論 文 の 内 容 の 要 旨

論文題目　インターネット環境における
メタ・サーチエンジンに関する研究

氏　名　荻野 調

インターネットの普及はとどまるところを知らない。多くの新技術がそうであるように登場してすぐに実用となるわけではない。やっとメディアとしての基盤ができあがっても、その利用法が試行錯誤されている状況は今でも変わっていない。インターネット利用者の急増は「鶏と卵」の相乗効果の結果であり、より良い・より興味を引く・より実用的な情報発信が行われるに連れ、これまではあまり興味を持たなかった層も加わってきている。インターネットで提供されるべきサービス・必要とされているサービス・提供方法など、これまでと違った世界に適応していく必要がある。既存のインターネット上での情報検索サービスであるサーチエンジンは過去の文書検索技術を基にして、単にネット上のホームページをかき集めてインデックス化しワードマッチングしていることが多いが、このままではユーザーのニーズに応えられずに、数年以内には消滅してもおかしくない。

これまでの文書検索技術としてワードマッチングが主に用いられてきたのにはもちろん理由があるが、暗黙の了解事項として「少量の文書」「精度の高いインデックス化」「適切なキーワード」を定めていたと言える。既存アルゴリズムは文書と文書の関連精度を計算し、それに基づいて高精度の文書検索を行おうという手法が多い。この手法の場合、比較対象となる文書がユーザーの要求を正確に表したものである必要があり、また複雑な過程に時間がかかるだけの結果が得られているか、という疑問がある。実際のところ、インターネットにおける対象文書数は非常に多いため時間はなるべくかけないで済むアルゴリズムでなければならない。またユーザーが入力するキーワードは平均1.5語であり、これだけでユーザーの要求を正確に読み取るのは非常に難しい。つまり既存のアルゴリズムでは、インターネットのような文書群を対象とした検索プログラムとしては、不十分もしくは不

適切と言える。インターネットという新しい社会に合わせた新しいアルゴリズムを開発する必要があるのである。

まず、インターネットを文書母体とした検索システムにはどのような性質が備わっている必要があるかを列挙する。

- 大規模データベースに対応する省容量インデックス化手法
- 大規模データベース上の高速インデックス検索技術
- データの freshness を保つ高速アップデート機構
- 高速かつ精度の高い応答速度
- 少ないキーワードからユーザーの意図を予測するアルゴリズム
- 関連精度が高い文書のみに回答を絞り込む技術

メタ型の場合、自分ではデータベースを持たないので、1, 2, 3 番目は従来型サーチエンジンに依存する。これらは十分な成熟度にあるので、問題となる 4, 5, 6 を実現できればインターネットにふさわしい検索技術と言える。

メタ・サーチエンジンは次のような仕組みで動作する。以下に検索の流れを説明する。

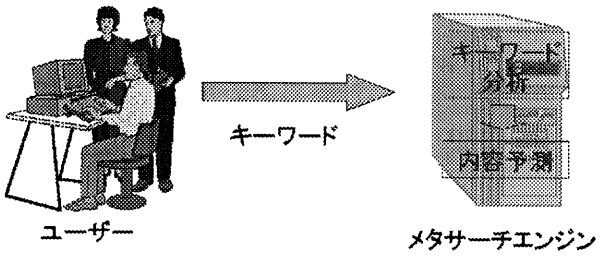


Fig.1: メタ・サーチエンジンの仕組み（1）

メタ・サーチエンジンはユーザーからキーワードを貰うとそのままサーチを実行するわけではなく、ユーザーのニーズを予測し、適当なキーワード群に置き換える。この作業は専門用語抽出や類義語辞典、単語間関連性度データベースを用いて行われる。

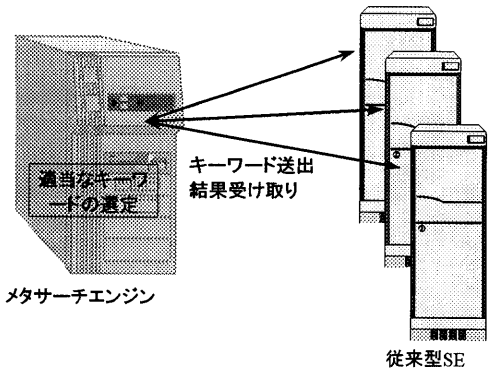


Fig.2: メタ・サーチエンジンの仕組み（2）

そしてその生成されたキーワード群を用いて各サーチエンジンに query を送る。この query の生成は各サーチエンジンの持つオプションを最大限に利用した形となる。

