



# kearch

<https://kearch.info>

分野限定型検索エンジンを  
複数組み合わせた分散型検索エンジン

河田旺 稲垣悠一



河田 旺  
@a\_kawashiro

全体設計  
自然言語処理



稲垣 悠一  
@gky360

kubernetes関係  
フロントエンド

Google

Yandex

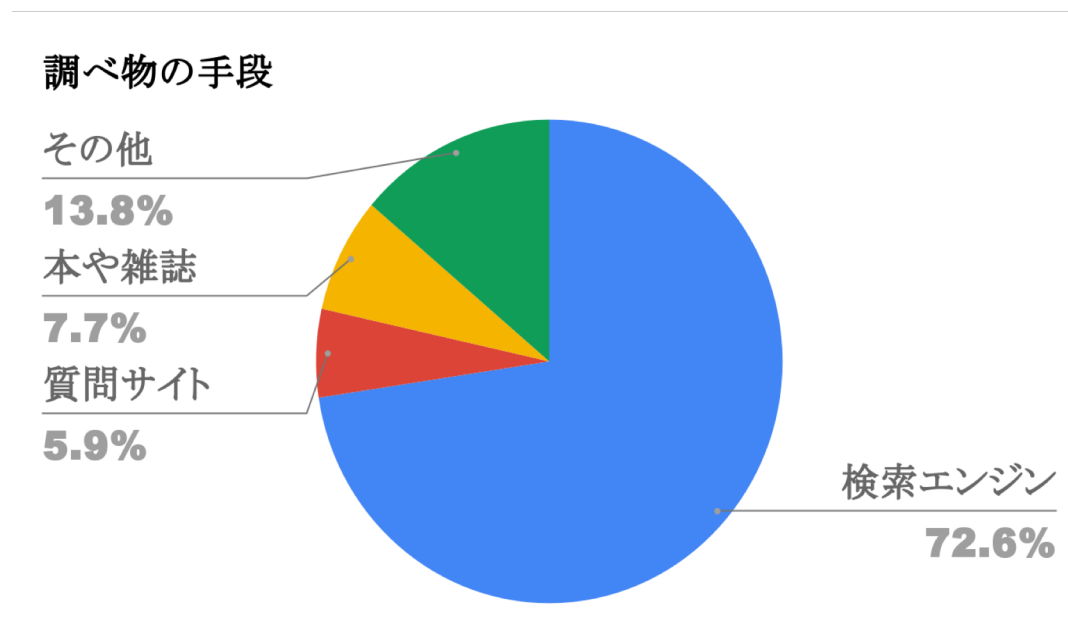
 Bing

Baidu 百度

# 検索エンジンの現状

## やっぱり検索エンジンは重要

日常生活における調べ物の約70%が検索エンジン経由



※総務省「社会課題解決のための新たなICTサービス・技術への人々の意識に関する調査研究」（平成27年）



## 世の中の検索エンジンへの不満

- **一部の企業しか公開できない**
  - Google, Baidu, Microsoftをあわせた世界シェアは94%※2
  - 莫大なリソースが必要
  
- **検索エンジンの不透明性**
  - 民間企業による非公開のアルゴリズム
  - 国家権力による検閲の可能性※1

※1 China: World Leader of Internet Censorship, <https://www.hrw.org/news/2011/06/03/china-world-leader-internet-censorship>, (2019/02/11)

※2 Search Engine Market Share, <https://www.netmarketshare.com/search-engine-market-share>, (2019/02/11)

