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

― 自由な検索エンジンの開発 ―

1. 背景

検索エンジンはインターネットを利用する上で不可欠である。インターネットで何か調べるとき、まず検索エンジンのページを開く。多くの人々は検索エンジンで表示されたものがWeb 上にある全ての情報だと考える。そのため検索エンジンには透明性が求められる。

しかし以下の 2 つの理由により既存の検索エンジンは透明性に欠ける。1 つ目は多くの検索エンジンのランキングアルゴリズムが公開されていないこと、2 つ目は検索結果に対して行政による検閲が行われ、特定の事項に関する Web ページが表示されない可能性があることである。

これまでにも透明性の高い検索エンジンを作ろうとする試みはあった。例えば、Yacy などの分散型検索エンジンが挙げられる。Yacy は検索に必要なデータベースを分割し、世界中のコンピュータに分散して保持する。この仕組みで巨大なデータセンタ無しで検索エンジンを実現できる。

しかし Yacy は民間企業による検索エンジンに比べて利用者が少ない。これは、検索結果の品質に問題があるからである。Yacy は Web 上の全文書を分野の区別なしにデータベースに登録するが、リソース不足のため全文書を十分に網羅することができない。このため、リソースを確保できる民間企業の検索エンジンに比べて品質の劣るものになってしまう。

2. 目的

本プロジェクトの目的は、既存の検索エンジンの透明性の問題を解決することである。特に、分散型検索エンジンに分野を指定したクロール技術を導入することで、検索結果の品質を向上させ、分散型検索エンジンの利用を広める。

さらに本プロジェクトでは分野ごとに分かれた検索エンジンを結合し、より広い分野に対応する検索エンジンを作るソフトウェアも提案する。本プロジェクトの提案するソフトウェアを使えば、分野ごとの検索エンジン同士を接続して、同時に検索することができる。これにより複数の分野の検索エンジンを接続して、徐々に広い分野に対応した検索エンジンを作ることができる。最終的には数多くの分野の検索エンジンをつなぎ合わせ、透明性を担保された分野を限定しない検索エンジンを作ることを目標にする。

3. 開発の内容

本プロジェクトで開発した kearch は、図 1 のように以下の 2 種類の検索エンジンから構成される。

- 専門検索エンジン
  ある特定の分野に限定して、Web 上の文書のクローリングを行い、検索機能を提供する役割を持つ。
- メタ検索エンジン

複数の専門検索エンジンをつなぎ合わせることで、より広範囲の分野をカバーした検索機能を提供する役割を持つ。

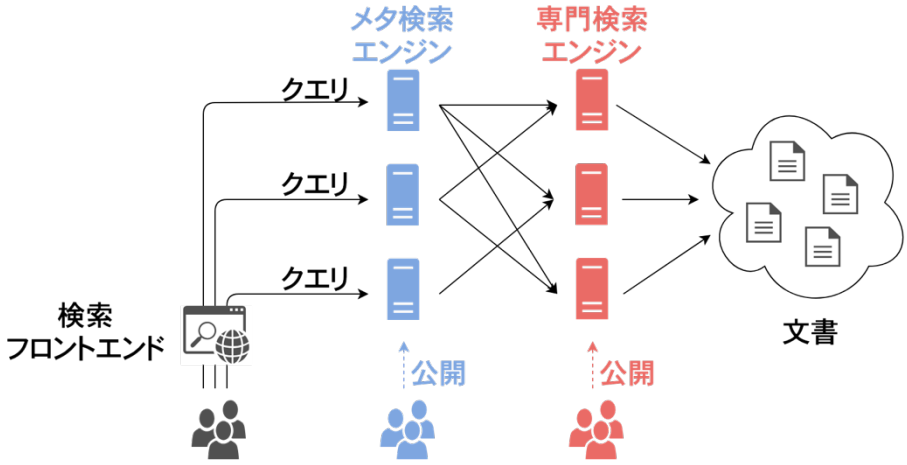

図 1. kearch の全体構成

kearch でユーザが検索を行った際の処理の流れを説明する。ユーザが検索を行うと、まずメタ検索エンジンが検索クエリを受け取る。そしてメタ検索エンジンが接続している専門検索エンジンの中から適切なものを選択し、再度検索クエリを投げる。専門検索エンジンはクロールした文書の中からクエリに適合する文書を選択し、メタ検索エンジンに返す。メタ検索エンジンは複数の専門検索エンジンの結果を統合して、ユーザに表示する。

