DEIM Forum 2009 A5-2

# Web文書の特徴語を利用したWeb Index(WIX) アタッチ支援

渋谷　雄一† 　　遠山元道 ††

† †† 慶應義塾大学情報工学科　〒 223–8522 神奈川県横浜市港北区日吉 3–14–1

E-mail: †shibu@db.ics.keio.ac.jp, ††toyama@ics.keio.ac.jp

**あらまし**　Web Index(WIX) は，キーワードとそれに対応する Web 文書をエントリとする XML ファイルである．ユーザは Web 文書を閲覧している際，この WIX ファイルとの結合（アタッチ）を行うことにより，多くのキーワードに対して目的のコンテンツをもつ Web 文書に即座にアクセスすることができる．しかし，複数のエントリに同一キーワードが存在すると，内容に関係なく同時にヒットしてしまう．そこで本研究では，Web 文書の持つ特徴語を抽出することにより，文脈を考慮して結果をランキングを行い，その精度の評価を通して，WIX システムにおける本手法の有効性を検証する．

**キーワード**　Web Index, WIX, attach

# Web Index(WIX) Attachment Support : utilize the representative words in the web documents

Yuichi SHIBUYA† and Motomichi TOYAMA††

† ††Department of Information and Computer Science, Faculty of Science and Technology,
Keio University
Hiyoshi3–14–1, Kouhoku-ku, Yokohama-shi, Kanagawa, 223–8522 Japan
E-mail: †shibu@db.ics.keio.ac.jp, ††toyama@ics.keio.ac.jp

**Abstract**　Web Index(WIX) is a XML file which has entries that are composed of keywords and a corresponding Web document. When browsing Web documents, users can quickly access the objective documents corresponding a lot of keywords in the browsing documents by attaching them to the WIX files. However, if two or more entires have same keywors then multi-hits problem occurs regardless of the context. In this paper, we proposed an attaching method which consider the context, extracting the representative words from the Web documents in the WIX files. We examine the availability of this proposal in WIX System trough accuracy of results.

**Key words**　Web Index, WIX, attach

## 1. はじめに

### 1.1 背　　景

インターネットが普及し，SNS [1] やブログ [2]，Wikipedia [3] などを利用することによって個々人が手軽に情報を発信できるようになった．現在の Web 技術におけるハイパーリンクは，Web 文書作成者が一つひとつ指定することによって実現している．そのため，ユーザはある内容についてもっと詳しい情報を得たいときには，検索エンジンを用いて適切な情報のある Web 文書を探し当てるのが一般的である．たとえば，外国語の文書で分からない単語が出現した場合には，いくつかの辞書あるいは翻訳サイトを探して調べることが多い．また専門的な文書であれば，その単語や関連語句を並べて検索エンジンにかけることによって目的の情報を探すことになる．

当研究室では Web の新しい形として，Web 文書を閲覧するユーザ主導のハイパーリングを実現する Web Index（以下，WIX と呼ぶ）を提案、開発している [7] [8] [9] [10]．WIX システムでは，Web 文書と WIX ファイルを結合することにより，キーワードをハイパーリンクに変換することが可能になる．WIX ファイルはキーワードとそれに対応する Web 文書をペアとするエントリの集合であり，XML 形式で記述する．

WIX ファイルは，企業や団体だけでなく個人での作成・公開も想定している．多種多様な WIX ファイルの中から，翻訳辞書や専門的な内容に特化した WIX ファイルを利用すれば，単語一つひとつを調べる必要がなくなり，かつ WIX ファイル作成者にフィルタリングされた信頼性の高い Web 文書に容易に到達することができる．

### 1.2　問題および目的

WIX ファイルにおけるエントリは，キーワードとそれに対応する Web 文書である．対応する Web 文書が複数ある場合，あるいは意味する対象は異なるが同じキーワードがある場合には，そうしたキーワードに対して異なる Web 文書をペアとするエントリが存在することになる．Web 文書との結合はキーワードベースで行うため，そうしたキーワードに対しては内容に関係なく複数のエントリが同時にヒットしてしまうという問題が生じる．そこで本研究では，アタッチされた文脈を考慮して結果を提示することとし，その評価により WIX システムにおける有効性を検証する．

また従来の WIX システムではキーワードの完全一致が前提であったが，現実の文書では表記ゆれなどわずかに異なる場合が多々存在する．こうした意図しないミスマッチを回避するため，あいまい一致によるアタッチを提案する．

