L_015

利便性の高いメタサーチエンジンの構築とその評価
Construction of Meta Search Engine with High Usability and its Evaluation

及川 啓† Kei Oikawa　西川 修平‡ Shuhei Nishikawa　児玉 英一郎† Eiichiro Kodama　王家宏† Jiahong Wang　高田 豊雄† Toyoo Takata

1.はじめに

近年，Webページの総数は110億URL以上[1]にも達し，利用者は目的のWebページを直接発見することがますます困難になってきている．このため，利用者は，通常，Webページの発見のためにYAHOO! JAPANやGoogleといった検索エンジンを利用している．しかし，現在の検索エンジンには以下の2つの代表的な問題があり，利用者にとって利便性の点で十分満足のいくものとはなっていない．

第1の問題は，現状の検索エンジンが，通常，Webページ間のリンク関係に基づいたランキングを採用しており，Webページの内容を考慮したランキングをおこなっていないため，検索結果の上位に利用者の検索意図と反したWebページが混在してしまうことである．この問題はトピックドリフト問題[2]として知られている．例えば，動物のリスについて知りたいという検索要求を持つ利用者が，「リス」という検索語を入力した場合，検索エンジンはWebページ内のテキストにリスが含まれるWebページを検索し，順位を付けて提示する．その結果，リストラやクリスマスといったWebページが検索結果の上位に提示されてしまうことが実際に起こっている．この場合，「リス」という検索語は部分文字列として含まれているが，検索要求とは一致していない．このようなことから，利用者は時間的及び精神的負担を強いられている．

第2の問題は，再検索負荷の問題である．即ち，利用者の知識に応じて何度も検索を行わなければならないということである．これは利用者ごとに持っている知識に差があるにもかかわらず，現在の検索エンジンがこれに配慮したインタフェースを有していないことに起因する．例えば，検索要求が用語の調査であり，検索結果の記述の中に利用者の知らない用語があった場合，再びその用語について検索しなければならない．

このため，本研究では，次の2点を目的としている．

第1の目的は，現状の検索エンジンにおけるトピックドリフト問題を軽減し，利用者の時間的及び精神的負担の軽減を行うことである．このためには，例えば「リス」という検索語で検索した場合，リストラやクリスマスに対応するWebページが検索結果の上位に現れずに，リスに対応するWebページが検索結果の上位に現れるような工夫を要する．

第2の目的は，検索要求に関連する概念を提示することで，利用者の再検索負荷を軽減することである．このためには，例えば「リス」という検索語で検索した場合，関連する概念としてケージなどの検索結果を同時に提示するなどの工夫が必要である．

本論文では，この2つの目的を達成するための動的再ランキング及び関連概念提示を提案し，これらを用いた利便性の高いメタサーチエンジンの構築とその評価について報告する．

†岩手県立大学大学院 ソフトウェア情報学研究科
‡株式会社エーエスエル

2.ベクトル空間モデル

ベクトル空間モデル（Vector Space Model）[3]は，情報検索アルゴリズムのひとつであり，1975年にGerald Saltonらによって提唱されたものである．このベクトル空間モデルでは，文書をベクトルで表現し，ベクトルの向きによって内容を判断する．

ある文書をベクトル化する場合，全文書中から重要語を選び，この重要語の個数を次元とするベクトルを考える．即ち，全文書からn個の重要語を選んだ場合には，n次元ベクトルが考察対象となる．この際，ベクトルの成分には重要語に対する重要度を用いることが多い．

このように文書をベクトル化することにより，同一次元のベクトル同士として文書の比較が可能となる．即ち，2つのベクトル間の類似度を算出し，文書間の類似度をベクトル間の類似度に置き換えて考えることができる．このようにして，ベクトルの類似度が高いということは文書が類似していることと同義で扱うことができる．

3.動的再ランキングの提案

前述のトピックドリフト問題は，現状の検索エンジンで利用されているランキング手法（PageRank [4]やHITS[5]など）が，主にWebページ間のリンク関係をもとにランキングを行っているだけであり，Webページの内容を考慮していないことに起因している．

このため，本研究では，Webページの内容に着目し，Webページの内容をある程度考慮したランキングを行うことを試みる．即ち，本研究では前述のベクトル空間モデルを利用し，既存の検索エンジンの通常の検索結果内のWebページに対し検索語との類似度を計算後，類似度が高いWebページを検索要求に意味が近いページとみなして，類似度の高いWebページから表示を行う．

