中澤ゼミ

Web検索におけるきめ細かいユーザ支援システム

A1200407　蛭子小百合

1　はじめに

現在我々は，情報収集やコミュニケーション手段としてインターネットを利用し，その効率性や迅速性から多くの便益を得ている．総務省の調査によると，平成16年末の時点でインターネットの人口普及率は62.3%に達し，特に40～50歳代のユーザが大幅に増加している傾向にある[11]．インターネットユーザが，ポータルサイト[1]を利用する主な目的は情報検索であり，各サイトは検索エンジンという機能を設置している[4][6][14]．検索エンジンの利点は容易な操作で，迅速に検索結果を表示できる点である．

しかし，現在の検索エンジンは検索語入力による検索を基本としており，必ずしも使いやすいものであるとはいえない．一般的に，検索エンジン利用時に1つの検索語を入力しただけでは，膨大な検索結果が表示されてしまう．それゆえユーザが目的の情報を探し出すためには，検索結果を絞り込むための適切な検索語をさらに追加入力することが必要である．しかし，ユーザが検索対象について事前にある程度の知識を有していなければ，的確な検索語を選択できず，検索結果の過多・過少を招きやすい．この結果，ユーザは検索語の選択および入力に，多くの労力を費やさなければならない．

これらの問題点を解決し，よりよいサービスを提供するために，ポータルサイトでは様々な試みがなされている．近年着目されているのは，ユーザが入力した検索語と関連するキーワードを逐次表示する，キーワード提示システムである[5][7]．これらのシステムでは，多数のユーザによって頻繁に利用された単語をキーワードとして提示する方法が採用されている．この方法の問題点は，多義語を入力した際に認知度の高いトピックの関連語ばかりが提示される可能性が高く，認知度の低いトピックのWebコンテンツを検索する場合に的確なユーザ支援が困難となる点である．

本研究では，現行のキーワード提示システムが抱えるキーワード候補の偏りを解消し，よりよい入力支援をするために，検索頻度に依存しないキーワード提示法を提案する．

2　Web検索エンジンの現状と課題

2.1　現在のWeb検索エンジン

これまでWeb検索エンジンといえば，Yahoo!を代表とするディレクトリ型[2]と，Googleを代表とするロボット型[3]の2種類であるといわれてきた．しかし近年は，ほとんどのポータルサイトでディレクトリ型とロボット型を兼ね備えたハイブリッド型検索エンジンが主流となっている．

これらの検索エンジンでは，より信頼性の高い検索結果を提供するために，ページにランクをつけている．検索要求に対して出力される検索結果は膨大であり，単に羅列するだけではユーザの負担が大きくなってしまう．そこで，様々な方法を用いて，出力する検索結果を順位表示している．特にGoogleでは独自のPageRank[4]という判定方法を用いて，各Webページに10点満点で評価値を与えている．そして，各ページの点数と検索結果を照らし合わせ，最終的な評価値が高い順に検索結果を表示している．

2.2　検索エンジンの問題点

前述したように検索エンジンの性能は日々向上しているが，サイバー空間[5]も拡大を続けており，検索エンジンの登録インデックス数は最大規模のGoogleで80億超と膨大である[13]．当然，現状のままでは検索結果の肥大化を防ぐことはできない．たとえ検索結果が順位表示されていても，膨大過ぎる検索結果はユーザに混乱を与え，目的の情報にたどりつくまでの妨げとなる可能性がある．そのためユーザは検索結果を絞り込むために，適切な検索語を選択し，検索条件[6]を指定するなどの対応策をとらなければならない．

検索結果を的確に絞り込むためには，適切な検索語を選択すること，詳細な条件を指定することの2点が重要となる．特にどんな検索語を選択するかということは，検索結果に大きく影響する．なぜなら，適切な検索語を選択できない場合，いくら検索条件の設定を変更しても，目的のトピック[7]を対象とする結果が得られないからである．しかし適切な検索語であるかどうかの判断は，検索対象や検索方法に関するある程度の知識がなければ難しい．

