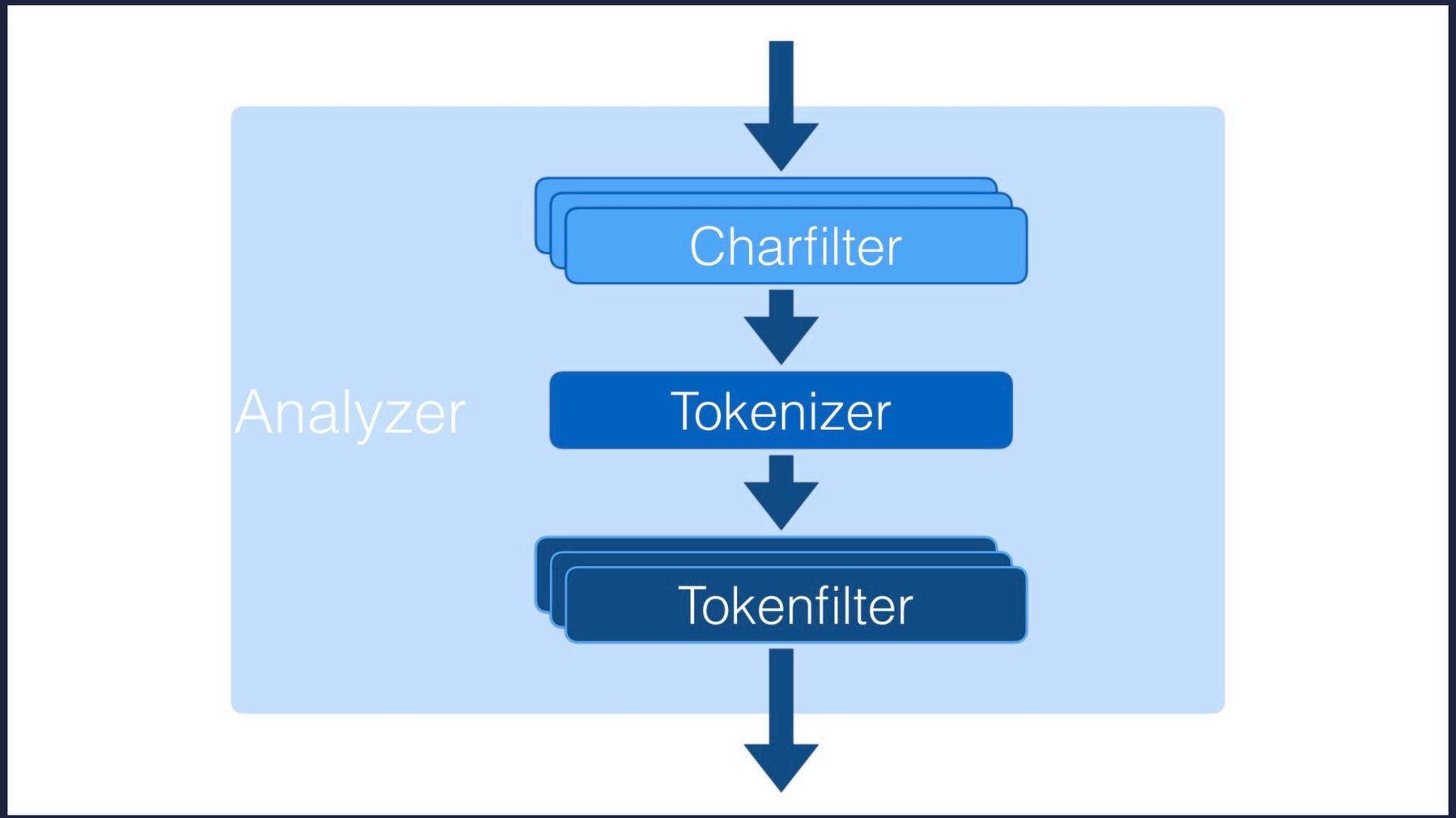
Text Analysis - Elasticsearch

- 入力文字列を一定のルールに基づいて、単語列にして出力する