ベクトル空間モデルを利用する場合，考察対象となる文書数を絞らないとベクトルの次元が膨大となり検索システムを構築し難いという欠点があるが，ベクトル空間モデルを適用する対象を検索結果の上位に制限することで次元を抑えて利用することが可能となる．

本研究ではWebページに対して静的にランク付けしておくのではなく，検索時に通常の検索結果内のWebページに対してベクトル空間モデルを適用し，動的にランキングを行う動的再ランキングを提案する．以下に，本提案の動的再ランキングの詳細を示す．

(1) 検索結果内の各 Web ページに含まれるテキストを形態素解析によって形態素に分解する．
(2) tf・idf 法を用いて検索結果内の各 Web ページに対する重要語の決定を行う．但し，検索語は常に重要語として扱う．
(3) (2)で得られた検索結果内のすべての Web ページにおける重要語とそれぞれの tf・idf 値を利用し，各 Web ページをベクトル化する．但し，同義語をそれぞれ個別の重要語として扱うと，それらはベクトル化したときに複数の成分に分かれてしまい，検索要求との比較の際に一部が無視され，検索要求に合致した Web ページを検索意図と反した Web ページとして扱ってしまう．そこで，セマンティック Web で利用されている Web オントロジを使用することによって，同義語を同一の語として扱い，ベクトルの次元の縮退を行う．
(4) (3)で得られた各ベクトルを利用し，検索語をベクトル化したものとの類似度を次式により決定する．

$$sim(D,\ Q) = d \cdot q$$

但し，D は Web ページ，Q は検索要求，d, q はそれぞれをベクトル化したものを表す．この際，検索要求のベクトル化は，重要語内での検索語に対応する成分を 1 とし，他の成分は 0 とすることで行う．
その後，類似度の近い Web ページから検索結果の上位に来るように再ランキングを行う．

4.関連概念提示の提案

前述の再検索負荷の問題は，現状の検索エンジンが，利用者の知識レベルに差があるにもかかわらず，これに配慮した検索結果を提示していないことに起因している．
そこで，検索要求について専門の知識を持たない利用者を想定し，検索語に関連する概念の提示を行うことで再検索負荷の軽減を図る．
このため，本研究では，Web オントロジから検索要求に関連する概念の抽出を行い，これを検索結果とあわせて提示する関連概念提示を提案する．以下に，本提案の関連概念提示で利用する関連概念の抽出手順を示す．
(1) 検索語と関連する概念を Web オントロジに問い合わせ，Web オントロジ内の owl:Class 要素の rdf:ID 属性の値に検索語があるかどうかを確認する．
(2) (1)で確認された owl:Class 要素が rdfs:Property 要素を持っているかを確認し，存在する場合，その rdfs:range 要素の rdf:resource 属性の値を検索語に関連する概念の候補とする．
(3) (2)で発見した検索語に関連する概念候補が検索結果内の Web ページをベクトル化した際に現れる重要語中に存在した場合，これを検索要求に関連する概念とする．

5.利便性の高いメタサーチエンジンのモデル

前述の動的再ランキングと関連概念提示を利用した利便性の高いメタサーチエンジンのモデルを図 1 に示す．
以下，図 1 の構成要素とそれらの動作について順に説明する．