3　キーワード提示システム

先に述べた検索エンジンの問題点を解消すべく研究が進められているものに，キーワード提示システムがある．キーワード提示システムとは，ユーザの検索語入力に対して関連キーワードを提示することにより，ユーザの入力を支援し，検索効率を高めるためのものである．代表的なものにgooの「キーワードアシストβ」[5]，Googleの「Googleサジェスト BETA日本語版」[7]がある．

3.1　キーワード提示方法

キーワードアシストβでは，ユーザの入力する検索語が確定すると，それに関連するいくつかのキーワードを提示する．しかし現段階では，最初に入力された検索語に関連するキーワードしか提示していないため，検索結果の絞り込みの点に問題がある．

一方，GoogleサジェストBETA日本語版では，ユーザ

1 数多くのリンク集と検索エンジンで構成されるWebサイトのこと．
2 Webサイトをポータルサイト運営者が手作業でカテゴリ別に分類し，登録する方法．
3 コンピュータが自動でWebサイトを探索し，更新・登録を行うものである．
4 リンクを張る，張られる，の関係に着目し「信頼度が高いページAからリンクを張られているページBは，ページAと同様に信頼度が高い」というように相対的に各ページの点数をつける．
5 高度なネットワークとコンピュータによって形成される電子空間のこと．
6 キーワードを「含む」「含まない」や，「AかB」，「AとB」のように様々な条件を設定できる．
7 同一の話題，テーマごとのまとまり．概念．

中澤ゼミ

が検索語入力欄に一文字入力する度に，その先の入力を予測し，随時関連のあるキーワードのいくつかを利用頻度順に表示する．例えば「ファイナンシャルプランナー」と入力する際，入力欄に「f」を入力した時点から，関連のあるキーワードの提示が始まる．そして「ふぁいなん」まで入力した時点で「ファイナンシャルプランナー」が上位に表示される．この技術によりユーザの入力文字数が減少し，関連キーワード提示に加えて更なる入力支援を実現している．

3.2 キーワード選定方法

上記の両システムでは，多くの人が繰り返し検索している単語を有益なものと考え，提示するキーワードを選定している．特に，キーワードアシストβでは，以下の条件に基づいて提示するキーワードを選定している[5].

- 対象の単語と組み合わせて頻繁に検索される単語を抽出（gooウェブ検索のログ[8]より）
- ログは過去2週間分を使用
- 一定数のユーザ以上が利用した組合せ
- 1週間利用されなければ対象より削除
- 提示するキーワードは毎日更新

3.3 既存のキーワード提示システムの問題点

Googleサジェストやキーワードアシストβのような既存のシステムは，入力された検索語に対して，最近多く組み合わせられたキーワードを提示することしかできない．これにより，認知度が高い単語と，認知度が低い単語それぞれに問題が生じている．

認知度が高い単語は，頻繁に利用される検索語であるといえる．そしてそのような検索語と組み合わせられる検索語は，やはり多くの人に利用されているといえる．したがって，提示するキーワードを検索語の利用頻度順に抽出すると，多義語を入力した際により認知度の高い特定のトピックに関するキーワードが多く提示されてしまう．例えば，「フィッシング」という検索語を入力すると，「メール，詐欺，被害，手口，犯罪，対策，セキュリティ」等のキーワードが提示される[7]．フィッシングには魚釣りのFishingと，詐欺の一種のPhishingという2つのトピックがあるにもかかわらず，より話題性のあるPhishingに関してのキーワードが多く提示されている．このように，世間の流行や利用者層の偏りによって，キーワードが左右されてしまう危険性がある．

一方認知度の低い単語は，当然検索される頻度も少なくなる．その場合，ある人が偶然組み合わせた検索語が，利用頻度の高い検索語と判断されてしまう可能性が高い．さらに現在のシステムでは，過去に検索要求がないキーワードの組合せは提示することができないため，検索頻度が低ければ十分なキーワード提示ができない．