## プロジェクトの目的

# 自由な検索エンジンを作る

- 省リソースで誰でもデプロイ可能
- ランキングアルゴリズムが公開
- 検閲が困難

イントロ

検索エンジンプラットフォーム kearch

デプロイの流れ

デモ

技術詳細



# kearch

<https://github.com/kearch/kearch>

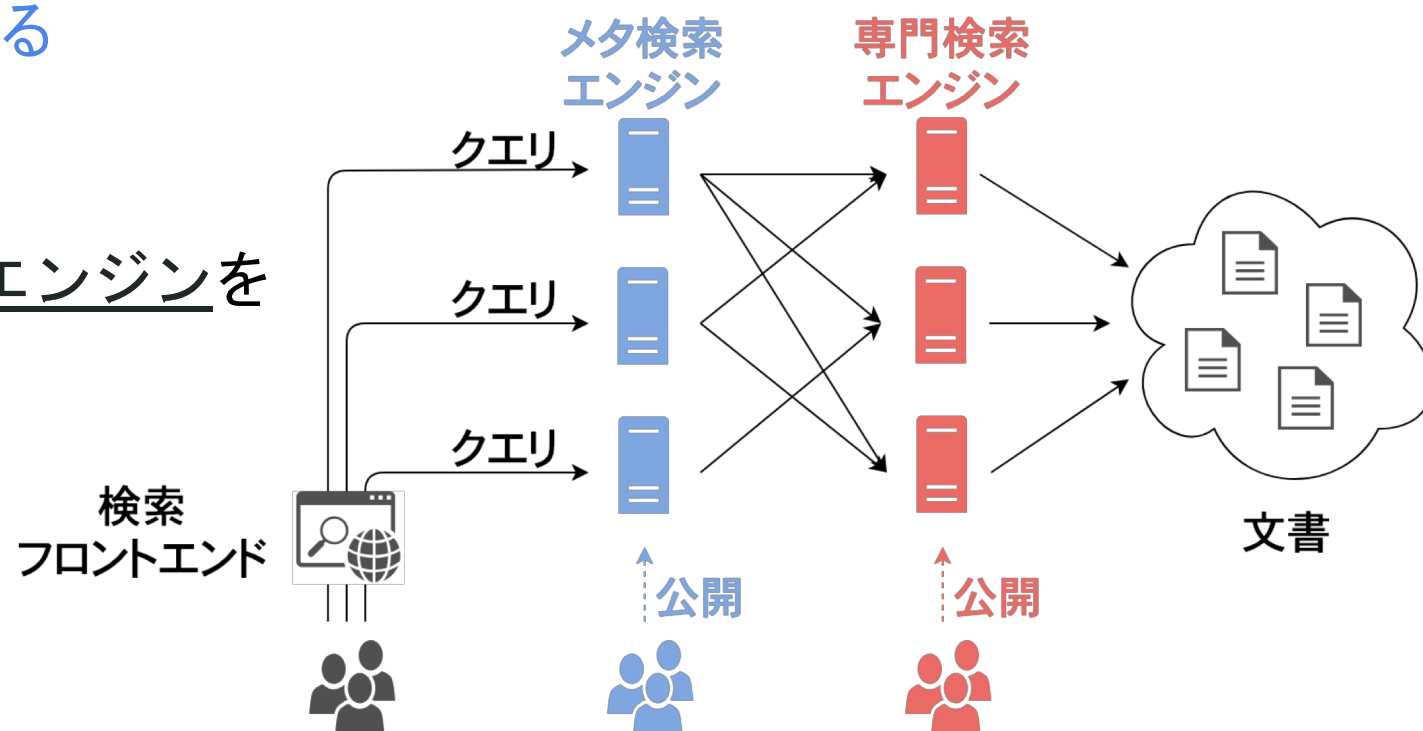
<https://kearch.info>



# kearch: 自由な検索エンジンプラットフォーム

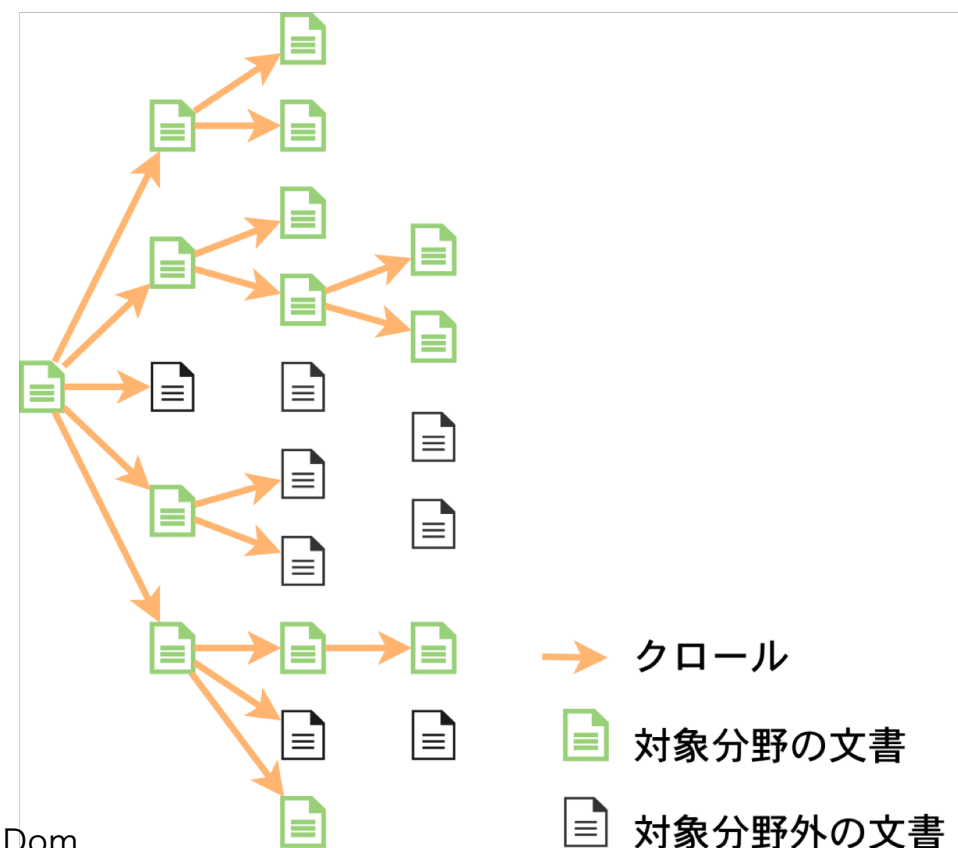
- a. 分野を限定した検索エンジンを
- b. 複数つなぎ合わせる

ことで、  
ボトムアップに検索エンジンを  
実現する



## 専門検索エンジン

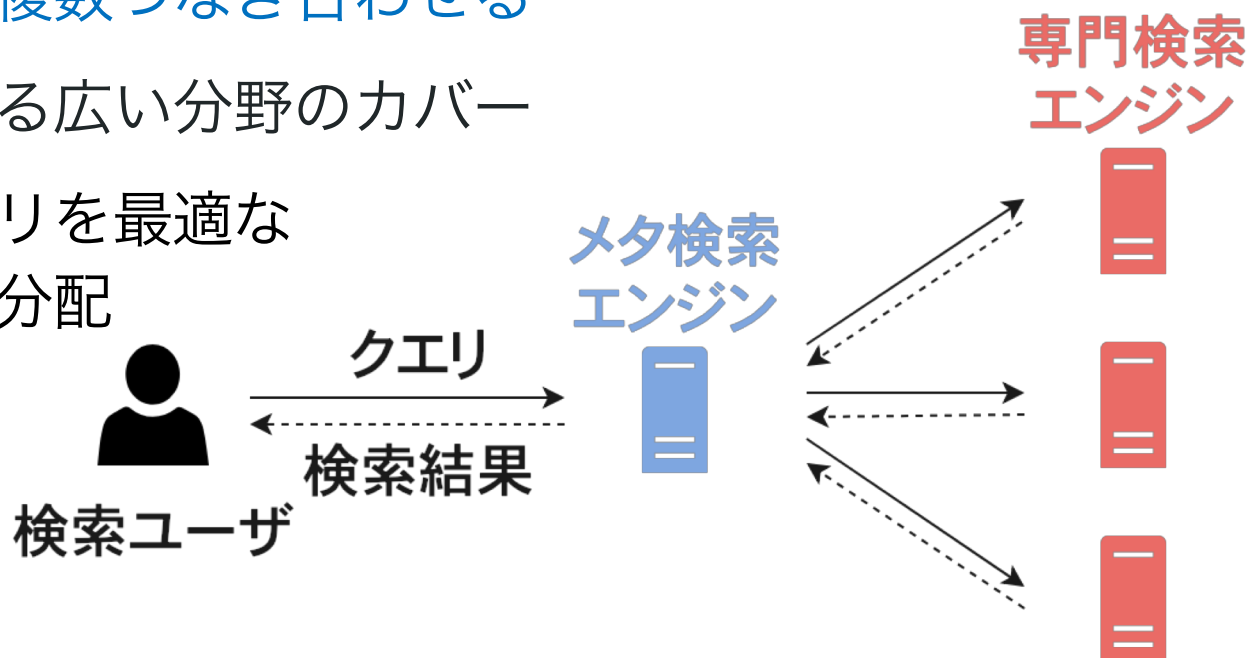
- Focused crawling ※
  - 分野を限定してクローリング
  - 分野を限定することによるリソースの節約、精度の向上
  - 分野の定義方法
    - URLリスト
    - 単語



※ Chakrabarti, Soumen, Martin Van den Berg, and Byron Dom.  
"Focused crawling: a new approach to topic-specific Web resource discovery."  
*Computer networks* 31.11-16 (1999): 1623-1640.

## メタ検索エンジン

- Federated search ※
  - 専門検索エンジンを複数つなぎ合わせる
  - 複数分野の接続による広い分野のカバー
  - ユーザーからのクエリを最適な専門検索エンジンに分配



※ "Distributed information retrieval." *Advances in information retrieval*. Springer, 2002. 127-150.

## kearchの特長

- コマンド一つでデプロイ可能
  - ansible-playbookだけでデプロイできる
- 検索エンジンを構成する各サーバは省リソース
  - CPU: 4core/メモリ: 8GB/ストレージ: 100GB