専門検索エンジンとメタ検索エンジンをそれぞれ Kubernetes クラスタ上に構築した。Kubernetes はコンテナオーケストレーションツールであり、コンテナ管理機能、デプロイ機能、負荷分散機能、自動復旧機能が簡単に利用できる。

専門検索エンジンとメタ検索エンジンは別のクラスタ上に構築した。これは、専門検索エンジンとメタ検索エンジンは必ずしも同じユーザが作成するとは限らないためである。専門検索エンジンとメタ検索エンジンを別クラスタとすることで、それぞれの検索エンジンを別々のユーザがデプロイすることが可能である。専門検索エンジンとメタ検索エンジンは、kearch の通信プロトコルに従ってエンドポイントを設けることで、相互に通信・接続が可能である。

kearch フレームワークの中で、専門検索エンジンは特定分野の Web ページのクロールと、クロールした Web ページの検索機能を担う。専門検索エンジンの管理者は、まず専門検索エンジンの分野を決定し、その後クロールを開始する。管理者はその専門検索エンジンをメタ検索エンジンに接続することもできるし、単体の検索エンジンとして提供することもできる。

専門分野の設定にはナイーブベイズ分類器を用いる。ナイーブベイズ分類器で文書を分類するためには、どのような文書が該当分野に属していて、どのような文書が属していないのかという情報を事前に与える必要がある。kearch ではこの情報を与える方法として、URL と単語頻度の二種類の方法を用いることができる。

専門検索エンジンの管理者は図 2 のようにクロールの起点となる URL をいくつか設定する。クローラはこの URL を起点としてリンクを辿って Web 上の文書にアクセスしていく。ただし、事前に学習したナイーブベイズ分類器がアクセスした文書を該当分野外だと判断したら、その文書より先の文書はクロールしない。このようにして専門検索エンジンは Web 上の該当分野の文書だけをクロールする。

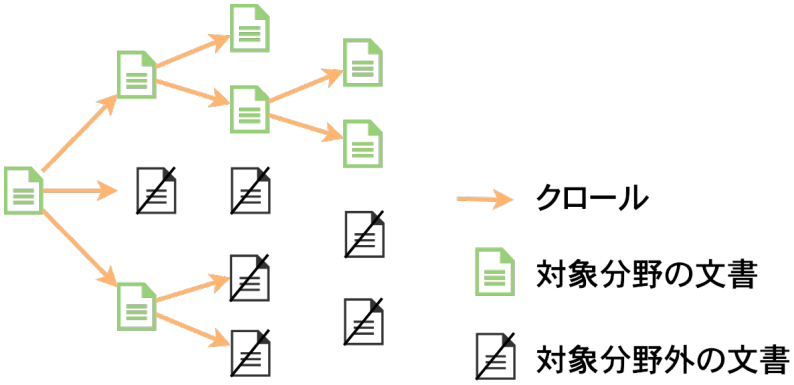

図 2. 専門検索エンジンのクロール

メタ検索エンジンは、複数の専門検索エンジンをつなぎ合わせることで、より広範囲の分野をカバーした検索機能を提供する役割を持つ。

メタ検索エンジンは接続している各専門検索エンジンからサマリと呼ばれるデータを収集する。そして、そのサマリをもとに検索クエリを専門検索エンジンに割り振る。検索クエリに対して各専門検索エンジンに適合する確率を計算し、確率の上位 3 位までに検索クエリを投げる。そしてその結果を統合してユーザに表示する。