このように現在のキーワード提示システムでは，提示するキーワードの偏りが大きく，効率的に検索結果を絞り込むために必要なキーワード提示ができない．

3.4 先行研究

現在のキーワード提示システムの問題を解決する方法として，ユーザの求めるトピックを示すキーワードを，検索結果に含まれるWebコンテンツから抽出する手法がある[15]．この手法では，単語の出現頻度や単語間の共起[9]関係を考慮し，目的の検索結果にたどりつくために追加する唯一のキーワードとして，適切と思われる単語を選定している．しかし一般的に，同じ検索語であっても，ユーザが求めるトピックは人それぞれ異なっており，検索語とトピックを一意に結びつけることはできない．その上，提示できるキーワードには限りがあるため，多様なトピックの一部しか提示できず，ユーザの要求に十分な対応ができない．

さらに，この手法で提示されるキーワードは，どのキーワードを選択しても，同じような検索結果を出力する可能性がある．なぜならば，提示されるキーワード間の関連を考慮していないため，キーワード同士が共起する可能性を否定できないからである．

以上の点を踏まえると，より良いユーザ支援とは，検索結果をいくつかのトピックに分類し，それらのトピックを示すキーワードを提示することであると考える．その際，キーワードの示すトピック間には，できる限り重なりがないことが望ましい．そのようなキーワード提示を行うことで，ユーザは調べたい情報についての知識が曖昧だったとしても，効率的に検索結果を絞り込むことができるだろう．次のセクションでは，これを実現するためのキーワード提示システムを提案する．

4 複数のトピックに対応するキーワード提示システム

本研究では，前述したキーワード提示を実現するために，最初の検索語によって絞られたWebページの集合を10個のクラスタ[10]に分割し，各クラスタを代表する単語をキーワードとして提示する．

なおクラスタを代表する単語を抽出する方法としてTFIDFを，文書間の類似度を計算する方法としてLSAを利用する．LSAについては付録A. 2を参照されたい．これらを利用することにより文書を内容の類似度で分類することが可能となり，検索結果に含まれる複数のトピックを幅広く示すことができる．

まず文書D_iに単語w_jが出現した回数をr_{ij}として，ベクトル空間Rを次のように定義する．

$$R = \begin{pmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ r_{21} & r_{22} & \cdots & r_{2n} \\ & & & \\ r_{m1} & r_{m2} & \cdots & r_{mn} \end{pmatrix}$$

ここで，行列Rのi行は文書D_iの文書ベクトル

$$D_i = (r_{i1}, r_{i2}, r_{i3}, \cdots, r_{in})$$

であり，j列は単語w_jの単語ベクトル

$$w_j = (r_{1j}, r_{2j}, r_{3j}, \cdots, r_{mj})$$

を表している．検索質問に適合した文書集合F は

8 コンピュータの利用状況やデータ通信の記録のこと．ここでは検索エンジンユーザが入力した検索語を記録したものを指す．

9 ここでは，同じ文書に出現することを共起と呼ぶ．

10 類似した文書，共通点のある文書同士のまとまり．本研究では，文書間の類似度によって分類された，1つ1つの集合を指す．

$$F = \{D_1, D_2, D_3, \cdots, D_m\}$$

となる．このベクトル空間Rに対し，次の手順で提示キーワードを抽出する．

(1) ベクトル空間$R_{m\times n}$ から，文書集合F を特徴付ける単語を抽出する

ここで$R_{m\times n}$ は，文書総数m，単語総数nによって作られるm行n列の行列である．

(2) 単語･文書行列を作成する

(3) LSAを用いて文書間の類似度を計算する

(4) 文書をクラスタリング[11]する

(5) 各クラスタの代表単語を決定する

(6) クラスタの大きさ順に代表単語を提示する

以後それぞれのステップを記述する

(1) 文書集合Fを特徴付ける単語の抽出

文書集合F に出現する全単語を対象とすることも可能だが，ベクトル空間を縮退するためにF の特徴を表す単語のみを利用する．F の特徴を表す単語については，以下のTFIDFを用いて抽出する．

