Vol.2013-IOT-20 No.4
2013/3/14

検索されたWebページにおける検索語に基づく重要箇所の表示

横尾 駿—[1,a)]　吉浦 紀晃[1,b)]

概要：検索エンジンで検索を行った際に，検索された Web ページにおいて検索ワードの存在する箇所が非表示になっている場合，検索ワードがすぐに見つからない問題がある．この問題を解決するため，使用された検索ワードに基づいて Web ページ内の HTML 要素の中から最も検索ワードと関連が深いと思われる箇所を探し，その箇所を表示させるソフトウェアを開発した．結果，既存の類似したソフトウェアよりも正確に検索ワードの存在する箇所を表示し，ユーザが検索ワードを探す手間を省くことができる場合が存在することが確認できた．

Indicating Important Parts in Searched Web Pages by Retrieval Terms

Yokoo Shunichi[1,a)]　Yoshiura Noriaki[1,b)]

Abstract: Users cannot always find retrieval terms immediately in web pages that are obtained by search engines. There are several reasons, one of which is that the retrieval terms are hidden in searched web pages. To solve the problem, this paper develops software that indicates important HTML elements in searched web pages. The software finds HTML elements which are the most closely related to the retrieval terms and indicates important part in searched web pages. This paper also evaluates the software by experiments. The results of the experiments show that the software can indicate important parts more correctly than existing similar software and reduce time taken to find the retrieval terms in searched web pages.

1. はじめに

情報処理技術の発達と普及に伴い，Web ページのデザインが多様化している．HTML[1] に加えて CSS(Cascading Style Sheets)[2] や JavaScript を用いることによって，Web ページの見せ方も細かく演出できるからである．

その弊害として，作り込みが複雑な Web ページの中には，Web ページ内で，その Web ページを発見するために検索エンジンで使用した検索ワードをすぐに見つけられないものもある．例えば，タブ形式のインタフェースを使用している Web ページである．タブ形式のインタフェースとは図 1 のようにインデックス部分 (図の赤枠で囲った部分) をクリックすると表示内容が変化するというもので，現在表示状態になっていないタブに存在する文字列は検索エンジンでの検索時には考慮されているが，ページ内検索では発見できない．このような Web ページでは，ユーザが自身の使用した検索ワードの存在する箇所（=ユーザの見たい箇所）を探す必要があり手間である．またユーザによっては発見できないこともある．図 1 では「Mac 保護製品」という部分をクリックすると図 2 のように表示内容が変化する．このような Web ページにおいては，ユーザの使用した検索ワードが存在する場所をいち早くユーザに表示することが必要である．

本論文ではこの問題を解決するために行ったソフトウェア開発について述べる．次章では検索ワードが Web ページ内ですぐに発見できない原因について述べる．3 章では開発したソフトウェアについて述べる．4 章では開発した

[1] 埼玉大学大学院理工学研究科
Graduate School of Science and Engineering, Saitama University
[a)] yokoo@fmx.ics.saitama-u.ac.jp
[b)] yoshiura@fmx.ics.saitama-u.ac.jp

ⓒ 2013 Information Processing Society of Japan

図 1 タブ形式のインタフェースの例
Fig. 1 Sample of tab UI

図 2 図 1 のタブを切り替えた状態
Fig. 2 Fig. 1 that changes by clicking tab

ソフトウェアで行った実験について述べる．5 章ではまとめと今後の課題について述べる．

2. 原因

ここでは，検索に使用した検索ワードが検索された Web ページ内ですぐに発見できない原因について述べる．なお本論文における「検索ワード」とは検索エンジンに入力された文字列を空白記号で分割したものとし，検索エンジンに対する特殊なコマンドは対象としない．例えば「検索　テスト」と検索エンジンに入力した場合「検索」と「テスト」のそれぞれを検索ワードとする．また本研究では HTML に記述されたテキストデータのみを対象とし，Adobe Flash や画像などのコンテンツは対象としない．Web ページ内ですぐに検索ワードが見つからない原因として次のことが考えられる．

(1) 検索エンジンのもつ情報が最新でない
(2) 検索ワードの代わりにその同義語または類語が存在する
(3) 検索ワードが分割されて使用されている
(4) 検索ワードの存在する箇所が表示領域内にない
(5) 検索ワードの存在する箇所が表示状態でない