・インタフェース
本インタフェースでは，利用者からの検索語を取得すると共に，本メタサーチエンジンによる検索結果を利用

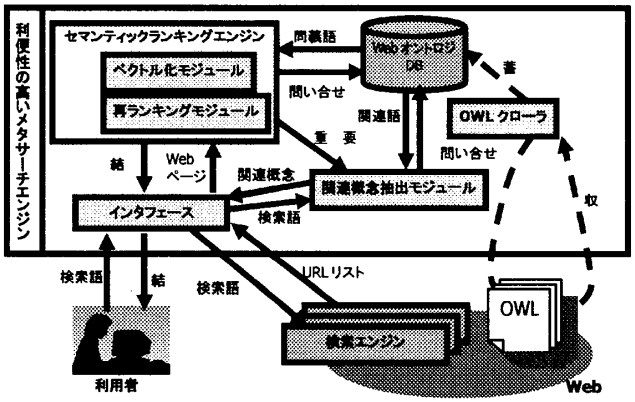

図 1 利便性の高いメタサーチエンジンのモデル

者に提示する．ここで利用者とは，検索要求に対して専門の知識を持たない利用者を想定している．
本インタフェースでは，まず，利用者から取得した検索語を既存の複数の検索エンジンと関連概念抽出モジュールに送信する．
次に，複数の検索エンジンからの検索結果を取得し，検索結果として提示された URL を抽出後，取得した URL から Web ページを得る．また，得られた Web ページをセマンティックランキングエンジンに送信する．そして，セマンティックランキングエンジンからのランキング結果と関連概念抽出モジュールからの関連概念を取得する．
最後に，本メタサーチエンジンを再度利用して，関連概念に対応する Web ページを得るための検索を行い，結果をまとめて利用者に提示する．

・セマンティックランキングエンジン
セマンティックランキングエンジンは，本提案の動的再ランキングを行う部分であり，ベクトル化モジュールと再ランキングモジュールから構成される．

➢ ベクトル化モジュール
インタフェースから取得した Web ページを，形態素解析によって形態素に分解し，前述のベクトル化の手順に従い，各 Web ページをベクトル化する．また，Web オントロジ DB に問い合わせ，検索語の同義語が存在するかを確認し，同義語が存在した場合は，ベクトルの次元の縮退を行う．その後，重要語を関連概念抽出モジュールに送信する．

➢ 再ランキングモジュール
ベクトル化モジュールによって得られた Web ページをベクトル化したものと 検索語をベクトル化したものとの類似度を計算し，動的にランキングを行う．その後，ランキング結果をインタフェースに送信する．

・関連概念抽出モジュール
インタフェースを通して取得した検索語と重要語をもとに関連概念の抽出を行う．この際，抽出は関連概念提示で示した抽出手順に従って行う．

・**OWL(Web Ontology Language) クローラ**
OWL ファイルを Web から収集し，Web オントロジ DB に蓄積する．関連する概念の決定においては，Web オントロジの情報量が重要となる．従って，Web 上に存在する既存の Web オントロジを収集，蓄積する機能が重要で

ある．

・**Web オントロジ DB**

OWL ファイルを蓄積するデータベースである．Web オントロジは，重要語の同義語を同一の語として扱い，次元を縮退する場合と，関連概念の問い合わせの際に利用する．

6.利便性の高いメタサーチエンジンの構築

本メタサーチエンジンの構築に用いたコンピュータは，Celeron プロセッサ 2.0GHz の CPU と，256MB のメモリを搭載している．ソフトウェアの主な構成は，OS として Windows 2000 Professional，Web サーバとして Tomcat4.0.6，データベースとして Xindice1.0，形態素解析ソフトウェアとして Chasen2.2.3，プログラムの開発環境として Java2sdk1.4.2 を使用した．

本メタサーチエンジンは，インタフェース，HTML パーサ，セマンティックランキングエンジン（ベクトル化クラス，類似度算出クラス），関連概念抽出クラス，OWL クローラ，Web オントロジ DB から構成され，Java を用いて実装した．以下，本メタサーチエンジンの各構成要素の実装について述べる．

・**インタフェース**

Java サーブレットを用いてサーバプログラムとして実装した．

本インタフェースでは，利用者の検索要求を複数の検索エンジンに渡すが，本実装においては YAHOO! JAPAN，Infoseek Japan，freshEYE の 3 つの検索エンジンを使用した．また，各検索エンジンの検索結果の上位 10 件を解析対象とした．

検索結果の提示方法としては，通常の検索においてトピックドリフト問題が生じない場合もあることを考慮し，通常の検索結果も提示する．そして必要に応じて，本提案の動的再ランキングと関連概念提示を使用した結果を提示できるようにした．

・**HTML パーサ**

各検索エンジンの検索結果から URL を抽出し，URL に対応する Web ページを取得する．その後，取得した Web ページに対し形態素解析を行う．

形態素解析には，「茶筌」[6]を利用した．

茶筌の呼び出しには，Process クラスの exec メソッドを用いた．検索結果からの Web ページの取得，茶筌による形態素解析部分はそれぞれスレッドとして並列に処理している．