ある単語w_j に対して，

$$\mathrm{TF}_{\mathrm{F}}(w_j) = \text{集合F中での単語}w_j\text{の出現回数}$$

$$\mathrm{IDF}_{\mathrm{F}}(w_j) = \frac{\text{集合Fの要素数}}{\text{集合F中で単語}w_j\text{が出現する文書数}}$$

を求め，TF×IDFの値が高い単語を抽出する．この値は，集合F 中で特定の文書に数多く出現している単語の評価値が高くなる．この値が高い単語は，集合F をいくつかのクラスタに分ける際の判断基準とすることができる．

(2) 行列を作成する

(1)で抽出した単語によって単語･文書行列$R_{m\times n}$を $R_{m\times(n-\lambda)}$ に縮退する．ここで，λは(1)で抽出されなかった単語の数である．

(3) LSAを用いて類似度を計算

類似度計算（付録A. 1を参照）の前処理として，特異値分解（付録A. 2を参照）を行い，行列$R_{m\times(n-\lambda)}$ を$R'_{m\times(n-\lambda)}$ に近似する．以降，$R'_{m\times(n-\lambda)}$ はR'と表記する．

特異値分解をすることで，類似語を多く含んでいる文書間や間接的に関連のある文書間でも適正な類似度を得ることができる．特異値分解によって算出されたR' の各行（文書ベクトル）を$D'_1, D'_2, \cdots, D'_r$ とし，新たな文書集合をF´とする．F´に含まれるすべての文書ベクトルの類似度を計算する．

(4) 文書をクラスタリングする

文書集合F´に対して，すべての文書をそれぞれ1つのクラスタとする状態から始めて，類似度の高い順にクラスタを順次結合する．そして，クラスタの数が10個になる，あるいはクラスタ間の最大類似度が0 になった時点で結合を終了する．なお，クラスタ間の類似度の定義には最長距離法[12]を用いる．類似度の定義法は多種あるが，最長距離法を用いた場合には，特定の文書以外は類似度が低いクラスタや，極端に類似度が低い文書を1つでも持つクラスタとは結合しない．そのため，結合後のクラスタにおいて文書間の類似度が極端に低くなることはない．よってこのアルゴリズムでは，強制的に10個のクラスタになるように結合するため，クラスタのサイズが大きくなる可能性が高い．類似度に大きな差異がある文書が，同一のクラスタに含まれることを避けるためにも，上述の性質を持つ最長距離法が望ましい．

以上のことを厳密に記述すると次のようになる．ここでCはクラスタの集合，mは文書の総数である．

1. $C = \{c_j \mid 1 \leqq j \leqq m\}$, $c_j = \{t_j\}$
2. $^{\forall}i, ^{\forall}j$, similarity(c_p, c_q) [13]$\geqq$ similarity(c_i, c_j) となるp, qを探し
3. $c_p = c_p \cup c_q$ ならば C から c_p を取り除く

この手順を繰り返し，

$|C|$ [14]$= 10$ または

$^{\forall}i, ^{\forall}j$, $0 \leqq$ similarity(c_i, c_j)

を満たすまで繰り返す．このとき生成された集合を，便宜上 $|c_i| \geqq |c_{i+1}|$ とする．

(5) 代表単語の決定

クラスタc_i を代表する単語w_j^*は，c_i では頻出し，それ以外のクラスタでは出現回数，出現する文書数ともに低いことが望ましい．そのため，c_i 中で多くの文書に出現し，かつc_i 内での出現回数が多いw_j^*を高く評価する．なお，c_i 以外の文書については，w_j^*の出現する文書数が少ないほど良い．なぜなら，現在の検索エンジンでは一度でもその単語が出現しているものを同一集合とみなすため，w_j^*の有無を考慮するだけでよいからである．以上のことを考慮して，以下のような変則的なTFIDFを用いて評価値を算出する．