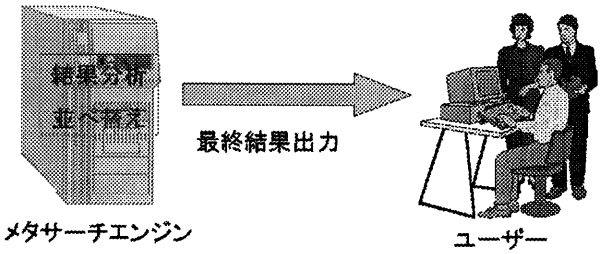


Fig.3: メタ・サーチエンジンの仕組み（3）

受け取ったリストに含まれるリンクが必ずしもユーザーの求めているものとは限らない。デッドリンクの除去とともに各リンクページを retrieve し、キーワード等のチェックを実行する。もし関連性度が低いと判断された場合にはリストから外す。

以上の作業を行うのがメタ・サーチエンジンであり、その中の一つ一つのサブ・プログラム（例：適当な検索語の選定、関連性度の算出）をいかに考えていくかがメタ・サーチエンジンとしての性能の善し悪しとなる。

本論文で提案する SmartSearch は、動作速度と出力精度のバランスを取ることを目的として開発された。いろいろな要素が考えられ、どれを選択して実装するかが重要な鍵となるが、現在の SmartSearch にはメタ・サーチエンジンには常に搭載される基本的な要素に加えて、以下のものが実装されている。

- *同一 WWW サーバー上にあるデータの数*
- *専門用語かどうかによって検索領域を限る*
- *どの従来型サーチエンジンが出したリンクかにより信頼性度を掛け合わせる*
- *ホストのタイプ(com, edu, org, etc.)から情報の種類を推測*
- *URL の長さによって重要性を推測*
- *デッドリンクを削除*
- *同一情報源の場合、結果を一つに限定*
- *サブ・メタサーチの統合*

以上をまとめて SmartSearch の処理フローを図にすると下のようになる。

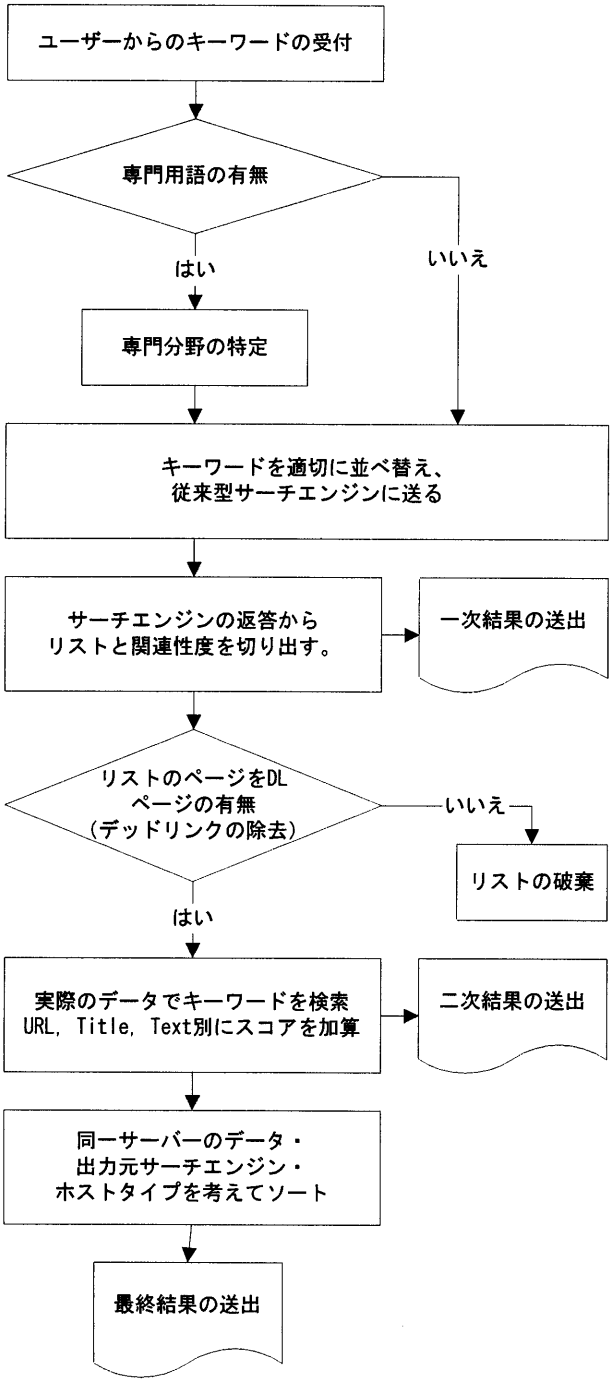
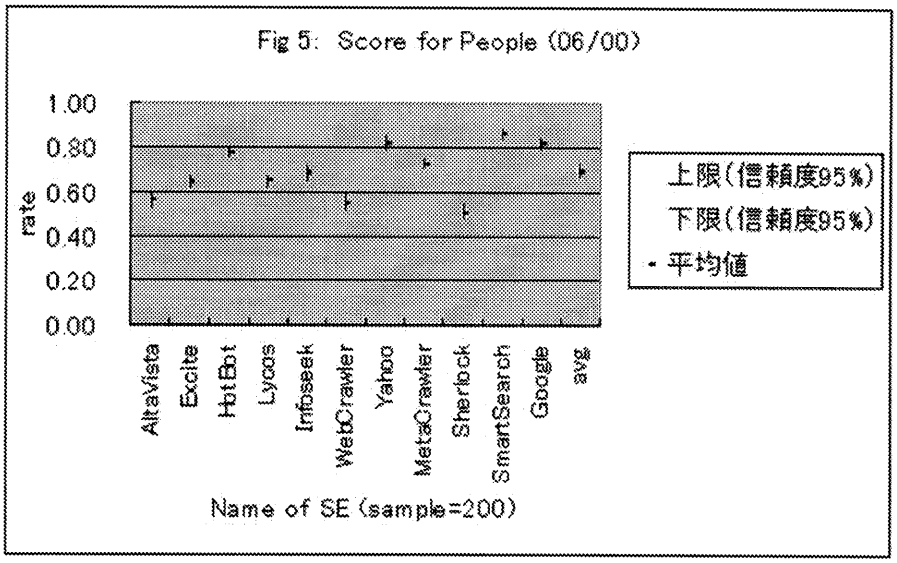