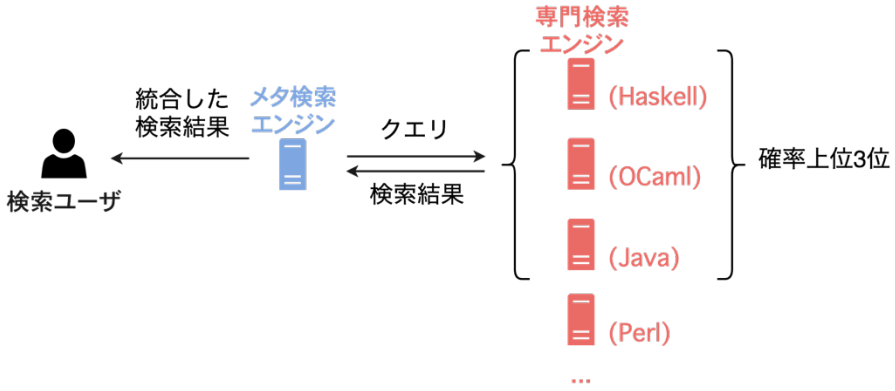

図 3. メタ検索エンジンの動作

図 3 にメタ検索エンジンの動作の様子を示す。図 3 では、メタ検索エンジンに Haskell, OCaml, Java, Perl といったプログラミング言語に関する専門検索エンジンが接続されている。ユーザが“Haskell tutorial”と検索したときについて考える。ユーザからのクエリはまずメタ検索エンジンに渡される。メタ検索エンジンは“Haskell tutorial”がどの専門検索エンジンに適合するかを計算する。この場合、Haskell、OCaml、Java、Perl の順で適合度が高いという結果になっている。次に、メタ検索エンジンは適合率上位 3 位までの専門検索エンジンで“Haskell tutorial”のクエリを使って検索し、検索結果を得る。最後に、3 つの専門検索エンジンからの結果を結合し、ユーザに表示する。

メタ検索エンジンから得られる検索結果は接続している専門検索エンジンに大きく影響される。例えば、“Haskell tutorial”での検索結果は従来型の分散検索エンジンに比べて優れているが、“Kyoto”などの接続されていない分野での検索結果は非常に劣るものとなる。

4.　従来の技術（または機能）との相違

kearch と従来の検索エンジンとの相違を述べる。

kearch と現在広く使われている検索エンジンサービスとの違いは次の 2 点である。まず、kearch はランキングアルゴリズムが公開されており、ユーザは検索結果の並び順の理由を知ることができる。更に、kearch では専門検索エンジン、メタ検索エンジンともに省リソースで動作し、大企業ではない一般人でもそれらを立ち上げることが可能である。具体的にはメモリ 8GByte、ストレージ 100GByte 程度のサーバで十分に動作させることができる。

次に Yacy などの従来の分散検索エンジンとの違いについて述べる。Yacy は分散検索エンジンではあるが、Peer to Peer ネットワーク上に構成されている。このため kearch のようにメタ検索エンジンと専門検索エンジンがツリー状に接続する構成を取れない。このため、kearch のように分野を限定してクロールし、それを段階的につないでいくことはできない。

kearch は以上の 2 点において、既存の検索エンジンよりも優れている。つまり、ランキングアルゴリズムが公開されていて、省リソースで構築を開始でき、段階的に検索エンジンの構築が可能である。

5.　期待される効果

本プロジェクトの技術的な貢献は、個人レベルでもウェブ検索エンジンの開発が可能であることを示したことにある。現在の検索エンジンの開発は大企業または巨大なコミュニティが長期間かけて行うのが常識であるが、分野を限定することで個人レベルでウェブ検索エンジンが作成できることを示した。今後、検索エンジンを開発しようとする技術者が増加することが見込まれる。

本プロジェクトの産業的な貢献は、大企業が独占的に提供する検索エンジンサービスを個人レベルで提供可能にしたことにある。検索エンジンサービス市場におけるグローバル企業の寡占状態を打破し、健全で公正な競争状態をもたらすことが期待できる。

6.　普及（または活用）の見通し

開発成果は GitHub 上にドキュメント付きで公開されており、検索エンジンのデプロイも非常に簡単なので、今後利用者は増えていくと思われる。また、個人的にも自宅のサーバで専門検索エンジンを運用していく予定である。

7.　クリエータ名（所属）

河田 旺（京都大学 大学院情報学研究科 五十嵐研究室）
稲垣 悠一（京都大学 大学院情報学研究科 大木研究室）

（参考）関連 URL

- https://github.com/kearch/kearch
  kearch のドキュメント及びソースコード