- *Analyzer*と呼ばれる仕組みにより実現

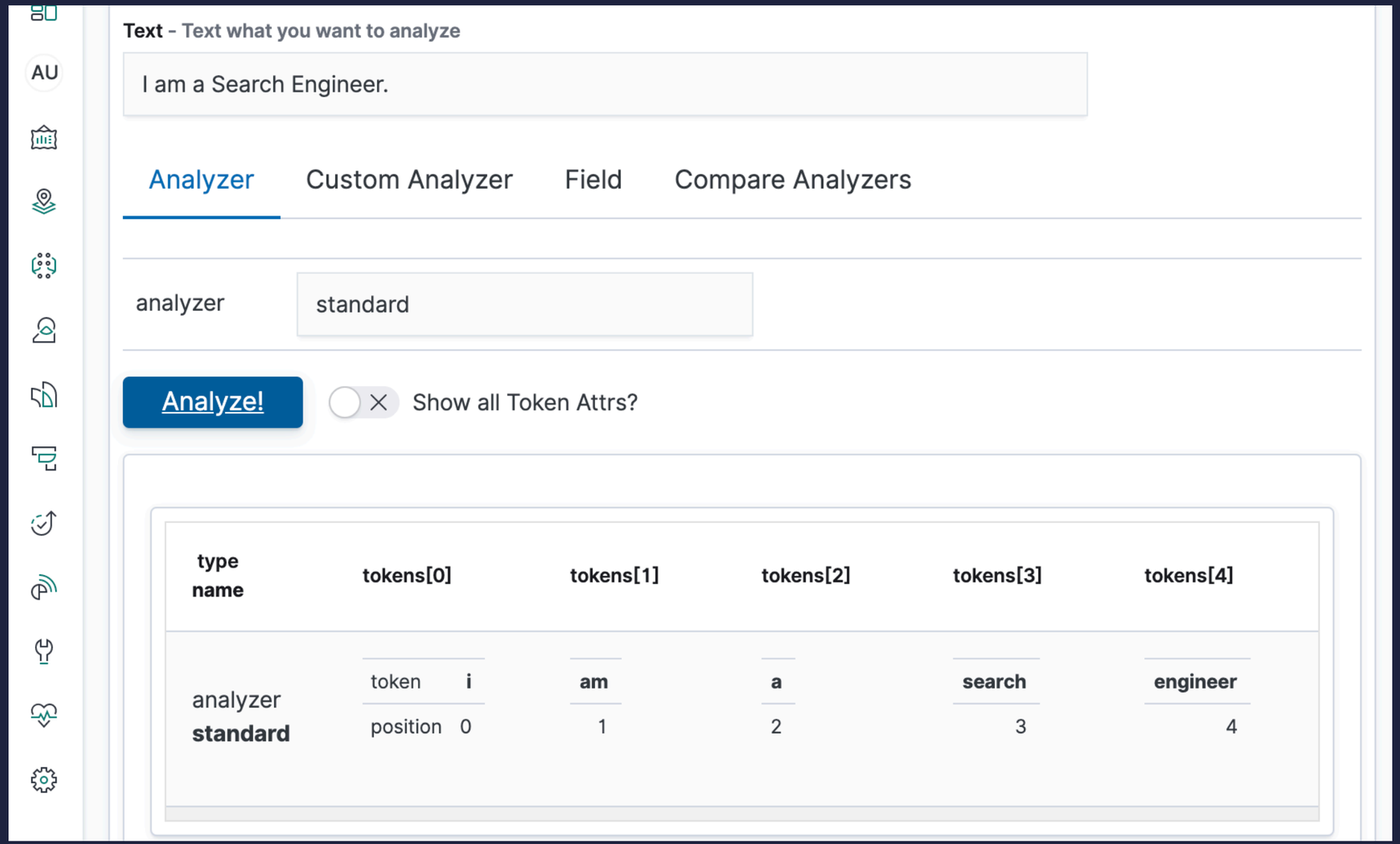
Text Analysis - English



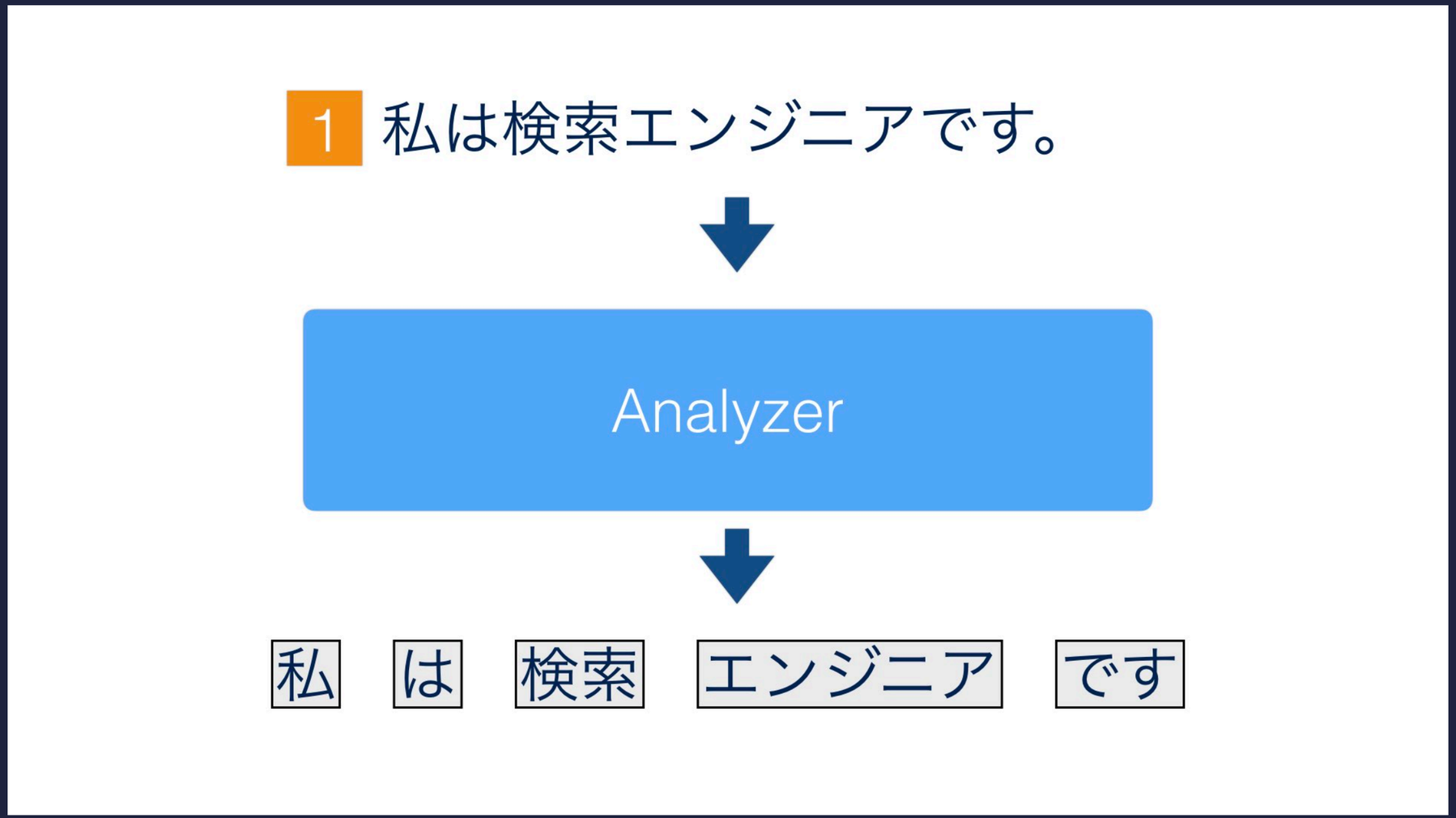