Fig.4: SmartSearch のアルゴリズム

既存の検索アルゴリズムと同様に、既存の評価方法もインターネット上の検索システム評価へと拡張するには無理がある。いずれも従来の手法である再現率（recall rate）と適合率（precision rate）を元にした新しい評価基準を設け、SmartSearch を含むインターネット上のサーチエンジンを比較した。



対人評価基準は以下の式で表される。

$$\frac{relevant - repeats - nonEnglish}{Total}$$

一方、従来型サーチエンジンをメタ・サーチエンジンの元データとした場合の価値を計る評価基準（対 MSE 評価基準）は以下の式となる。

$$\frac{relevant - \frac{1}{2}repeats - nonEnglish}{Total - dead - \frac{1}{2}repeats}$$

この評価の結論としてメタ・サーチエンジンの効果がどう得られているかを前節の結果を元に分析してみる。

- 「dead link」が多い対人評価の悪いサーチエンジンでもその中にある価値の高いリンクを抽出して利用できること
- 登録型サーチエンジンのようにカテゴリー別に仕分けされたサーチエンジンの場合、そのカテゴリーを指定して検索することによって関連性度の高いリンクを抽出できること
- 従来型サーチエンジンの持つ得手不得手分野を吸収することで、よりばらつきの少ない安定した結果を出すことができること。

評価結果より、SmartSearch がメタ・サーチエンジンの長所を生かしながら極めて良い結果を出していることを示した。しかしながら、一部のケースではまだ力が及ばないものがあった。これに対応する方法は主に 2 通りあり、一つはユーザーのキーワードから連想されるテーマを再度ユーザーに問い合わせることでテーマの限定を行う場合、もう一つはユーザーが探索したいエリアを最初から指定することによって対象となる文書を限定する方

法である。どちらも従来のサーチエンジンにおいてオプションとして提供されている機能であり、目的に応じた使い方には十分答えてくれるものではあるが、大きな難点がある。それはユーザーに多くの情報を要求しているということである。

SmartSearch においてこの問題に対する対処を次のように行う。まず、個々の分野のメタ・サーチエンジンの能力を高めることを目標とする。これまでの「General SmartSearch」では少なくとも関連する文書へのリンクが得られれば、十分としてきたが、個々の分野のみを対象とする Domain-Oriented な SmartSearch においては出力されたページの重要性が十分なものであるかを吟味することを最終目標とする。ユーザーは「General SmartSearch」にキーワードを入れるだけで、その内容が適切な「Sub-SmartSearch」に送られ、結果を返すことが可能となる。この機構を可能ならしめるためには、「General SmartSearch」における振り分け技術の開発と、各分野の「Sub-SmartSearch」の充実を図る。

現在、国連大学高等研究所と東京大学生産技術研究所は、相互協力・技術提携関係にあり、その一環として著者も国連大学の電子大学計画（Virtual University Project）に参加している。このため実際に Educational SmartSearch を作成し、以下のようなアルゴリズムをインプリメントしている。

- 学術系サイトの優先
- 電子図書館への問い合わせ
- 学術系用語の有無をチェック

Educational SmartSearch を用いることによって学術関係の検索キーワードに対する評価が高くなったのはもちろんのこと、General SmartSearch が出力する結果とは違うものが得られており、情報の幅を広げるという意味でも効果が出ている。携帯端末のように情報のやり取り自体が制約されるような端末が普及し 2000 年現在で 4000 万台に上り、PC からのアクセスを数倍追い越した。このような端末からの利用を視野に入れた場合、ここの分野別の Sub-SmartSearch がユーザーの満足度の高い結果へと導くと考えられる。