・**ベクトル化クラス**

重要語を保持する String 型の term と重要語の tf・idf 値を保持する int 型の tf_idf をフィールドに持つ WebVector クラスを作成し，WebVector クラスの配列を作ることによって，Web ページをベクトル化する．また，検索語についても同様にベクトル化する．この際，どの Web ページをベクトル化したかの対応を取るために WebPege クラスを作成し，WebVector クラス，Web ページのタイトル，Web ページの URL，動的再ランキングによる順位を保持するよう実装した．

重要語の決定については，まず，形態素解析の結果から「名詞 一般」のものを重要語候補として抽出する．次に重要語候補に対し，tf・idf 法を用いて重要度を決定し，すべての重要語候補の重要度の平均値を求め，平均値から±1.5 の範囲のものを重要語と決定した．この範囲にしたのは，次元を 100 程度に抑えるための実装上の工夫である．

・**類似度算出クラス**

WebVector クラス内の tf_idf フィールドの値を用いて，検索語と Web ページの類似度を計算し，動的再ランキングを行う．

・**関連概念抽出クラス**

検索語と重要語をもとに，前述の関連概念抽出のルールに従って，検索語に関連する概念を Web オントロジ DB 内の OWL ファイルから抽出する．

Web オントロジ DB へのアクセスには XPath を用いた．

・**OWL クローラ**

Google のファイルタイプ検索を利用し，filetype:owl で検索を行い，OWL ファイルを収集するよう実装した．

Google へのアクセスは Java からのアクセス用 API である Google Web APIs[7]を利用した．

・**Web オントロジ DB**

データベースには，XML DB である Xindice を利用した．

7.利便性の高いメタサーチエンジンの評価

7.1 評価目的

本研究で実装した本メタサーチエンジンが，トピックドリフト問題を軽減できているか，再検索負荷が軽減できているか，利用者に対する利便性の向上が図れているかといった観点から評価を行う．また，本研究で実装した本メタサーチエンジンがどの程度の性能を持っているかを検証するため，基本性能の評価もあわせて行う．

7.2 評価方法

トピックドリフト問題が軽減できているか，再検索負荷が軽減できているか，利用者に対する利便性の向上が図れているかの 3 点を検証するために，被験者をたて，既存の検索エンジン（YAHOO! JAPAN）を比較対象とし，アンケートと時間計測による評価を行う．

トピックドリフト問題の軽減に関する評価を行うために，既存の検索エンジンと本メタサーチエンジンを用いて被験者に検索をしてもらい，被験者の検索意図に最も適合する Web ページの平均順位を比較する．

続いて，再検索負荷について評価を行うために，既存の検索エンジンを用いた場合の再検索回数と，本メタサーチエンジンを用いた場合の再検索回数の比較を行う．

次に，利便性の向上に関する時間的観点からの評価を行うために，既存の検索エンジンを用いて検索を行った場合と，本メタサーチエンジンを用いて検索を行った場合の所要時間の比較を行う．ここで，所要時間とは利用者が再検索を含め，検索意図についての知識を十分に得られるまでの時間である．従って，本評価では，既存の検索エンジンを用いる場合と，本メタサーチエンジンを用いる場合とで検索意図と検索語を変える．これは，同一の検索語を用いて評価を行った場合，2 度目の検索の際に利用者が意図する Web ページを発見するのが容易になり，本来意図しない時間短縮が行われるのを防ぐためである．

また，基本性能の評価のために，検索語 3 語を使用し，ベクトル化，再ランキング，関連概念抽出の処理時間の計測を行う．

7.3 評価結果

アンケートによる評価と所要時間の計測を被験者 10 人に対して行った．以下，評価結果について示す．

・トピックドリフト問題の軽減

既存の検索エンジンを用いた場合の結果（抜粋）を表 1 に示す．また，本メタサーチエンジンを用いた場合の結果（抜粋）を表 2 に示す．

表 1，表 2 から，既存の検索エンジンにおける検索意図に適合した検索結果の平均順位が 7.0 位（10 人の場合は 4.7 位）であったのに対し，本メタサーチエンジンでは 1.8 位（10 人の場合は 1.8 位）であることがわかる．このことから，本メタサーチエンジンによるトピックドリフト問題の軽減が確認できる．

