学習型 WWW 検索エンジン Verno

6Z－7

田川信一† 香月智典 竹岡厚 沼尻務 渡辺高志 上田和紀

早稲田大学大学院理工学研究科 情報科学専攻†
早稲田大学理工学部 情報学科

1 Verno の目的

WWW 上には goo[2] や Altervista[3] といった全文型検索エンジンが複数存在する。

しかし WWW 上の情報量の増加に伴い、検索結果として出力される URL 数が膨大になってしまい、検索結果全てを参照するのは不可能に近い。その一方で検索結果の中には既に存在しない URL があったり、複数のサイトに同一文書が存在していたりと、情報として無益なものも含まれていることが少なくない。

そこで本研究では、ユーザの行動履歴や HTML ファイルの特徴からユーザに有益と思われる情報の抽出を行ない、検索結果に反映させることで精度の高い検索結果を出力する WWW 全文検索エンジン Verno[1] の設計と実装を進めている。

2 Verno の特徴

Verno では WWW の jp ドメイン内に存在する HTML ファイルを検索の対象としている。任意の日本語キーワードで検索が行なえるように、全角文字は N-gram[4] 方式を用いて HTML ファイル全文をデータベース化している。検索方法は基本的にはキーワードを入力することによって行なうが、検索結果に対する付加機能として以下のようなものが実装されている。

ダイジェストの表示 検索結果には URL の他にキーワードの出現部分を、保有している HTML ファイルを参照しダイジェストとして出力する。この機能によって、検索結果上でユーザが URL の選別が可能である。(図 1)

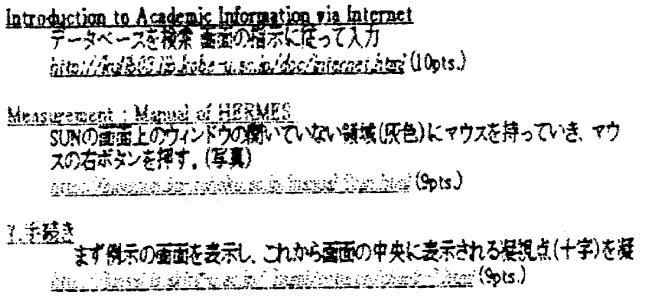

図 1: ダイジェスト表示機能

URL カテゴライズ HTML ファイル内に記述されているタグや URL ドメイン名といった情報を利用することによって、URL に対して [リンクテーブルである] や [画像主体である] といったような特徴付けを行なう。この特徴によって URL を分類し、検索要求に検索条件の指定として付加したり、階層的構造を用いた検索結果表示を行なうことを可能にした。

この結果出力画面に大量の URL 情報を出力することが可能である。

類似性の抽出 キーワードと出力 URL の関係を用いて図 2のように URL 間またはキーワード間での内容的類似性を発見する。この類似性を利用することにより、 URL のグループ化ならびに検索結果へのスコアの付加を行なう。

ソーシャルフィルタリング 検索エンジンを利用するユーザは URL を参照した後、その HTML ファイルに必要な情報がないと判断した場合、再び検索エンジンの出力結果に戻ってくるという特性がある。

Verno: A Learning WWW Search Engine.
Shin'ichi Tagawa, Tomonori Katsuki, Atsushi Takeoka, Tsutomu Numajiri, Takashi Watanabe, Kazunori Ueda.
Information & Computer Science Course, Graduate School of Science & Engineering, Waseda University.