- オープンソースでアルゴリズムが公開
  - GitHub上で公開



イントロ

検索エンジンプラットフォーム kearch

デプロイの流れ

デモ

技術詳細

## デプロイの流れ

1. 専門検索エンジンのデプロイ
2. 専門分野設定
3. クロール開始
4. メタ検索エンジンのデプロイ
5. メタ  $\longleftrightarrow$  専門の接続
6. 検索

# 1. 専門検索エンジンのデプロイ

```
$ git clone https://github.com/kearch/kearch.git  
$ cd kearch  
$ ansible-playbook sp-playbook.yml -i <HOSTNAME>,  
-u <USERNAME> --ask-become-pass -vvv
```

kubernetesクラスタの構築  
コンテナビルド  
クラスタへのデプロイ

## 2. 専門分野設定

専門検索エンジン  
管理画面

Learn Parameter for the classifier in the crawler using  
word frequency

language

en

Word frequencies in your crawling topic

haskell 210  
language 55  
programming 43

use default dict

Word frequencies in random topic

LEARN CRAWLER PARAMETERS

## 3. クロール開始

専門検索エンジン  
管理画面

URLs to crawl separated by newlines

URLs

[https://en.wikipedia.org/wiki/Speech\\_recognition](https://en.wikipedia.org/wiki/Speech_recognition)  
[https://en.wikipedia.org/wiki/Distributed\\_artificial\\_intelligence](https://en.wikipedia.org/wiki/Distributed_artificial_intelligence)  
<https://en.wikipedia.org/wiki/Anatomy>