$\mathrm{TF}_{ci}(w_j)$ = 集合c_i中での単語w_jの出現回数

$\mathrm{DF}_{cj}(w_j)$ = 集合c_i中で単語w_jが出現する文書数

$\mathrm{DF}_{\mathrm{F}'}(w_j)$ = 集合F´中で単語w_jが出現する文書数

$$\frac{\mathrm{DF}_{ci}(w_j)}{\mathrm{DF}_{\mathrm{F}'}(w_j)} \times \mathrm{TF}_{ci}(w_j)$$

この値が最も高いw_j をクラスタc_i の代表単語w_j^*とする．

(6) 代表単語の提示

以上の手順により求められた代表単語は，文書集合c_i の要素数が多い順に表示する．要素数が多い集合ほど，検索される可能性は必然的に高くなるためこの方法により，上位に提示したキーワードほど，平均的なユーザの検索要求にこたえる可能性を高めている．

5 むすび

提案したキーワード提示手法では，文書を類似度によって分類し，その分類されたクラスタを代表する語を抽出しているため，利用頻度には依存しない．そのため

11 クラスタを作成する作業を指す．

12 クラスタ間の距離を定義する方法の 1 つ．2 つのクラスタ間で，最も類似度の低い要素間の距離をクラスタの距離とする．

13 クラスタc_pとクラスタc_qの類似度．初期の段階では文書間の類似度であるが，クラスタ結合が進むにつれて，両クラスタ内で最も類似度が低いものを算出した上で類似度を求める．

14 |・|は集合の要素数を表す．

既存のシステムのように，利用頻度の高い特定のトピックに含まれる単語の羅列ではなく，検索結果中のトピックを示す単語を提示することができるようになる．これによって，ユーザの多様な検索要求に対して幅広いキーワード提示が可能になる．

例えば，既存のキーワード提示システムで「チェックイン」という言葉を1つ目の検索語とした場合を考える．この場合，2つ目以降に組み合わせるキーワードとして提示されるものは，「新橋，JAL，ホテル，チェックアウト，バリ，ホテル，時間，空港，飛行機，意味，松山」などであり，ホテルや飛行機など特定のトピックに関連する単語が多い[7]．しかし，本手法を用いることで，「チェックイン」という検索語に対して，「ホテル，空港，新幹線，コンピュータ，暗号」といったいくつかのトピックを示すキーワードを提示することが期待できる．これらのキーワードを提示することで，ユーザは事前に専門的な知識を有せずとも，目的の情報にたどりつくことができる．

今回はシミュレーションによる性能評価を実施することができなかったが，本システムによって提示されるキーワードの傾向や先行研究との比較検討などに取り組みたい．

今後，キーワード提示システムの研究がさらに進めば，単にキーワードを列挙するだけでなく，それぞれのキーワードが意味するトピックの関係や階層構造を，視覚的に提示するシステムへと発展していくだろう．

参考文献

[1] 荒木賢治，自然言語処理ことはじめ，北森出版，2004年6月.

[2] 川口真司，”潜在的意味解析法LSAを利用したソフトウェア分類システムの試作”，情報処理学会研究報告，2003-SE-140，Vol. 2003，No. 22，pp. 55-62，2003年3月.

[3] 河野浩之，情報検索とエージェント，東京電気大学出版局，2002年3月.

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

[5] http://guide.search.goo.ne.jp/beta/gka/about.html gooキーワードアシスト，goo.

[6] http://www.google.co.jp/ Google.

[7] http://www.google.co.jp/Webhp?complete=1&hl=ja Googleサジェスト，Google.

[8] http://www.kusastro.kyoto-u.ac.jp/~baba/wais/pagerank.html Googleの秘密_PageRank徹底解析，馬場肇.