Analyzer Architecture - Elasticsearch



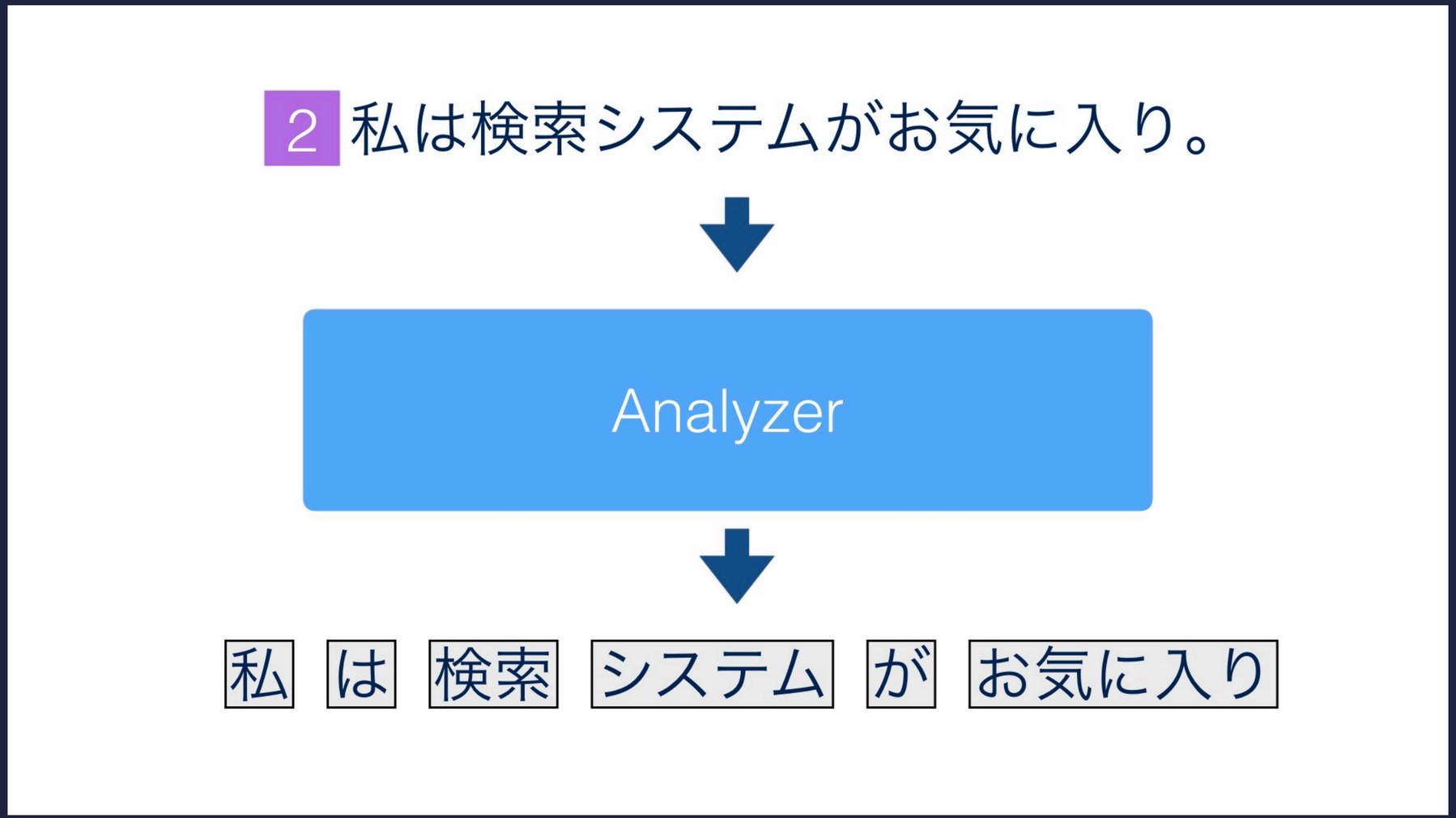