この特性を利用して、ユーザが最後に参照したURLは情報として価値のある可能性が高いと考え、キーワードによる検索結果にスコアを付加していく。

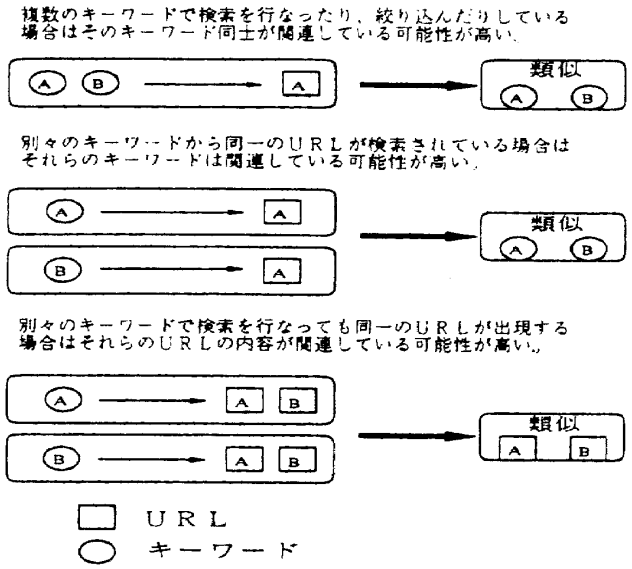

図 2: 類似性の抽出例

これらの機能によりVernoは複数のユーザに利用されればされるほど、検索精度や画面表示方法が洗練されていく検索エンジンとなっている。

3 Verno の現状

Vernoではインデックス方法がN-gram方式であり、またキーワードからURLの特定を行なった後にダイジェストの作成や学習されたデータの付加などを行なうので、検索処理に時間がかかってしまう。

そのため4台のPC(PentiumII 233MHz × 2, Pentium200MHz × 2)を図3のように配置し並列処理を行なうことによって検索速度の向上を図っている。

ユーザからの検索要求を受け取った親PEはその要求を各子PEに送る。子PEはローカルディスクにあるインデックスデータベースとカテゴライズデータベースから検索要求を満たすURLをダイジェストとともに親PEに送り返す。親PEはそれらをマージするとともに親PEが保持する学習データベースから類似性データやフィルタリングデータを付加し、検索結果の出力を行なう。

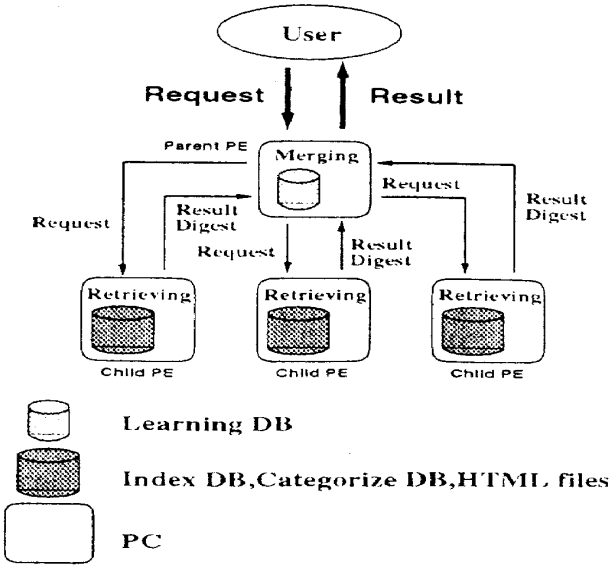

図 3: Verno の構成

各子PEが保持するデータベースはHTML文書をURLをハッシュキーとし、分散配置している。

4 まとめ

従来のキーワードによる検索に加え、ユーザの履歴を利用することによって、検索精度の高い検索結果を提供することができる。

しかしユーザの行動履歴が少ない段階では抽出されたデータの有用性も極めて低く、検索結果に反映できない。並列処理方法も現状では単独の検索要求に対する高速化は図れるが、複数の検索要求に対する高速化には言及していない。

今後はインターフェースとして簡易なプログラミング言語を用意することによって、検索条件の指定、出力方法やスコアリング方法などを自由に記述できるようにし、ユーザの思い通りになる検索環境を提供できるように改良を加えていく。

参考文献

[1] http://verno.ueda.info.waseda.ac.jp/

[2] http://www.goo.ne.jp/

[3] http://www.altavista.digital.com/

[4] 長尾 真 編: 自然言語処理, 岩波講座ソフトウェア科学 15, 岩波書店, 1996.

[5] 福田 剛志 他: データマイニングの最新動向, 情報処理, Vol.37,No.7, pp597-603, 1996.