以上の二点をもってユーザのアタッチ支援を行うことを目的とする．

### 1.3　構　　成

本稿の構成は次の通りである．2 章で Web Index(WIX) の概要を説明し，WIX システムの概略を示す．次に 3 章で本稿における WIX システムについて述べ，4 章で具体的な提案手法を示す．そして 5 章で評価および検討を行い，6 章で結論と今後の課題について述べる．

## 2.　Web Index(WIX)

### 2.1　概　　要

WIX ファイルは本の索引に例えることができる．基本的な Web Index はキーワードとそれに対応する文書のペアを並べたものである．キーワードは関係データモデルの主キーに相当し，文書は主キーに代表されるテキストや HTML，XML などのドキュメントである．例として人物に関する概念表を表 1 に示す．実際の WIX ファイルでは XML で記述する．

表 1　人　物.wix

| キーワード | ドキュメント |
|---|---|
| イチロー | http://web.51channel.tv/ |
| 浅田真央 | http://www.maomaiasada.com/message/mao.html |

例えば，ある Web 文書を閲覧しているときその文書中で紹介されている人物に関してより詳細な情報を得たいとする．このような場合には，一つひとつを検索エンジンで調べ，その結果から必要な情報のある文書を探しだして情報を収集していかなくてはならない．しかし，ここで　人物.wix　という Web Index を利用することによって，閲覧中の文書に出現した人物名に対して，対応する文書へのハイパーリンクを一斉に生成することができる．これは関係データモデルにおける外部キーと主キーに基づく結合と本質的に等しく，この結合を Web Index では「アタッチ」と呼ぶ．

辞書的なハイパーリンクを張ることのできるサイトには，はてなダイアリー [4] や Wikipedia [3] がある．はてなダイアリーは，記事を投稿すると特定のキーワードに対して「はてなダイアリーキーワード」と呼ばれる辞書型の文書へのハイパーリンクが自動的に生成されるというものであり，また Wikipedia は，記事の編集者の意図により [[ 記事を表す文字列 ]]　のように記号で囲むことによって，囲まれた文字列と一致する記事を検索し，そのような記事が存在すれば自動的にリンクを張ることができるというものである．

これらのサイトのシステムは，辞書的な役割を果たすリンクを自動的に生成する点で WIX システムと共通するが，同一ドメインに限られている点で異なる．Web Index では，ユーザはあらゆる Web 文書に対して自由に WIX ファイルをアタッチすることが可能であるため，これらのサービスよりもユーザの自由度が大きいと言える．

### 2.2　ア　タッチ

WIX システムでは，WIX ファイルと Web 文書の結合のことをアタッチと呼び，その利用形態により作者指定アタッチ，フルアタッチ，ユーザ指定アタッチが存在する．またアタッチに利用する WIX ファイルはユーザが自由に選択することが可能である．例えば同じ人物名をキーワードとするエントリをもつ WIX ファイルをアタッチに利用する場合について考える．このとき，人物の出身地に関する WIX ファイル，あるいは有名人の誕生日に関する WIX ファイル，または有名人のプロフィールに関する WIX ファイルといった各種 WIX ファイルが存在するならば，それらのどれを利用してアタッチするかを切り替えるだけで，簡単にユーザが欲する結果に切り替えることが出来る．さらに，複数の WIX ファイルを一つの文書にアタッチすることも可能であり，ユーザが予め必要な索引を用意しておきさえすれば，アタッチに利用する WIX ファイルを指定することによって，対応するキーワードがハイパーリンクに変換され，すぐさま情報を得ることが可能になる．

#### 2.2.1　フルアタッチ

キーワードに合致する文書中のすべての単語に対しリンクを生成する．Web 文書作成者の指定する WIX ファイル，またはユーザが指定する任意の WIX ファイルとアタッチを行い，WIX ファイルのキーワードに含まれるすべて単語について，Web 文書中の単語がハイパーリンクに変換される．

#### 2.2.2　ユーザ指定アタッチ

Web 文書を閲覧するユーザがアタッチを行いたい単語を指定し，指定された単語が WIX ファイル中のキーワードと合致すればリンクを生成する．

ユーザはアタッチする WIX ファイルと，詳細情報を知りたい単語を自由に指定することが可能である．閲覧している文書のあらゆる単語に適用できるためユーザの自由度が高い．ユーザが自分の要求を満たす情報を持つ WIX ファイルを知っていれば，必要な情報のみを簡単に得ることができる．

#### 2.2.3　作者指定アタッチ

Web 文書の作成者がアタッチを行いたい単語に対しアンカータグを付与し，指定された単語にのみリンクを生成する．Web 文書作成者の意図したハイパーリンクを適切にユーザに提供できる．本研究では使用しないため，詳しくは参考文献 [7] [8] を