本論文ではこのうち「検索ワードの代わりにその同義語または類語が存在する」，「検索ワードの存在する箇所が表示領域内にない」，「検索ワードの存在する箇所が表示状態でない」の 3 つの場合に起こる「検索ワードが検索された Web ページ内ですぐに発見できない問題」を解決の対象とした．次に対象とする各原因を詳しく説明する．

2.1 検索ワードの代わりにその同義語または類語が存在する

検索エンジンのあいまい検索により，実際に使用した検索ワードではなくその同義語または類語を含む Web ページが検索されることがある．同義語または類語が検索に使用されていることに気づかない場合，ページ内検索をしてもその同義語・類語が存在する箇所が表示されず，内容に目を通して探す必要が発生してしまう．

この問題については，Google Toolbar[3] および Google Quick Scroll[4] などのソフトウェアを使用すれば，検索エンジンが判断した検索ワードと関連の深い箇所を表示させることが可能である．この 2 つのソフトウェアは，検索エンジンで検索された Web ページを表示した際に，検索エンジンが検索ワードと関連していると判断した記述の一部を最大で 3 つ程度まで抽出したふき出しを画面右下に表示し，これをクリックするとその記述が存在する箇所にスクロールするというものである．ただしこれは，Google 検索 [5] でのみ有効である．

2.2 検索ワードの存在する箇所が表示領域内にない

Web ページが縦または横に長いと，Web ページの内容がブラウザの表示領域内に入りきらないことがある．その場合はコンテンツの一部のみが表示され，ブラウザのウィンドウの端にスクロールバーが現れる．現在表示されていない部分を見るにはマウスのホイールなどを使用して表示領域を移動させる必要がある．検索ワードが表示領域内に存在しない場合，検索ワードを発見するにはこの操作を行う必要があり，ページが長い場合はその手間も増加する．

この問題は，検索エンジンの判断した検索ワードと関連が深い箇所を表示させる Google Toolbar および Google Quick Scroll の機能によってある程度解決可能である．ただし検索エンジンの判断した箇所を表示するため，検索ワードが Web ページ内に複数存在する場合，その検索ワードについて詳しく記述された箇所が表示されないこともある．

2.3 検索ワードの存在する箇所が表示状態でない

図 1，図 2 のようなタブ形式のインタフェースの存在する Web ページにおいては，表示状態になっているタブ以外の内容は切り替えを行わないと見ることができず，ページ内検索の対象にもならない．内容に検索ワードを含むタブが Web ページの表示時に表示状態になっていない場合は，検索ワードを発見するまでに時間がかかることが考えられる．

この問題については，Web ページでのユーザの操作履歴から Web ページのコンテンツ改善案を提示する研究 [6] がある．しかしこれは改善案を提示するものであり，実際に改善されることは保証されない．他には，ブラウザが読

ⓒ 2013 Information Processing Society of Japan

み込む前に Web ページの HTML ファイルを書き換えることによって，Web ページの読み込み時に表示させるタブを検索ワードの存在するタブに変更する研究 [7] がある．これは jQuery[8] および jQuery UI[9] という JavaScript のライブラリを用いて実装されたタブ形式のインタフェースに対してタブの切り替えに成功しているが，それ以外の実装によるタブ形式のインタフェースには対応していない．

3. 開発したソフトウェア

「検索ワードの代わりにその同義語または類語が存在する」，「検索ワードの存在する箇所が表示領域内にない」，「検索ワードが存在する箇所が表示状態でない」ことに起因する「検索ワードが検索された Web ページ内ですぐに発見できない問題」を解決するため，次の手順で動作するソフトウェアを開発した．

(1) ユーザが入力した検索ワードを取得する
(2) 取得した検索ワードの同義語および類語を取得する
(3) Web ページ内で最も検索ワードと関連が深い箇所を判断する
(4) 最も関連が深い箇所が表示状態か判断する
(5) 表示状態になっていなければ表示状態にする
(6) 最も関連が深い箇所がブラウザの表示領域におさまるようにスクロールする