Demo - Kibaan Analyze API UI plugin



Text Analysis - Japanese



Text Analysis - Japanese

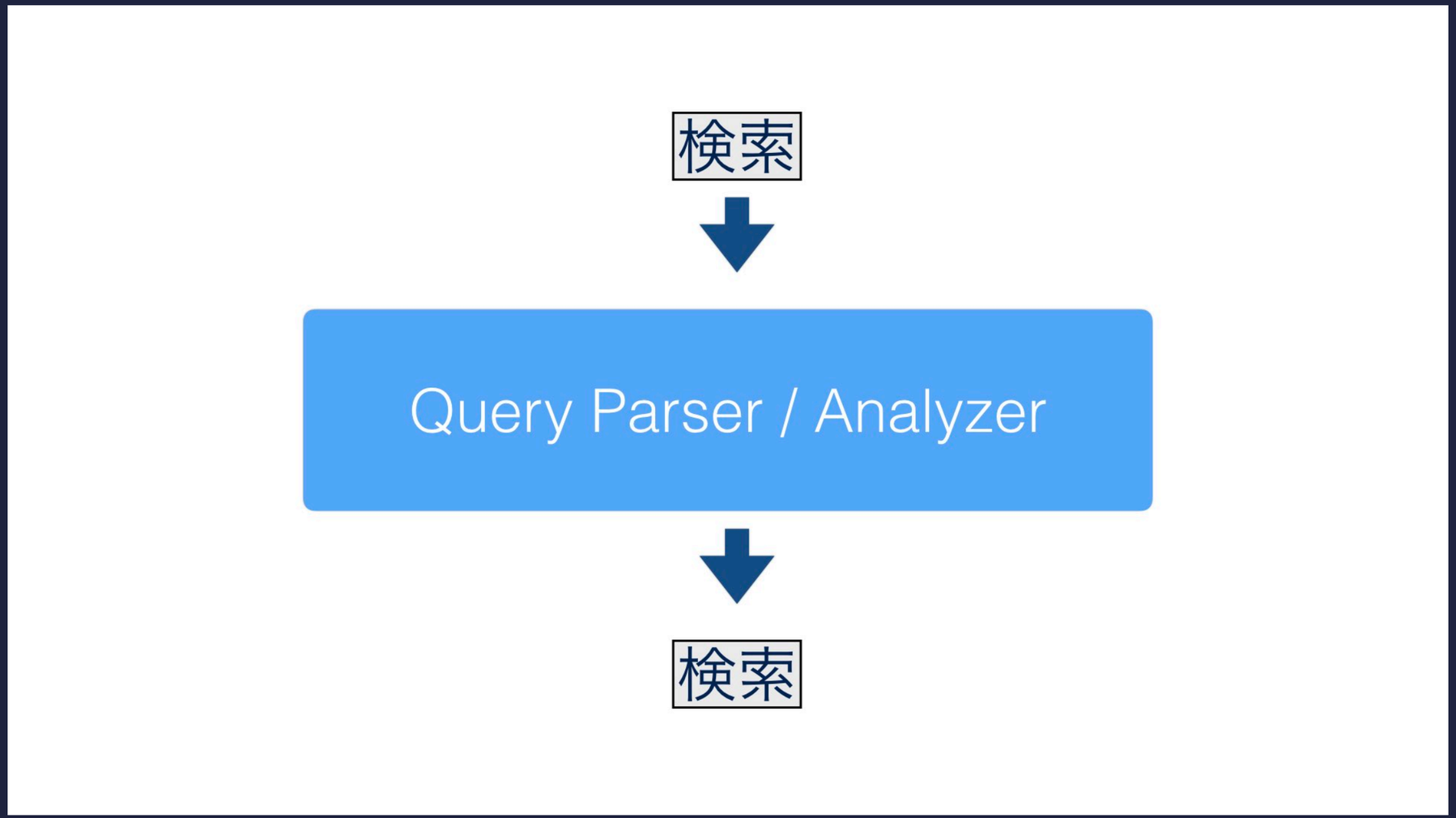