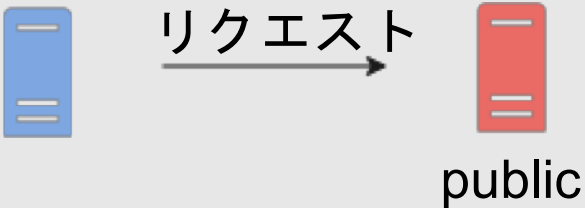
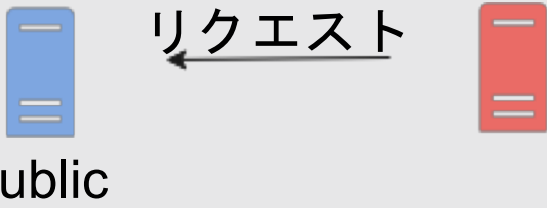
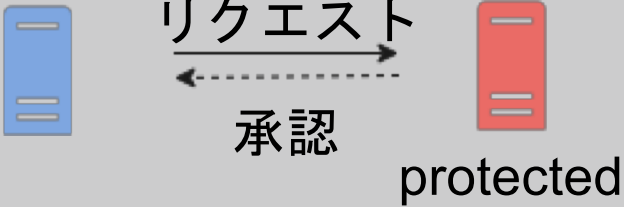
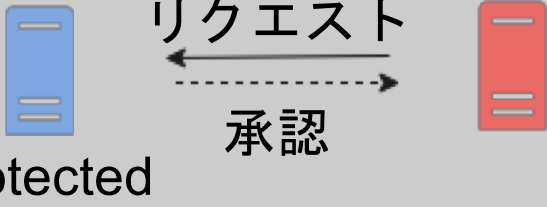
INIT CRAWL URLS

## 4. メタ検索エンジンのデプロイ

```
$ git clone https://github.com/kearch/kearch.git  
$ cd kearch  
$ ansible-playbook me-playbook.yml -i <HOSTNAME>,  
-u <USERNAME> --ask-become-pass -vvv
```

kubernetesクラスタの構築  
コンテナビルド  
クラスタへのデプロイ

# メタ ↔ 専門の接続モード

	メタ → 専門 に接続要求	メタ ← 専門 に接続要求
相手が public		
相手が protected		

## 5. メタ↔専門の接続

Requests from specialist servers to this meta server

sp_host	is_approved		
163.43.31.234	true	APPROVE	DELETE
153.125.225.80	true	APPROVE	DELETE
27.133.130.101	true	APPROVE	DELETE
163.43.28.11	false	APPROVE	DELETE



## 6. 検索

メタ検索エンジン  
検索画面



kearch

stack tutorial

RETRIEVE

Specialist server name

select automatically

### User guide - The Haskell Tool Stack

<http://docs.haskellstack.org/en/stable/GUIDE/>

The Haskell Tool Stack Home Changelog Tool documentation Install/upgrade User guide User guide Stack's functions Downloading and Installation Hello World Example Inner Workings

2.170303 from Haskell (163.43.31.234)

検索結果を返した  
専門検索エンジン

### stack/GUIDE.md at master · commercialhaskell/stack · GitHub

<https://github.com/commercialhaskell/stack/blob/master/doc/GUIDE.md>

Skip to content Why GitHub? Features → Code review Project management Integrations Actions Team management Social coding Documentatio

2.1658301 from Haskell (163.43.31.234)

# おまけ: 既存検索エンジンとの接続

README.md

## kearch-sp-google

Use Google Custome Search Engine as a specialist search engine

### Running the server

First you need to rewrite ./GoogleCustomSearchAPIKey and ./GoogleCustomSearchEngineID with your own API key and engine ID. You can get APIkey from <https://developers.google.com/custom-search/v1/introduction>. And you must rewrite ./Hostname with your hostname or IP adress of your server.