次では，要求される各機能の実装について述べる．なお本研究では Google Chrome Extension[10] としてソフトウェアを開発した．

3.1 検索ワードを取得する機能

ユーザの入力した検索ワードは，ユーザが検索エンジンにアクセスした際の URL に含まれるクエリストリングから取得する．クエリストリングとは URL の?以降の部分のことであり，GET メソッドでサーバに送られるデータが含まれている．検索エンジンの検索結果の Web ページであれば，この部分に入力された検索ワードが含まれている．開発したソフトウェアでは，Google 検索と Yahoo 検索 [11] においてユーザが入力した検索ワードを取得できるようにした．

3.2 同義語および類語を取得する機能

同義語および類語の取得には Weblio の類語辞典 [12] を利用した．Weblio とはオンライン辞書の一つで，様々な類語データベースを同時に検索できることが特徴となっている．英語の類語データベースでは日本語で入力した場合でもその同義語および類語が得られること，日本語のデータベースでは英語の類語データベースよりも多くのデータベースから検索が行われ，元の意味からかけ離れた類語も抽出されてしまうことから，本論文では英語の類語データベースのみを使用した．また，Weblio には API のようなものが見当たらなかったので，http://ejje.weblio.jp/english-thesaurus/content/に GET メソッドで問い合わせを行い，そのレスポンスの HTML データから必要な部分を抜き出すことで同義語および類語を取得することとした．

3.3 検索ワードと関連が深い箇所を判断する機能

検索ワードの存在する箇所を表示するだけでは，ユーザが求めていた箇所を表示できるとは限らない．検索ワードの数が少なければ，全ての検索ワードが登場する箇所が Web ページ内に複数存在する可能性がある．その中には検索ワードとはあまり関係のない箇所があるかも知れない．よって，検索ワードが存在するだけでなく，より検索ワードと関連が深い箇所を表示しなくてはならない．

本論文では「検索ワードと関連が深い箇所」（検索ワードに基づく重要箇所）とは「使用された全ての検索ワードを含んでおり、Web ページ内で重要とされている箇所」とする．ただし含まない検索ワードが 1 つのみ存在する場合，その同義語または類語を含んでいれば「検索ワードと関連が深い箇所」の候補とすることとした．「1 箇所」と見なす範囲は最大で 1 つのパラグラフまでとした．具体的には HTML 上でブロック要素 [1]1 つの範囲までである．ブロック要素を含むブロック要素も 1 つのパラグラフであると言えるが，本論文では自身の子であるブロック要素内に検索ワードが存在しないときにのみ，そのブロック要素が「検索ワードを含んでいる」と見なすこととした．

また，重要とされている箇所は，その箇所の見栄えがどのように強調されているかという情報から判断することとする．その理由は，一般の文書を対象とした重要文抽出の研究 [13], [14], [15], [16], [17] と同様に HTML を対象とした重要文抽出の研究においても文脈から重要箇所を判断するものが多いが [18], [19], [20]，見栄えの情報も重要箇所を判断するにあたって有用であると判断したことと，文脈の判断を行った場合，ブラウジングにあたってストレスを感じない処理時間での実装ができないと判断したことからである．例として，見出しになっていたり太字になっていれば見栄えが強調されているとした．

3.4 表示状態になっているか判断する機能

いくつかの Web ページを調査したところ，検索エンジンによる検索時には考慮されている文字列が Web ページの表示時に非表示となるのは，その文字列を包含する HTML 要素がスタイルシートによって非表示にされている場合であった．よって，対象の HTML 要素がスタイルシートによって非表示にされていないかを調査し，表示・非表示状態の判断をすることとした．具体的には，スタイル情報の display 属性が “none”，visibility 属性が “hidden” または “collapse”，opacity 属性が “0” のいずれかに設定されている場合，ブラウザ上で表示されない．

ⓒ 2013 Information Processing Society of Japan

表 1 検索に使用された同義語または類語のうち取得できたもの
Table 1 Synonyms that are used for search and obtained by the software

元の文字列 \ 同義語・類語	Google 検索				開発したソフトウェア
速さ	速度		早さ		速度
やり方	やりかた		方法		方法
バック宙	バク宙				-
バイト	アルバイト				アルバイト
初の	最初	最初の	初めて	初めての	-