[9] 神崎洋治・西井美鷹，検索エンジンのしくみ，日経BPソフトプレス，2004年12月.

[10] http://taWeb.aichi-u.ac.jp/saitom/joho/tekisutoshori.htm テキスト処理入門，齊藤正高.

[11] http://www.stat.go.jp/data/joukyou/12.htm ，家計消費調査，総務省統計局.

[12] http://www-tsujii.is.s.u-tokyo.ac.jp/enshu3/lsa.htm ，潜在的な意味解析を用いた柔軟な情報検索，東京大学理学部情報科学科辻井研究室.

[13] 村田剛志，“特集 検索エンジン 2005”，情報処理，2005年9月.

[14] http://www.yahoo.co.jp/ Yahoo Japan.

[15] 若木裕美，“検索語の曖昧性を解消するキーワードの提示手法”，情報処理学会研究報告，DBSJ Letters Vol. 4 No. 2 pp. 41-44，2005年

付録

A. LSA

潜在意味解析(Latent Semantic Analysis) 法とは，様々な文脈においての語の意味を意味空間として表現する理論であり，ベクトル空間モデル[15]で実現できなかった，間接的に関連のある単語間・文書間についても高い類似度を得ることができる[2][12].

A.1 類似度計算

任意の2つの文書ベクトルD_k，D_lに対して$cos\theta$を求めることで類似度を算出することができる.

$$cos\theta = \frac{D_k \cdot D_l}{|D_k||D_l|}$$

$cos\theta$ の値が*1*に近いほど類似度は高く，*-1*に近いほど類似度は低い．しかし，2つの文書間に全く同じ単語が含まれていない限り類似度は*0*となるため，類似語を多く含む文書間の類似度が適正なものとならない．そこでLSAでは$cos\theta$ を計算する前処理として特異値分解を採用することで，問題の解決を図っている.

A.2 特異値分解

特異値分解をすると，行列*R*を次のような行列に一意に分解することができる.

$$R = USV^T$$

$$= \begin{pmatrix} u_{11} u_{12} & \dots & u_{1l} \\ u_{21} u_{22} & \dots & u_{2l} \\ & & \\ u_{n1} u_{n2} & \dots & u_{nl} \end{pmatrix} \begin{pmatrix} s_1 & & 0 \\ & s_2 & \\ & & \\ 0 & & s_l \end{pmatrix} \begin{pmatrix} v_{11} & v_{12} & \dots & v_{1m} \\ v_{21} & v_{22} & \dots & v_{2m} \\ & & & \\ v_{l1} & v_{l2} & \dots & v_{lm} \end{pmatrix}$$

こうして分解された行列には，「$rank\ R = \eta$ のとき，上位η個の特異値のみを使って$U\ S\ V^T$を掛け合わせた結果は，最小二乗誤差になる」という性質があり，これを利用して行列*R*を

$$R_\eta = U_\eta S_\eta V_\eta^T$$

$$= \begin{pmatrix} u_{11} u_{12} & \cdots & u_{1\eta} \\ u_{21} u_{22} & \dots & u_{2\eta} \\ & & \\ u_{n1} u_{n2} & \dots & u_{n\eta} \end{pmatrix} \begin{pmatrix} s_1 & & 0 \\ & s_2 & \\ & & \\ 0 & & s_\eta \end{pmatrix} \begin{pmatrix} v_{11} & v_{12} & \dots & v_{1m} \\ v_{21} & v_{22} & \dots & v_{2m} \\ & & & \\ v_{\eta 1} & v_{\eta 2} & \dots & v_{\eta m} \end{pmatrix}$$

と変形する．この作業により，誤差を最小に抑えたまま，行列のサイズを縮小することができる．さらに，このR_η から行ベクトルを取り出し$cos\theta$ を計算することで，類似語まで含めた文書間の類似度を求めることができる.

15 文書に含まれる単語間の関連や類似度を計算するために，対象となる文書内の単語とその頻度から行列を作り，ベクトルを定義する.