And then

```
go run main.go
```

To run the server in a docker container

イントロ

検索エンジンプラットフォーム kearch

デプロイの流れ

デモ

技術詳細

イントロ

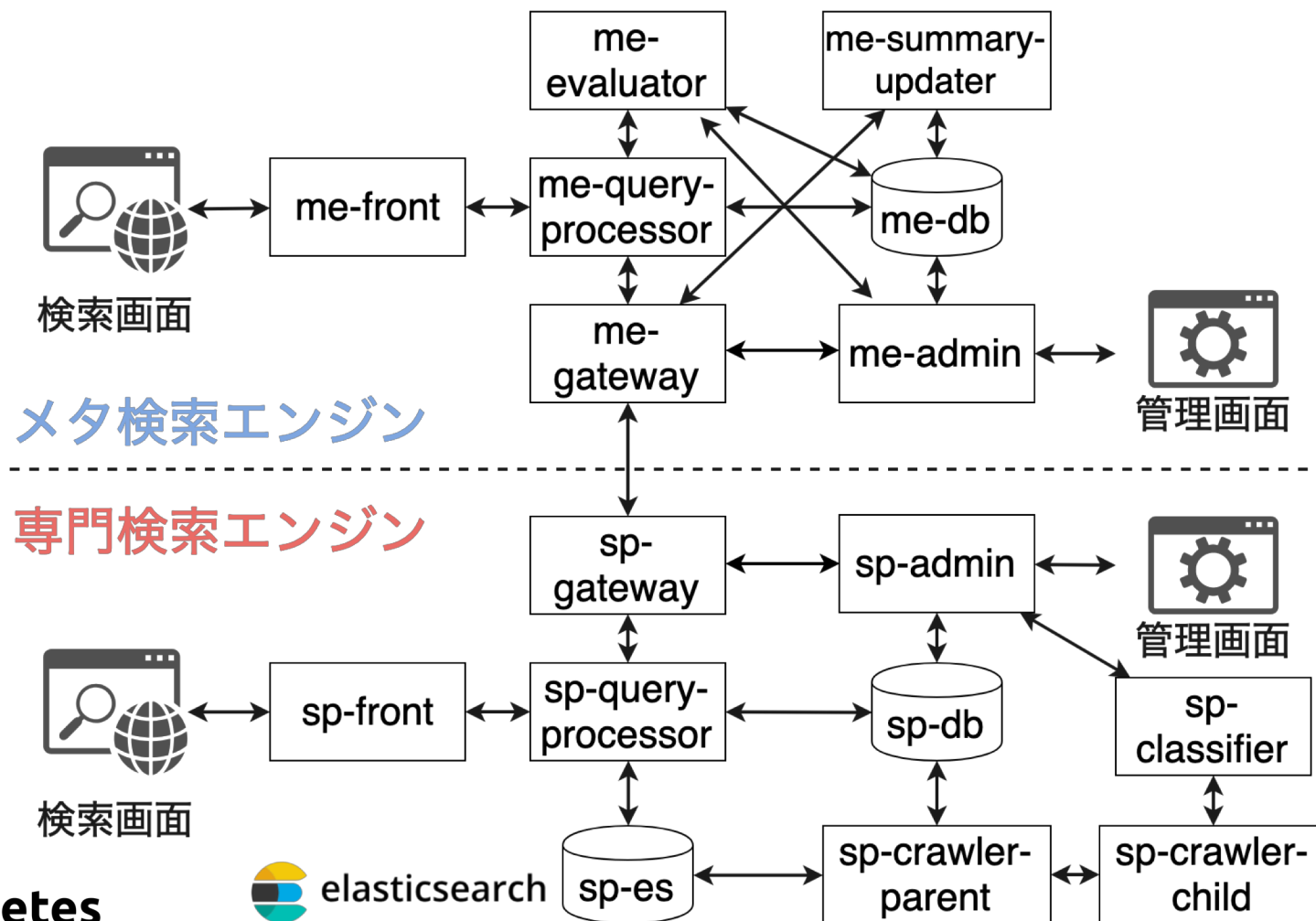
検索エンジンプラットフォーム kearch

デプロイの流れ

デモ

技術詳細

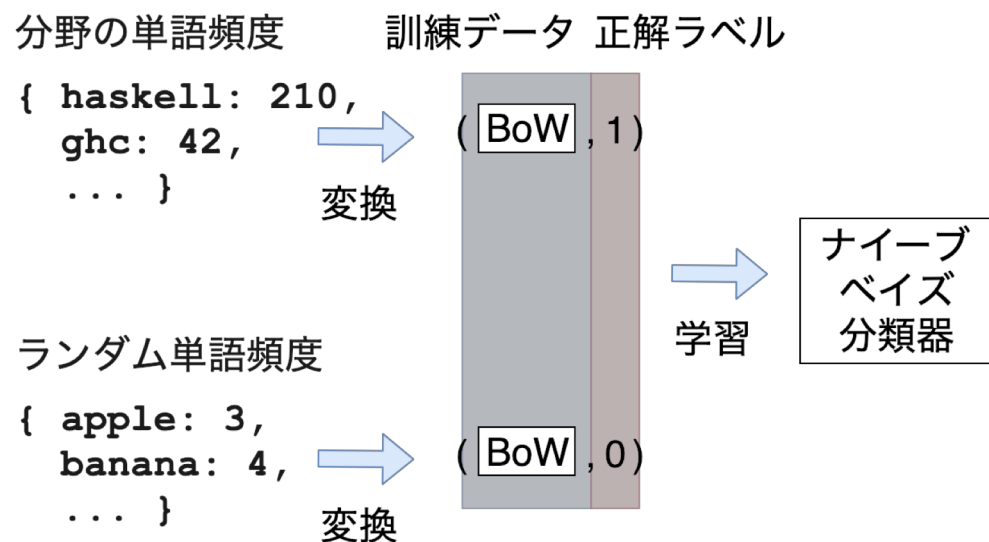
# 全体構成



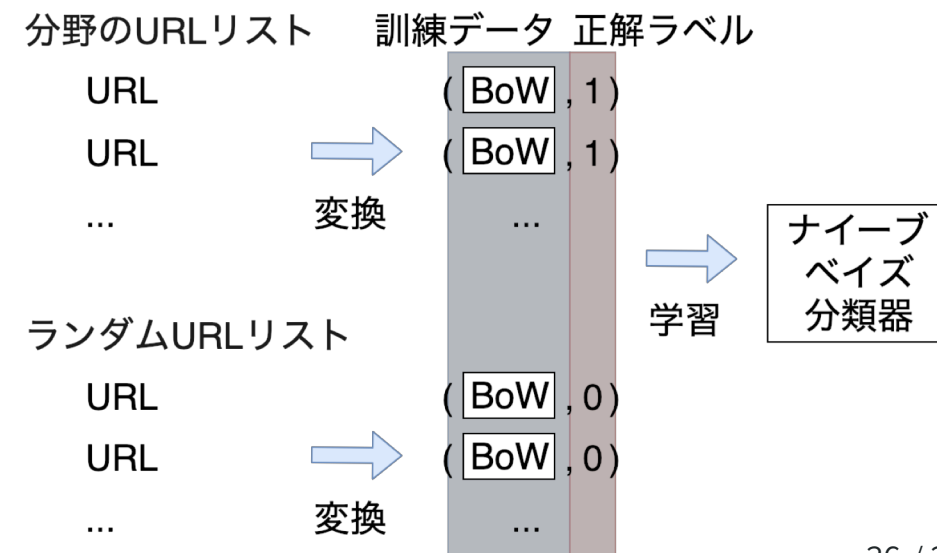