Inverted Index - Elasticsearch

Lucene/Elasticsearchの仕組み - Query

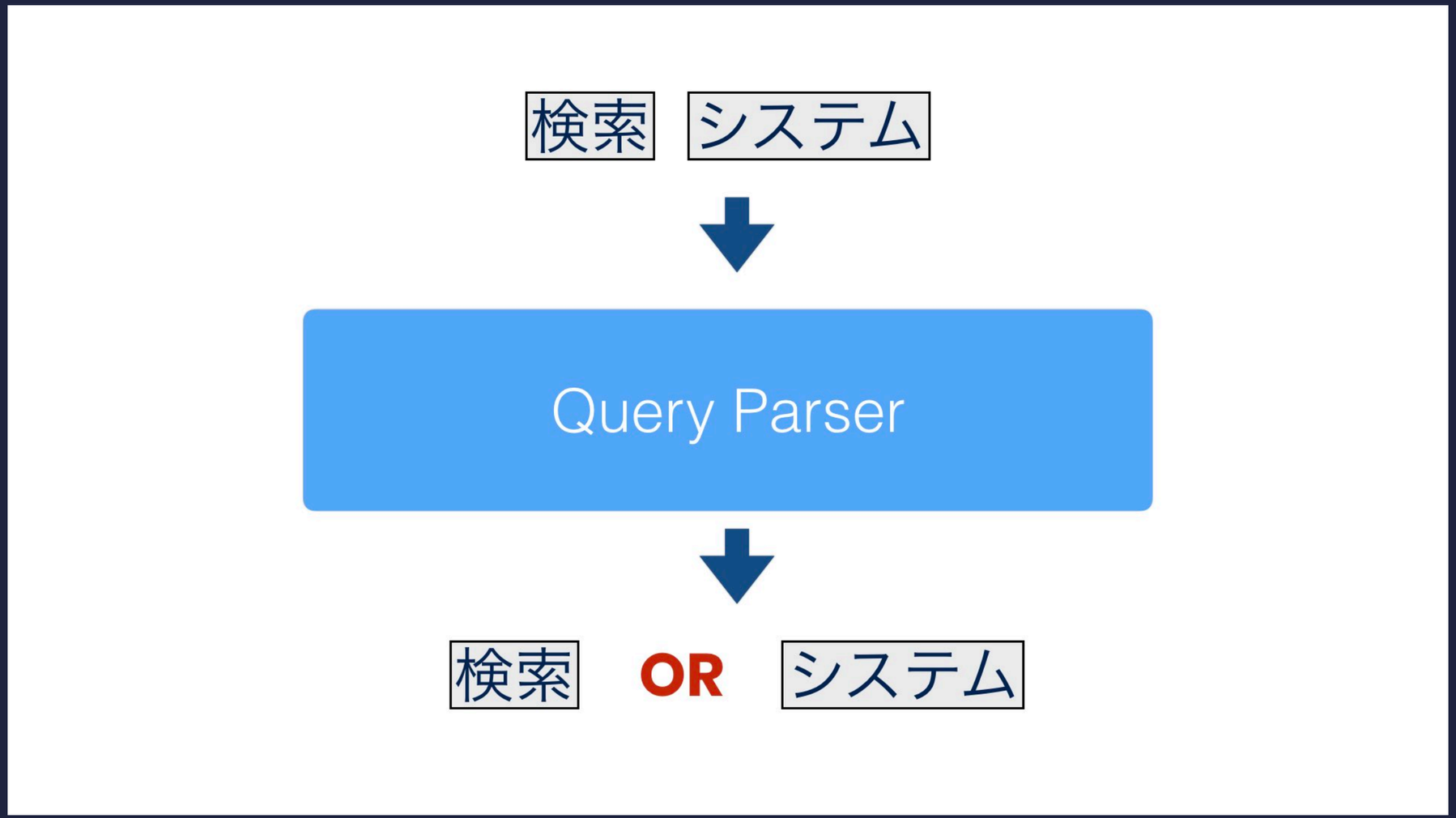
1. クエリの文字列を単語に分割
   - Query Parser / Analyzer
2. 分割された単語で転置インデックス検索
   - Searcher / Inverted Index

