# 検索語の重要性を考慮した重み付け手法を用いたウェブ検索支援の提案*

連 燦紅† 劉 健全‡ 陳 漢雄§ 古瀬 一隆¶

†,‡,§,¶ 筑波大学大学院システム情報工学研究科 〒 305-8577 茨城県つくば市天王台 1-1-1

## 1 研究背景

現在，情報爆発の時代に入り，ウェブ空間上に膨大な情報が存在する。欲しい情報を見つけるには，Google や Yahoo!などの代表的な検索エンジンを利用することが一般である。例えば，あるユーザが mac に関する技術の本 (book) を探したい場合，検索語を “mac book” として，これらの検索エンジンを使って検索を行う。しかし，MacBook に関するノート PC の情報が検索結果の上位に出てきてしまう。ユーザの欲しい情報が見つからない。この問題を解決するため，本研究では，ユーザの検索要望を表す検索語の重要性を考慮した独自の重み付け手法を提案し，検索語の重要性を指定可能にする，または自動判断できるようなウェブ検索支援システムを提案する。さらに，検索結果に対して従来の *tf-idf* 手法 [2] との比較実験を行い，本研究で提案する重み付け手法及びそれを用いたウェブ検索支援システムの有効性を検証する。

## 2 関連研究

ウェブ検索に対する関連研究は活発にされている。例えば，我々の先行研究では，検索語の関連度の可視化 [4] や，検索語の修正候補 [3]，インタラクティブなウェブ検索支援 [1] などをしていた。また，情報検索では，*tf-idf* 手法は汎用に使われている。この手法について，次の基本概念を簡単に説明する。$tf$(Term Frequency) は単語頻度といい，ある文書に単語の出現頻度を表す。$df$(Document Frequency) は文書頻度という。文書ごとに出現した単語の頻度を 1 回と数え，全ての文書 (N) に対する出現頻度を表す。$idf$(Inverse Document Frequency) は逆文書頻度といい，式 1 で表す。文書中の特徴語を抽出するため，式 1 で *tf-idf* 値を計算する。この計算法では，単語が重要と見なさるほど，*tf-idf* 値が大きくなる。

$$idf_t = \log(\frac{N}{df_t}),\ \textit{tf-idf}_{t,d} = tf_{t,d} \times idf_t \quad (1)$$

さらに，*tf-idf* 値を用いて，検索に対して文書のランキングを行う。よく使われている方法は，式 2 のような類似度のベクトル計算モデルである。そこで，検索対象となる文書の集合を $D = \{d_1, d_2, \cdots, d_N\}$ とする。このベクトル計算モデルでは，抽出した全ての単語の *tf-idf* を用いて，文書 $d_i$ をベクトル $\vec{d_i} = (\textit{tf-idf}_{i,1}, \cdots, \textit{tf-idf}_{i,n})$ で表す。また，クエリ $q$ を一つの短い文書とみなす。同様に，抽出した全ての単語に対して，$n$ 次元のベクトル $\vec{q}$ で表す。$\vec{q}$ では，クエリに出現した単語を値 1 とし，出現していない単語を値 0 とする。よって，文書 $d_i$ とクエリ $q$ との類似度を式 2 を用いて計算する。計算した類似度をソートすれば，文書をランキングすることができ，上位に出現する文書はクエリ $q$ と最も類似した検索結果になる。さらに，次の例を用いて，この計算モデルを詳しく説明する。

$$sim(q, d_i) = (\vec{q} \cdot \vec{d_i})/(|\vec{q}| \cdot |\vec{d_i}|) \quad (2)$$

例 1 ：簡単な例を挙げると，あるユーザが Linux における環境上の LDAP 技術を探したいときに，検索語を “Linux LDAP” として，*tf-idf* に基づくベクトル計算モデルを用いて検索を行う。クエリを $q$ とし，文書集合は $\{d_1, d_2, d_3\}$ とする。これらから抽出した単語は，windows, linux, ldap, os との 4 つであると仮定する。*tf-idf* に基づき，式 2 を用いて，それぞれの文書とクエリとの類似度を計算する。計算結果は下記の表で示す。

| | $d_1$ | $d_2$ | $d_3$ | $q$ |
|---|---|---|---|---|
| windows | 0.60 | 0.50 | 0 | 0 |
| linux | 0.10 | 0.10 | 0.35 | 1 |
| ldap | 0.70 | 0.50 | 0.30 | 1 |
| os | 0.38 | 0.70 | 0.89 | 0 |

| | $sim(q, d_i)$ |
|---|---|
| d1 | 0.57 |
| d3 | 0.46 |
| d2 | 0.42 |

しかし，ランキング結果をよく見ると，$d_3$ にはユーザの欲しい情報が最も多く含まれているのに，順番は上位にならない。windows に関する内容である文書 $d_1$ が逆に上位になる。クエリ $q$ に入っている検索語の重

*Web search support by considering the importance of input keywords
†Canhong Lian, Graduate School of SIE, University of Tsukuba
‡Jianquan Liu, Graduate School of SIE, University of Tsukuba
§Hanxiong Chen, Graduate School of SIE, University of Tsukuba
¶Kazutaka Furuse, Graduate School of SIE, University of Tsukuba

要性は均等とみなすのは理由であると考えられる。この問題を解決するため，本研究は検索語の重要性を考慮した上，重み付け手法を提案して，前述のベクトル計算モデルを改良する。