・再検索負荷の軽減

再検索負荷に関しては，既存の検索エンジンにおける再検索回数は平均 0.6 回であり，本メタサーチエンジンでは平均 0.1 回であった．このことから，本メタサーチエンジンによる再検索負荷の軽減が確認できる．

・利便性の向上

アンケートによる利便性の 5 段階評価の結果（5 が良い評価）の平均を表 3 に示し，時間計測による所要時間の比較結果（抜粋）を表 4 に示す．

表 3 から，本メタサーチエンジンが，2.5 以上の高い利便性を持っていることがわかる．また，表 4 から，所要時間の観点において，既存の検索エンジンにおける所要時間の平均が 2 分 31 秒 78（10 人の場合は 2 分 8 秒 88）であったのに対し，本メタサーチエンジンでは 2 分 00 秒 07（10 人の場合は 1 分 50 秒 82）であることがわかる．このことから，本メタサーチエンジンが，高い利便性を持っていることが確認できる．

・基本性能

用意した 3 語の検索語を用いて，ベクトル化，再ランキング，関連概念抽出のそれぞれの処理時間を計測した．

その結果，ベクトル化の平均が約 28315.9msec，再ランキングの平均が 600.7msec，関連概念抽出の平均が 2001.5msec であった．

8.おわりに

本研究では，現状の検索エンジンの抱える問題として，トピックドリフト問題と利用者の知識レベルの差による再検索負荷の問題を挙げ，この解決策として，ベクトル空間モデルを利用した動的再ランキングと Web オントロジを利用した関連概念提示を提案した．

また，動的再ランキングと関連概念提示を利用した利便性の高いメタサーチエンジンのモデルを提案し，その実装環境を利用した評価を行った．

その結果，本提案手法により利用者の利便性が向上することがわかった．

今後の課題としては，OWL ファイルの収集法を検討し，Web オントロジの数を増やすことが挙げられる．

表 1 既存の検索エンジンを用いた場合の結果（抜粋）

	検索語	検索意図	結果内の順位
被験者 A	イヌ	動物	8
被験者 B	ペンギン	生態	9
被験者 C	ジャンプ	スキー	7
被験者 D	チャーハン	レシピ	5
被験者 E	すみれ	花	6

表 2 本メタサーチエンジンを用いた場合の結果（抜粋）

	検索語	検索意図	結果内の順位
被験者 A	マングース	動物	2
被験者 B	ハクビシン	生態	2
被験者 C	モーグル	競技	1
被験者 D	ホイコーロー	レシピ	1
被験者 E	Java	言語	3

表 3 アンケートによる評価の平均

項目	平均
関連概念提示の有用性	3.3
操作性	3.7
結果表示の有用性	4.0
体感速度	3.1

表 4 所用時間の比較結果（抜粋）

	既存の検索エンジン	本メタサーチエンジン
被験者 A	1'47''79	2'04''58
被験者 B	2'55''02	2'37''08
被験者 C	2'30''24	2'39''19
被験者 D	2'55''75	1'25''97
被験者 E	2'30''13	1'23''57
平均	2'31''78	2'00''07

参考文献

[1]Antonio Gulli, Alessio Signorini : The Indexable Web is More then 11.5 billion Pages, Proc. of 14th International World Wide Web Conference, pp.902—903 (2005).

[2] 荒谷 寛和，藤田 茂，菅原 研次：エージェントに基づくウェブページ分類の実験評価（1），電子情報通信学会技術研究報告，Vol.103, No.243, pp.49—54 (2003).

[3] Massimo Melucci, Joe Rehder : Using Semantic Annotations for Automatic Hypertext Link Generation in Scientific Texts, ISWC2003, Vol.83 (2003).

[4]S．Brin, L．Page : The Anatomy of a Large-Scale Hypertextual Web Search Engine http://www-db.stanford.edu/~176backrub/google.html

[5] Jon M. Kleinberg : Authoritative Sources in a Hyperlinked Environment, Journal of the ACM, Vol.46, No.5, pp.604—632 (1999).

[6] 奈良先端科学技術大学院大学情報科学研究科自然言語処理学講座，日本語形態素解析システム「茶筌」 http://chasen.naist.jp/hiki/ChaSen/?FrontPage

[7] Google Web APIs http://www.google.com/apis/