Query Parser / Analyzer - Elasticsearch



Inverted Index - Elasticsearch

Query Parser / Analyzer - Elasticsearch



Query Parser / Analyzer - Elasticsearch

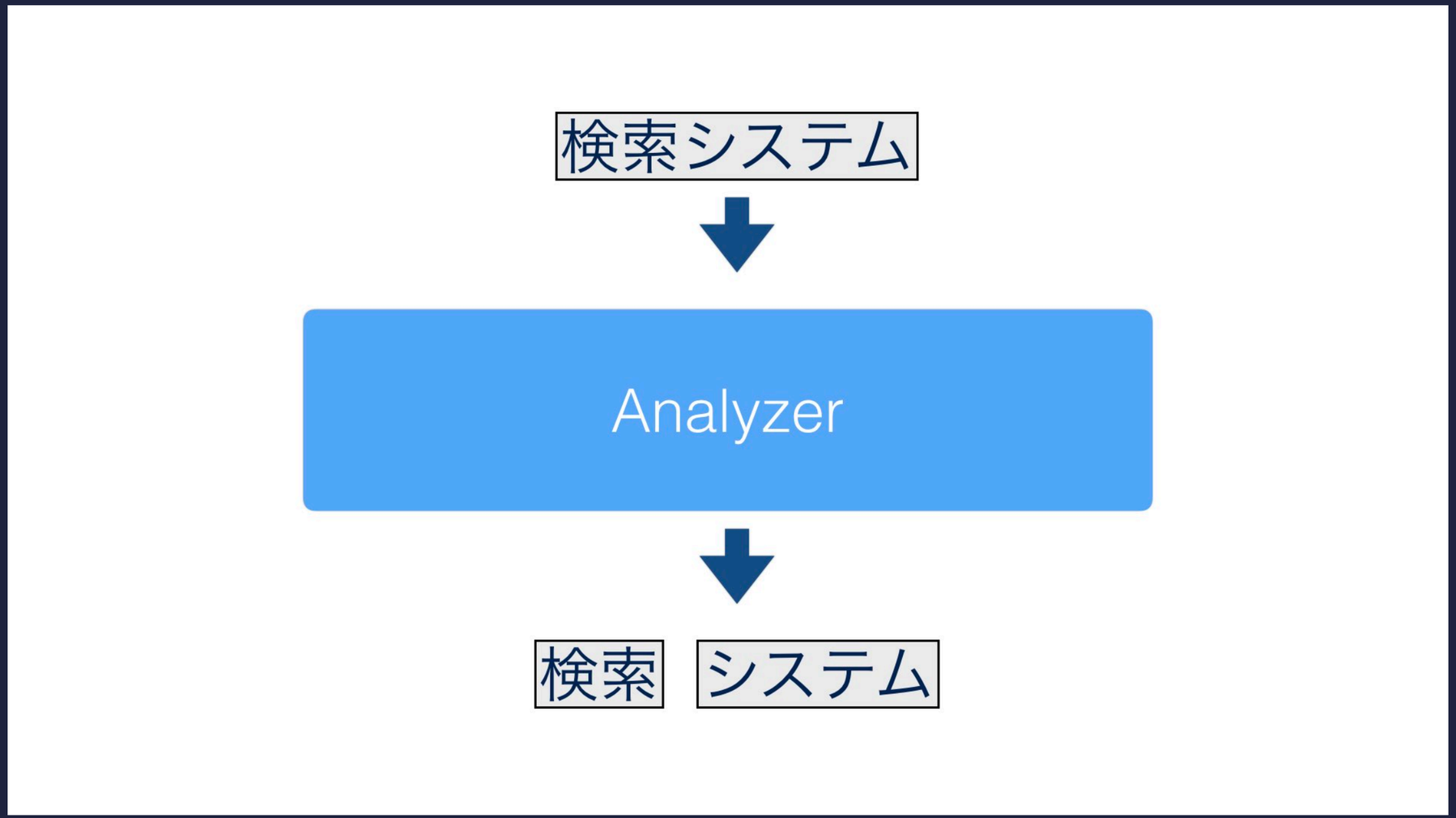
- デフォルトはOR検索
- ANDにも変更可能

Inverted Index - Elasticsearch

フレーズ(熟語)は？

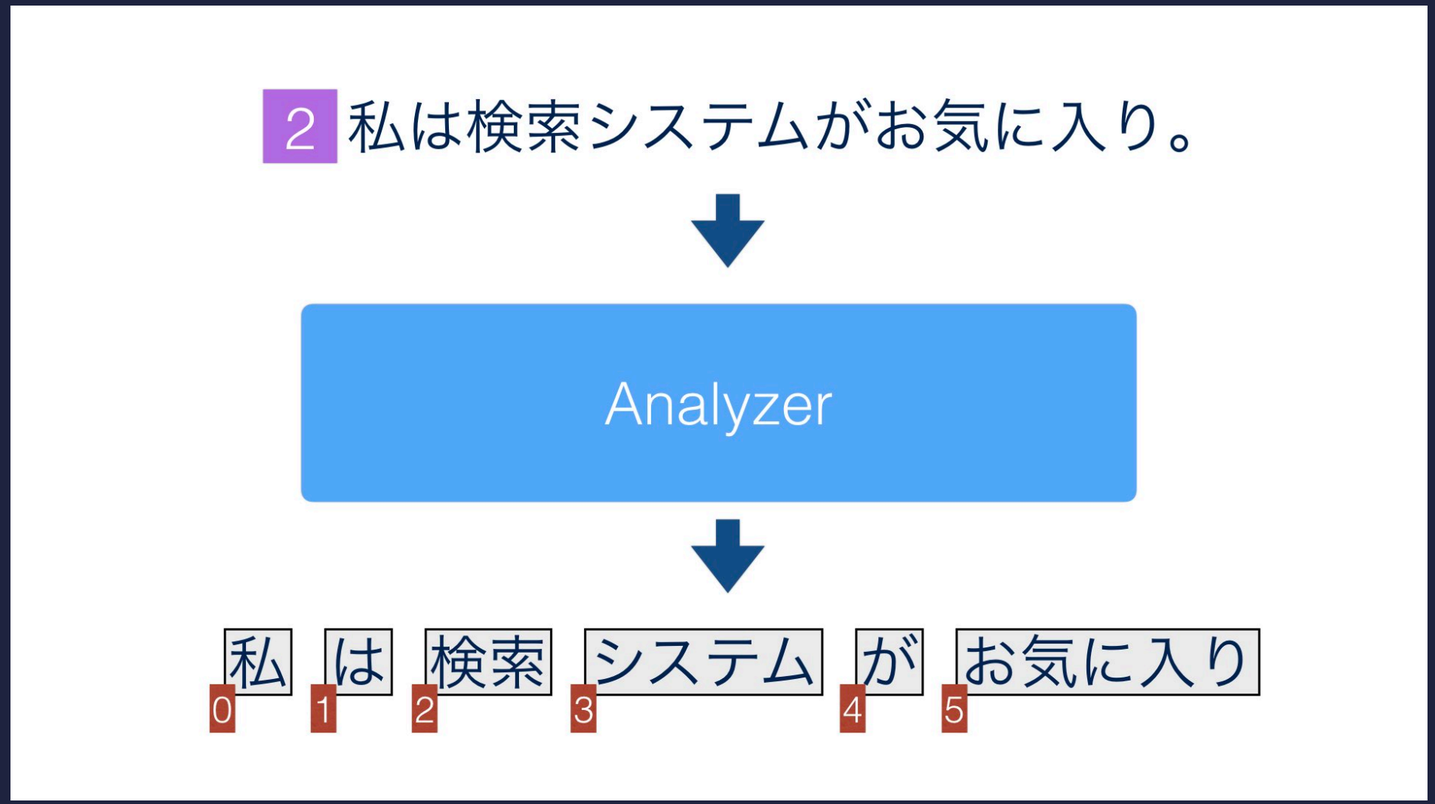
- 「検索」「システム」と別れて出てくる
- 普通に検索すると、個別の単語として検索

Phrase Query - Elasticsearch



フレーズ検索 - *Elasticsearch*

- 単語の出現位置も考慮して結果を返す



正規表現？(前方一致とか)

^私は*

できると思います?

正規表現？

Regexp Query - Elasticsearch

- Elasticsearchにも[Regexp Query]というものがある
- これは、単語の一覧に対して正規表現で検索

検索結果の並び順

- 検索エンジンはスコアでソート可能
- もちろんスコア以外の項目でソートも可能

スコア？

- *BM25*
- *TF-IDF*

その他の検索の機能

- ファセット - 絞り込みのヒントとして
- *Geo*検索 - 位置検索
- ハイライト - どこがマッチした?
- サジェスト - オートコンプリーション

よりよい検索にするためには？

検索エンジンだけでは対応できないことがいっぱい

- *UI/UX*
- 検索ログ
- 検索結果の評価

検索システムでの選択肢 - ミドルウェア

- *RDB*
- *Apache Solr / Elasticsearch*
- *Groongaファミリー*
- *etc.*

検索システムでの選択肢 - *SaaS*

- *Algolia*
- *Elastic Cloud App Search*
- *Azure Cognitive Search*
- *Amazon CloudSearch / etc.*

まとめ

- *Grep*と検索の仕組みの違い
- 仕組みをしることで何ができて何ができないかを理解
  - また、なにが問題でうまく動いてない

Thanks!

@johtani

おすすめ本

https://booth.pm/ja/items/1865120

参考文献

- *Elasticsearch: The Definitive Guide*
- *Luceneのインデックスファイル構造*
- *Analyze API UI plugin*

[WIP] ほかにも書いていきたいこと

Elasticsearchの検索のフェーズの話

- Queryフェーズと Fetchフェーズ
- dfs
- Aggsとの関係

転置インデックス詳細

- *term dictionary*
- *postings list*

スコアの詳細?

- 式とか?
- *Script Score*とか