# 専門検索エンジンの動作: 分野の学習

分野をBag of Words (BoW)表現に変換し、  
ナイーブベイズ分類器を学習

## 単語頻度から学習



## URLリストから学習



# 専門検索エンジンの動作: 文書のクロール

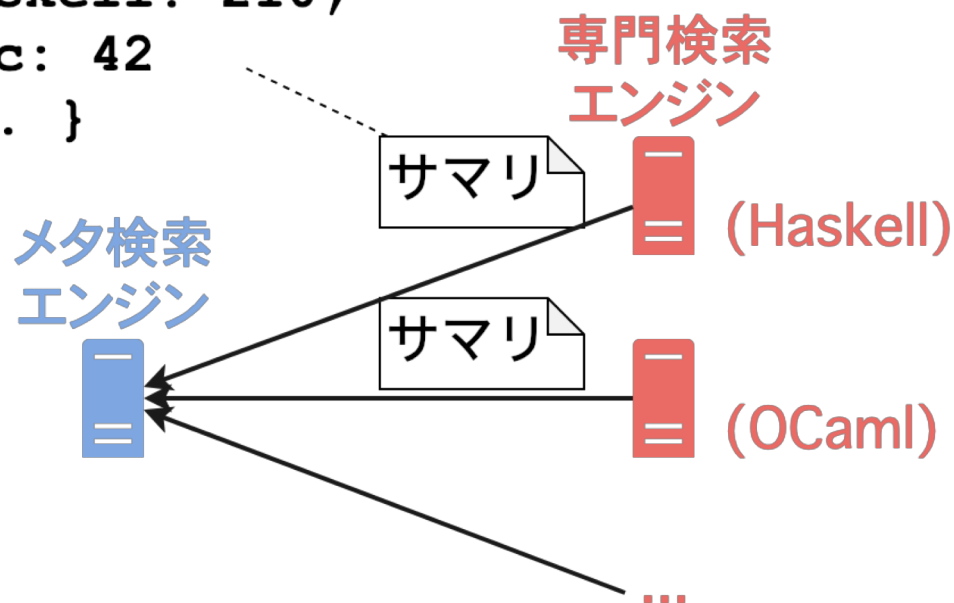
分野外の文書からはリンクを辿らないようにクロール



## メタ検索エンジンの動作: サマリ収集

専門検索エンジンがクロールした文書の単語出現頻度を分野のサマリとして収集

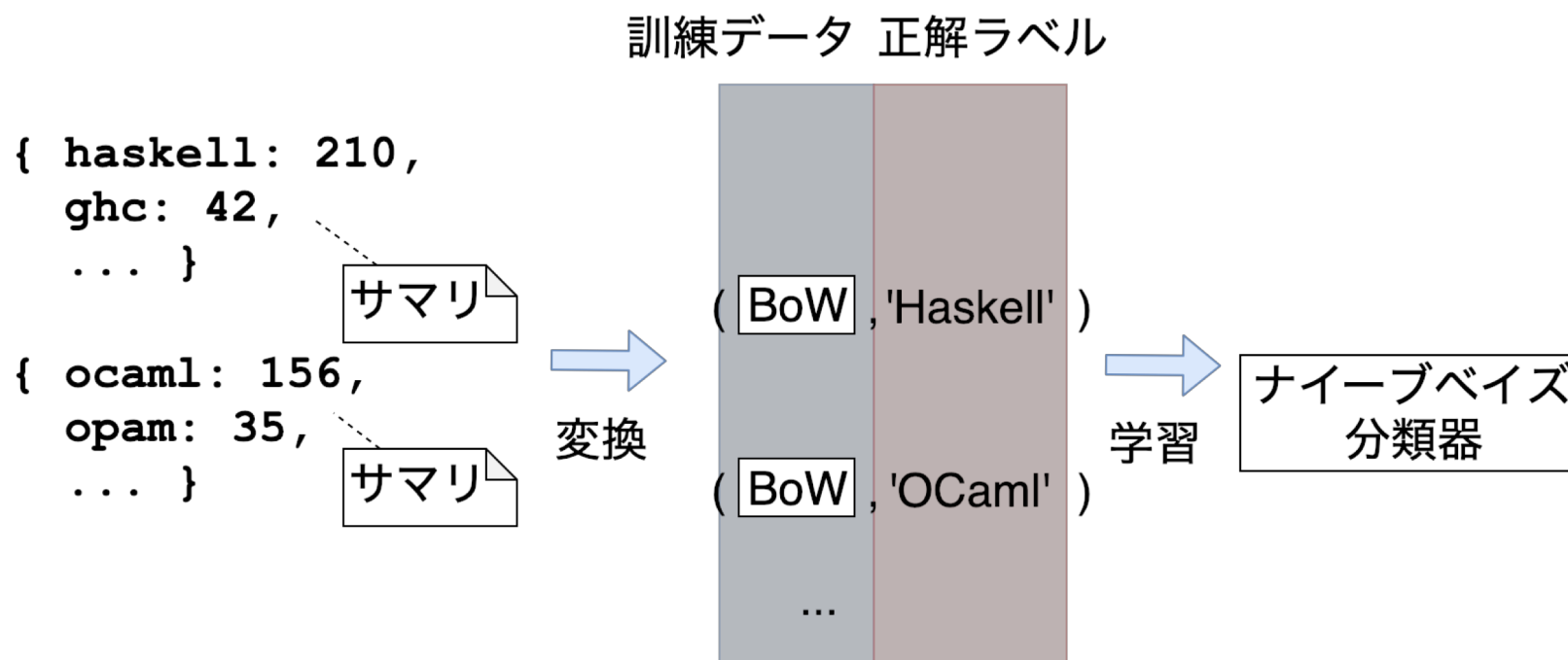
```