3.5 表示・非表示を切り替える機能

3.4 で述べたように，表示・非表示はスタイルシートが決めている．よってスタイルシートの情報を書き換えることによって表示・非表示を切り替えることとした．また，表示させたい HTML 要素がタブ内に存在する場合，そのタブを表示状態にするとともに現在表示されているタブを非表示にすることによって Web ページのデザインを保つよう工夫した．

3.6 ブラウザの表示領域に入れる機能

本論文では表示させたい箇所をブラウザの表示領域の中央に表示させるため，次の手順でスクロールさせることとした．

(1) 表示させたい HTML 要素の座標を取得する
(2) （その座標）−（表示領域のサイズの半分）の位置へスクロールする

4. 実験

開発したソフトウェアの有用性を評価するために行った実験の内容と結果および結果に対する考察を述べる．

4.1 同義語および類語を取得する機能の実験

同義語および類語を取得する機能の有用性を調査するため，5 つの文字列でそれぞれ検索を行い，その同義語または類語で検索された Web ページが存在した場合，その Web ページに含まれている検索ワードの同義語および類語を取得できたかを調べた．検索に使用する文字列は，略語やスラングを含むよう「速さ」「やり方」「バック宙」「バイト」「初の」の 5 つを選択した．結果を次の表に記載した．開発したソフトウェアが取得する同義語および類語の数は多くなるので，実際に検索に使用された同義語および類語の中で取得できたもののみを記載し，「-」は同義語および類語を取得できなかったことを表す．**表 1** を見るとGoogle 検索が使用した同義語および類語には取得できたものとできなかったものがあることがわかる．これは使用した類語データベースが検索エンジンの使用している類語データベースとは異なった性質のものであったためと思われる．具体的には，一般の類語データベースは入力が正確であることを前提としているが，検索エンジンの使用する類語データベースは間違った表記や未変換での表記にも対応しているためであると考えられる．このことから，検索エンジンの使用する同義語および類語を取得するには一般の類語データベースは適さないことがわかった．

4.2 検索ワードと関連が深い箇所を判断する機能の実験

検索ワードと関連が深い箇所を判断する機能の有用性を評価するため，比較対象として Google Quick Scroll を使用し，Google Quick Scroll が判断した検索ワードと関連が深い箇所と，開発したソフトウェアの判断した最も検索ワードと関連が深い箇所を比較した．以降の各段落の先頭で述べる 5 つの検索ワードおよびそれによって検索された Web ページの中からそれぞれ 1 つずつを選んで実験を行った．具体的なページのデザインおよび実際に検索ワードが存在する箇所等については本論文末の URL を参照されたい．また，Web ページ自体が実験時と異なっている可能性があることもご了承いただきたい．

「バンシィ」と入力して検索したところ, [21] の Wikipedia のページが検索された．この Web ページでは Google Quick Scroll はページの上部の「バンシィという通称をもつものも存在する」といった旨の箇所を抽出したふき出しを表示した．開発したソフトウェアはページ中程にある「バンシィ」という文字列が見出しになっている箇所を表示した．これについては開発したソフトウェアの方が検索ワードについて詳しく記述された箇所を表示できた．

「NVIDIA GeForce GTX 690　ベースクロック」と入力して検索したところ，[22] の製品のページ検索された．この Web ページでは Google Quick Scroll は「ベース」と「クロック」が離れた位置にある箇所を抽出したふき出しを表示したが．開発したソフトウェアは表示の切り替え機能も使用し「ベースクロック」と一繋がりで記述された箇所を表示した．これについては開発したソフトウェアの表示・非表示を切り替える機能が生かせた．

「永田町異聞　小沢」と入力して検索したところ，[23] のウェブログが検索された．この Web ページでは Google Quick Scroll は Web ページ全体が検索ワードに関連していると判断し動作しなかった．開発したソフトウェアは「小沢」という文字列が見出しに入っている記事を表示した．これについてはどちらの方が便利と感じるかは個人差が大きいと考えられるが，開発したソフトウェアの見出しを表示するというのは設計通りであり，要求通りの機能が得られている．