## 3 提案手法

汎用の検索エンジンでは，ウェブ上での多義検索語に関する検索結果は，発信量の強い面の情報が上位に出てしまう。このため，弱い面の情報を探したいときに，探しにくいあるいは見つからないということが生じる。このような検索を支援するため，本研究はクエリ $q$ での検索語の重要性を考え，ユーザの要望を反映するようなオペレータ $(>, <)$ を導入し，自動にまたは手動に重み付ける手法を提案する。例えば，例 1 での検索語に対して，ユーザの検索要望は Linux に関する面を強調したいとき，クエリ $q$ を $Linux > LDAP$ と書く。このような検索支援に対して，それぞれの検索語に対する重み付けの手法を提案する。

例えば，クエリ $q$ に単語 $t_i$ と $t_j$ が含まれ，ベクトル $\vec{q} = (0, \cdots, w_i, 0, \cdots, 0, w_j, 0, \cdots, 0)$ とすると，従来のベクトル計算モデルでは，$w_i, w_j$ をそれぞれ 1 を与える。本研究では，それぞれを平等ににはせず，ユーザの検索要望に従って，検索語の重要性を表すパラメーター $\alpha, \beta$ を与える。具体的には，ユーザの要望により，検索語の重要性を記号 $>$ または $<$ で表すことによって，下記の式によって，パラメーター $\alpha, \beta$ を計算する。例えば，ユーザが検索語を $t_i < t_j$ で指定する場合，単語 $t_j$ の重要性が $t_i$ より高いと考え，式 3 を使って計算する。逆に，$t_i > t_j$ を指定する場合，単語 $t_i$ の重要性を強めるため，式 4 を使う。これらの計算式では，2 つのパラメーターの和は必ず 1 と等しい。これによって，2 つのパラメーターのバランスを維持することができる。

$$\begin{cases} \alpha = min(df_i, df_j)/sum(df_i, df_j) \\ \beta = max(df_i, df_j)/sum(df_i, df_j) \end{cases} \quad (3)$$

$$\begin{cases} \alpha = max(df_i, df_j)/sum(df_i, df_j) \\ \beta = min(df_i, df_j)/sum(df_i, df_j) \end{cases} \quad (4)$$

さらに，例 1 をもとにパラメーター $\alpha, \beta$ を用いて，ベクトル $\vec{q}$ を従来のベクトル計算モデルに入れ替えて検索を行う。例えば，ユーザが入力する検索条件を "$Linux > LDAP$" に対して，式 4 を使って計算すると，$\alpha$=0.89, $\beta$=0.11 になり，$d_3$ は上位になってくる。ユーザの要望を満たすような計算結果は下記の表で示す。また，ユーザが $\alpha, \beta$ の値を指定すれば，そのまま使って計算し検索を行う。

| | $df$ | $d_1$ | $d_2$ | $d_3$ | q |
|---|---|---|---|---|---|
| windows | 4588 | 0.60 | 0.50 | 0 | 0 |
| linux | 3254 | 0.10 | 0.10 | 0.35 | 0.89 |
| ldap | 403 | 0.70 | 0.50 | 0.30 | 0.11 |
| os | 6919 | 0.38 | 0.70 | 0.89 | 0 |

| | sim(q,$d_i$) |
|---|---|
| $d_3$ | 0.43 |
| $d_1$ | 0.21 |
| $d_2$ | 0.18 |

## 4 実験と評価

提案手法の有効性を検証するため，本研究では提案手法を用いたウェブ検索支援システムを実装した。そして，実装したシステムを介して，従来の *tf-idf* 手法と提案手法との比較実験を行った。検索対象データは GOV Test Collection*を使って，全文書は 862,909 ページあり，約 14.5GB である。代表とする検索語サンプルは次のペアを選んだ。Linux>LDAP, shell(貝)<species(種類), network>ATM(Asynchronous Transfer Mode) である。2 つの手法を使って，検索結果の上位 10 件に対して，アンケートより 10 人のユーザ評価を行った。ユーザの検索要望を反映できるような精度と結果の満足度について，図 1 で示す。図 1 の結果から提案手法の有効性を示すことが出来た。

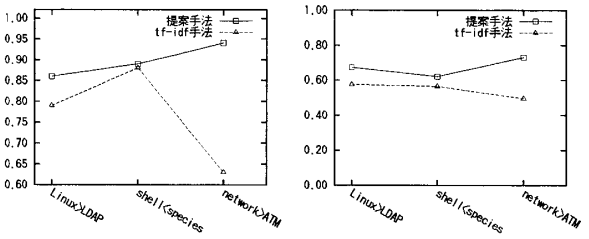


図 1: (左) 検索結果の精度，(右) 検索結果の満足度

## 5 まとめ

本論文では，ユーザの検索要望を表す検索語の重要性を考慮した独自の重み付け手法を提案した。評価実験により，提案手法の有効性を示した。今後の課題としては，汎用な重み付け法をさらに拡張したいと考えている。

## 参考文献

[1] J. Liu, H. Chen, K. Furuse, and N. Ohbo. Using an interactive interface to support web search for improving user experience. In *ICADIWT 2008*, pages 210–215, 8 2008.

[2] C. D. Manning, P. Raghavan, and H. Schutze. *An Introduction to Information Retrieval*. 2009.

[3] 劉健全, 陳漢雄, 古瀬一隆, and 大保信夫. ピクセル座標系に基づくハイパーリンク解析による修正候補の提供. In *WebDB Forum 2008*.

[4] 劉健全, 陳漢雄, 古瀬一隆, and 大保信夫. 関連度の可視的ズーミングによるウェブ検索支援. In *DEWS2008*.

*www.ted.cmis.csiro.au/TRECWeb/