{ haskell: 210,  
  ghc: 42  
  ... }
```





# メタ検索エンジンの動作: クエリ分類器の学習

各分野のサマ리를BoW表現に変換し、  
ナイーブベイズ分類器を学習



## メタ検索エンジンの動作: クエリの分類

クエリをBoW表現に変換し、  
各分野に分類される確率を計算

クエリ

"stack tutorial"

```
{ stack: 1,  
  tutorial: 1 }
```



変換

BoW



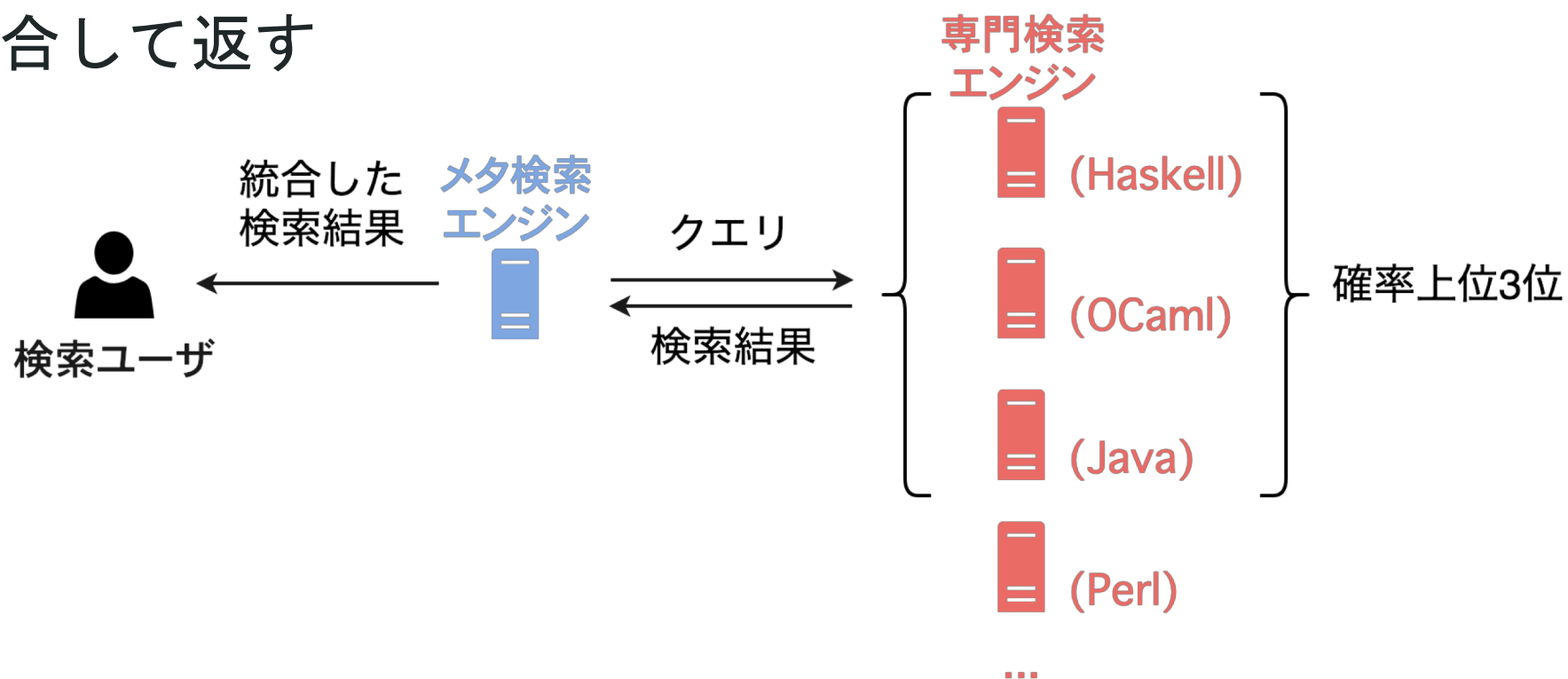
分類  
(ナイーブベイズ)