「横尾駿一　大学」と入力して検索したところ，[24] の Wikipedia のページが検索された．この Web ページでは Google Quick Scroll は「俊一」という記述のある箇所を抽出したふき出しを表示した．開発したソフトウェアは「横尾駿一」という記述を発見できずその旨を記したダイアログを表示した．このページには「横尾駿一」なる人物に関係した記述は存在しないので，開発したソフトウェアの動

ⓒ 2013 Information Processing Society of Japan

作の方が良い．

「successful failure kobo」と入力して検索したところ，[25] のウェブログが検索された．この Web ページには「kobo」という文字列を含む記事が 3 つ存在するが，Google Quick Scroll は上から 2 番目と 3 番目の記事の一部を抽出したふき出しを表示した．開発したソフトウェアは一番上の記事の見出しを表示した．これが Google Quick Scroll の仕様なのかは不明であるが，読み込み時に表示領域に入っている箇所と同等に重要な箇所があればそこへスクロールするという選択肢が存在することは開発したソフトウェアにはない利点である．

4.3 表示・非表示を切り替える機能の実験

表示・非表示を切り替える機能の有用性を評価するため，該当の機能のみを抜き出したソフトウェアを別で制作し，表示・非表示を切り替えるインタフェースの存在する Web ページに対して，表示状態になっていない部分に含まれる文字列を指定して表示させようとすることで実験を行った．ページ遷移を行わない表示・非表示を切り替えるインタフェースが存在する Web ページは全部で 47 ページ発見できた．内訳はタブ形式が 45，折りたたみ形式が 2 であった．そのうち 41 の Web ページで表示の切り替えに成功し，6 の Web ページで失敗した．失敗した原因は

- タブの実装が，タブのインデックス部分をクリックしたときに Ajax によって別の Web ページから内容を読み出すものである
- タブの実装が，スタイルシートの float 属性と overflow 属性を組み合わせたものである

のどちらかになっている場合であった．失敗した原因のうち，Ajax によって通信を行う形式のものについては，もとの Web ページ内に検索ワードがそもそも存在しないので，実際の検索時に検索されるとは考えにくく除外して良いと思われる．スタイルシートの float 属性と overflow 属性を使った形式のものについては調査不足であった．これについては今後対応したい．

4.4 手間をどれだけ軽減できるかの実験

検索された Web ページ内で検索ワードを発見するまでの手間をどれだけ軽減できるかを調査するため，コンピュータが苦手な人 2 人，得意な人 2 人の計 4 人のユーザが Web ページが表示されてから検索ワードの存在する箇所を発見するまでに要した時間と，開発したソフトウェアを使用してその箇所を表示させるのに要した時間とを比較した．ユーザが発見するまでの時間とソフトウェアが表示するまでの時間を比較したのは，同じユーザに同じ Web ページに対してソフトウェアを使用して貰った場合，実験の記憶から正確なデータが得られないと考えたからである．その結果を次に示す．使用する検索ワードは，検索ワードの 1 つがタブ形式のインタフェースによって非表示になっている Web ページが検索結果の 1 番上になるようこちらで指定した．被験者 A と被験者 B はコンピュータが苦手な人，被験者 C と被験者 D はコンピュータが得意な人であり，「–」は検索ワードを 180 秒かかっても発見できなかったことを表す．**表 2** を見ると，コンピュータが苦手な人は 180 秒かかっても検索ワードを見つけられないことも多いことがわかる．発見までの時間は個人差も大きいが，コンピュータが苦手な人は 60 秒以上，得意な人でも 10 から 20 秒程度は要することがわかった．

表 2 検索ワードを発見するまでに要した時間
Table 2 Time taken to find retrieval terms

Web ページ \ 被験者	被験者 A	被験者 B	被験者 C	被験者 D
Web ページ A	–	175sec	63sec	23sec
Web ページ B	60sec	49sec	18sec	11sec
Web ページ C	–	–	26sec	17sec

表 3 表示までに要した時間
Table 3 Time taken to indicate important parts

Web ページ \ 表示させた者	筆者	開発したソフトウェア
Web ページ A	3sec	1sec
Web ページ B	4sec	1sec
Web ページ C	10sec	1sec

開発したソフトウェアを使用した場合に，Web ページの表示が始まってから検索ワードが存在する箇所を表示させるまでに要した時間を以下に示す．参考までに，検索ワードがどこに存在するかをあらかじめ把握している筆者がその箇所を表示させるのに要した時間も記載する．**表 3** を見ると，人間の手で行う場合に比べて速く表示を切り替えることが可能なことがわかる．ユーザが発見するまでの時間とソフトウェアを使用した場合の表示までの時間を比較しているため，ソフトウェアを使用した場合に実際に発見するまでの時間は表 3 に記載した時間に加えて多くて数秒程度かかることが予想されることに注意が必要である．また開発したソフトウェアは Web ページの読み込みが完了してから動作するが，実験に使用したページは全て読み込み完了から 1 秒以内に動作が終了した．よって，それを加味しても検索ワードを発見するまでの手間を 10 秒から 100 秒以上短縮できると考えられる．

5. まとめ

本研究では「検索ワードの代わりにその同義語または類語が存在する」，「検索ワードが存在する箇所が表示領域内にない」，「検索ワードが存在する箇所が表示状態でない」ことによって起こる「検索エンジンで検索を行った際に，ユーザが検索された Web ページ内で検索ワードを発見するまでに時間がかかる問題」を解決することを目的とした

ⓒ 2013 Information Processing Society of Japan

ソフトウェアを開発した．

ソフトウェアには主に「同義語および類語を取得する機能」と「検索ワードと関連の深い箇所を判断する機能」および「表示状態を切り替える機能」を実装し，検証を行った．同義語および類語を取得する機能については，インターネット上に存在する公開された類語データベースを使用したが，一般の類語データベースの使用は検索エンジンが使用する同義語および類語を取得する手段としてはあまり適さないことがわかった．検索ワードと関連の深い箇所を判断する機能は，検索ワードがどこに存在するかとその箇所がどのように修飾されているかで判断することとし，Google Quick Scroll との比較実験を行った．結果，検索ワードと関連の深い箇所を Google Quick Scroll よりも的確に判断できる場合が存在することが確認できたが，Google Quick Scroll の方がより良い動作をする場合も発見できた．表示状態を切り替える機能は HTML 要素のスタイル情報を書き換えることによって実現し，発見したほとんどの Web ページで表示の切り替えに成功したが，一部対応してない Web ページも存在した．最終的に，いくつかの Web ページで実験を行ったところ，ユーザが検索ワードを探す手間を省ける場合が存在することが確認できた．

今後の課題としては

(1) 同義語および類語を取得する機能の改善
(2) 表示・非表示を切り替える機能の改善
(3) 他の検索エンジンへの対応

の 3 つが挙げられる．

同義語および類語を取得する機能については，単純な同義語および類語だけでなく，検索エンジンのあいまい検索が対応している表記揺れにも対処する必要がある．

表示・非表示を切り替える機能については，スタイルシートの float 属性と overflow 属性を使ったタブ形式のインタフェースに対応するとともに，さらなる調査を行い，他の実現方法が存在しないかどうかを確認したい．

また Google 検索以外の検索エンジンでも動作する点は開発したソフトウェアの利点であるので，現在対応している Yahoo 検索以外の検索エンジンにも対応させたい．