クエリが各分野に  
属する確率

Haskell	57%
OCaml	32%
...	...

## メタ検索エンジンの動作: 検索

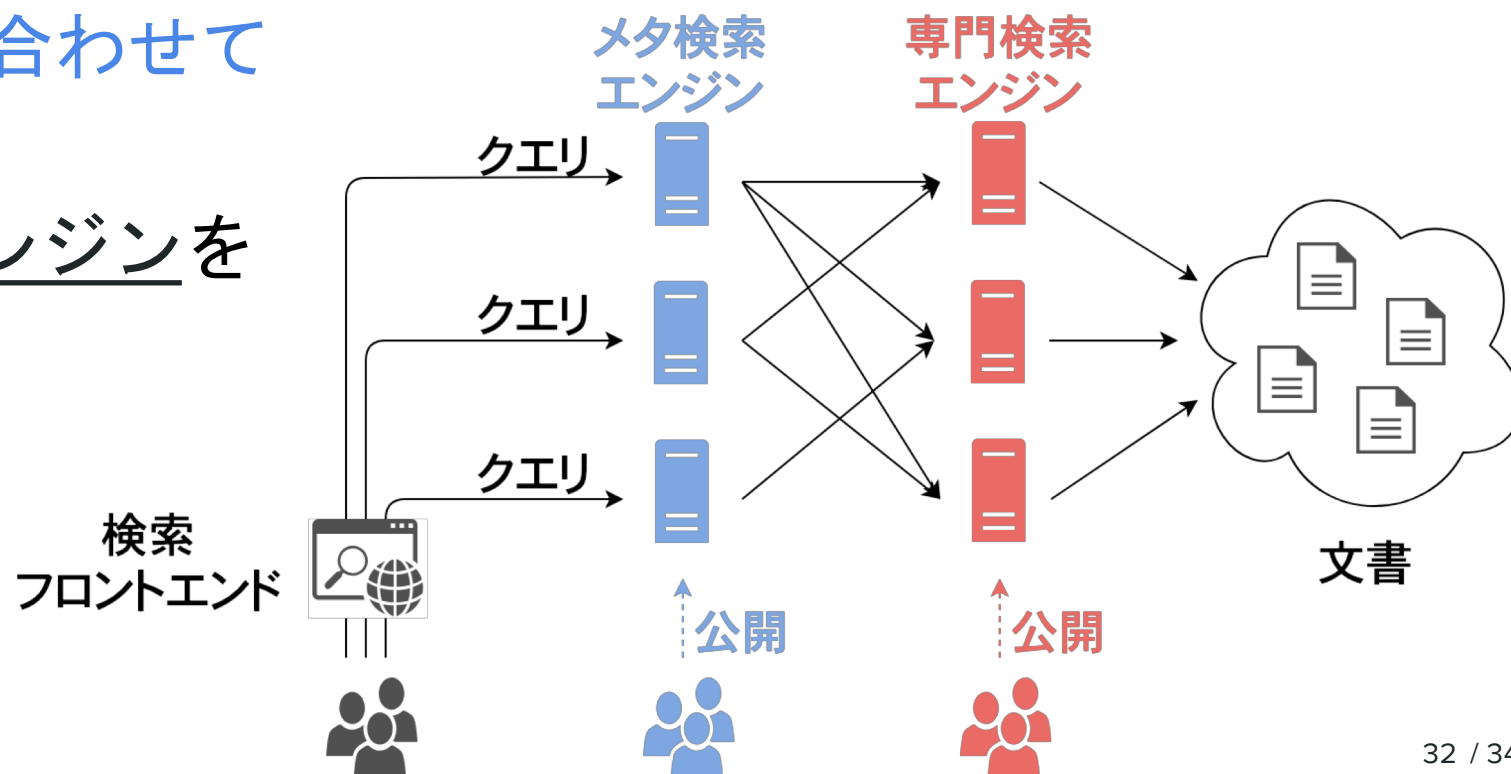
上位3分野の専門検索エンジンでの検索結果を統合して返す



# まとめ

- a. 分野を限定した検索エンジンを
- b. 複数つなぎ合わせて

自由な検索エンジンを作った



## 謝辞

- さくらインターネット 様
  - さくらのクラウド
- 石田一帆 様
  - ロゴデザイン
- Adam Jatowt 先生
  - 技術的なアドバイス
- コメント・フィードバックをくださった皆さま





# kearch

<https://github.com/kearch/kearch>

<https://kearch.info>