参考文献

[1] HTML 4.01 Specification, http://www.w3.org/TR/1999/REC-html401-19991224/.
[2] Cascading Style Sheets, http://www.w3.org/Style/CSS/.
[3] Google Toolbar, http://www.google.com/intl/ja/toolbar/ie/index.html.
[4] Google Quick Scroll, https://chrome.google.com/webstore/detail/google-quick-scroll/okanipcmceoeemlbjnmnbdibhgpbllgc.
[5] Google, https://www.google.co.jp.
[6] 木本亮司，市村　哲：ブラウザ操作履歴に基づいた Web サイト改善ツール，情報処理学会研究報告. GN, [グループウェアとネットワークサービス], Vol. 2011, No. 23, pp. 1–7（オンライン），入手先 ⟨http://ci.nii.ac.jp/naid/110008583737/⟩ (2011).
[7] 鈴木　景：検索された Web ページの表示に関する研究，埼玉大学工学部情報システム工学科卒業論文 (2011).
[8] jQuery, http://jquery.com.
[9] jQuery UI, http://jqueryui.com.
[10] Google Chrome Extension, http://developer.chrome.com/extensions/index.html.
[11] YAHOO! JAPAN, http://www.yahoo.co.jp/.
[12] Weblio 英語類語，http://ejje.weblio.jp/english-thesaurus.
[13] 任　福継，定永靖史：統計情報と文章構造特徴に基づく重要文の自動抽出，情報処理学会研究報告. 自然言語処理研究会報告，Vol. 98, No. 48, pp. 71–78（オンライン），入手先 ⟨http://ci.nii.ac.jp/naid/110002934727/⟩ (1998).
[14] 岡本　潤，石崎　俊：連想概念辞書の距離情報を用いた重要文の抽出，自然言語処理= Journal of natural language processing，Vol. 10, No. 5, pp. 139–151（オンライン），入手先 ⟨http://ci.nii.ac.jp/naid/10012426221/⟩ (2003).
[15] 大竹清敬，岡本大吾，児玉　充，増山　繁：重要文抽出，自由作成要約に対応した新聞記事要約システム YELLOW(< 特集 > 情報の検索とテストコレクション)，情報処理学会論文誌. データベース，Vol. 43, No. 2, pp. 37–47（オンライン），入手先 ⟨http://ci.nii.ac.jp/naid/110002726299/⟩ (2002).
[16] 平尾　努，前田英作，松本裕治：Support Vector Machine による重要文抽出，情報処理学会研究報告. 情報学基礎研究会報告，Vol. 2001, No. 74, pp. 121–127（オンライン），入手先 ⟨http://ci.nii.ac.jp/naid/110002934333/⟩ (2001).
[17] 平尾　努，磯崎秀樹，前田英作，松本裕治：Support Vector Machine を用いた重要文抽出法 (自然言語)，情報処理学会論文誌，Vol. 44, No. 8, pp. 2230–2243（オンライン），入手先 ⟨http://ci.nii.ac.jp/naid/110002711818/⟩ (2003).
[18] 相良直樹，砂山渡竹，谷内田正彦：HTML テキストの重要文を用いた画像ラベリング手法 (画像検索・映像データベース)，電子情報通信学会論文誌. D-I, 情報・システム, I-情報処理，Vol. 87, No. 2, pp. 145–153（オンライン），入手先 ⟨http://ci.nii.ac.jp/naid/110003171295/⟩ (2004).
[19] 柴田裕子，山内和子，石川千里，高田雅美，城　和貴：複数 Web ページの重要文抽出および直感的理解を支援するための GUI の開発，情報処理学会研究報告. MPS, 数理モデル化と問題解決研究報告，Vol. 2007, No. 128, pp. 81–84（オンライン），入手先 ⟨http://ci.nii.ac.jp/naid/110006595402/⟩ (2007).
[20] 砂山　渡，谷内田正彦：観点に基づいて重要文を抽出する展望台システムとそのサーチエンジンへの実装，人工知能学会誌，Vol. 17, No. 1, p. 98（オンライン），入手先 ⟨http://ci.nii.ac.jp/naid/20003320075/⟩ (2002).
[21] ユニコーンガンダム - Wikipedia, http://ja.wikipedia.org/wiki/%E3%83%A6%E3%83%8B%E3%82%B3%E3%83%BC%E3%83%B3%E3%82%AC%E3%83%B3%E3%83%80%E3%83%A0, (accessed 2013-02-02).
[22] GeForce GTX 690 — NVIDIA, http://www.nvidia.co.jp/object/geforce-gtx-690-jp.html, (accessed 2013-02-02).
[23] 永田町異聞，http://ameblo.jp/aratakyo/, (accessed 2013-02-02).
[24] 早稲田大学の人物一覧 - Wikipedia, http://ja.wikipedia.org/wiki/%E6%97%A9%E7%A8%B2%E7%94%B0%E5%A4%A7%E5%AD%A6%E3%81%AE%E4%BA%BA%E7%89%A9%E4%B8%80%E8%A6%A7, (accessed 2013-02-02).
[25] A Successful Failure, http://blog.livedoor.jp/lunarmodule7/, (accessed 2013-02-02).

ⓒ 2013 Information Processing Society of Japan