参照されたい．

### 2.3　WIX ファイルの記述方式

WIX ファイルはエントリの集合であり，エントリとの結合によって対応するキーワードをハイパーリンクに変換する．しかし，WIX ファイルは膨大な数のエントリをもつ可能性があり，その作成や管理についても研究されている．具体的には図 1 のように記述し，1 つのエントリは 1 つ以上のキーワードと 1 つの Web 文書をもつ．

```
<entry>
  <kw>キーワード 1</kw>
  <kw>キーワード 2</kw>
  <doc>document1.html</doc>
</entry>
```

図 1　エントリの記述方式

固定的なコンテンツの場合であれば，対応する文書は文字列で直接記述すればよい．しかし，例えば地名をキーワードとする気象情報を提供するコンテンツを扱う場合には，対応する文書は時々刻々変化する性質のものとなる．そのため，Web Index のファイルタイプには大きく分けて

（1）　Static Web Index(静的 Web Index)

（2）　Dynamic Web Index(動的 Web Index)

の 2 つが提案されている．本稿では詳しい議論は行わないため，詳細は参考文献 [7] [8] を参照されたい．

### 2.4　利 用 形 態

WIX システムの将来的な利用予想図を図 2 に示す．

WIX ファイルの作成・公開の一つの形として，企業や団体が製品やサービスの WIX ファイルを提供することによって，さまざまな文書に現れるであろうそれらのキーワードを介して自社サイトに誘導するといった利用形態が考えられる．あるいは興味のある分野についての WIX ファイルをユーザが作成・公開し他のユーザと共有するといった利用形態も考えられる．そして，自ら作成した WIX ファイルや公開されている WIX ファイルを指定することで各種アタッチを行うことができる．よく利用する WIX ファイルはブックマークできるとよいだろう．

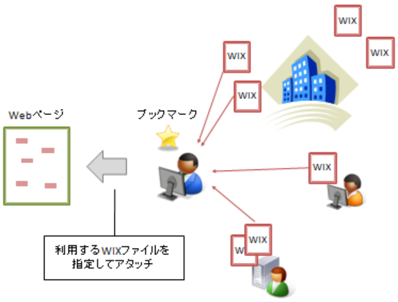



図 2　WIX システムの利用予想図

## 3.　提案システム

### 3.1　概　　要

従来の WIX システムの実装は，すべての処理をクライアント側で行うものか，あるいはサーバ側で行うものであった [7] [8] [10]．提案するシステムは，クライアントとサーバの両方で処理を連携することにより，より柔軟で高度な処理を実現することができるように実装を行った．図 3 に，本研究におけるシステムの概略図を示す．

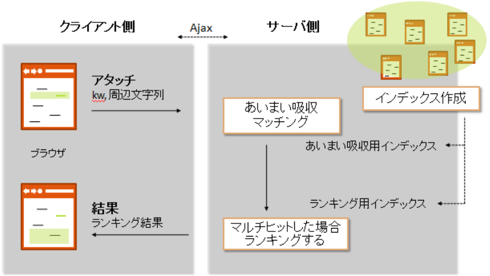



図 3　提案システムの概略図

クライアント側は Mozilla Firefox のアドオン [6] として利用可能にしている．そのため誰でも簡単に WIX システムを導入することができる．動作例は 4 章で詳しく説明するが，アタッチしたいキーワードをマウスで選択してユーザ指定アタッチを実行すると，図 3 に示すように選択されたキーワード (kw) とその周辺文字列をサーバに送信する．そしてサーバが出した結果を受け取りインラインフレーム上に表示するという流れである．インラインフレームを経由することで，クロスドメインでの Ajax 通信を実現している．ただし，本稿での実装はアタッチに使用する WIX ファイルとしてあらかじめサーバに保存したものを利用し，ユーザが自由に選ぶことは出来ない仕様になっている．

サーバは，クライアント側からキーワードと周辺文字列をアタッチ情報として受け取る．そして，使用する WIX ファイルから事前に作成したインデックスを利用して，あいまいを吸収したキーワードマッチングを行い，さらにヒットしたエントリをスコアリングしランキングする．そのランキング結果は，インラインフレームを介して，リンク先の Web 文書のタイトルを上位から順に並べることでクライアントに提示される．

### 3.2　マルチヒット問題

WIX ファイルの各エントリは一つの Web 文書をもつため，たとえば同じ対象について複数の Web 文書が存在する場合には，その Web 文書の数だけエントリが必要になる．あるいは異なる対象であっても，同一名称であったり愛称・略称が同じである場合には同様のことが言える．このように WIX ファイルとのアタッチでは，同じキーワードに対して，同時に二つ以上のエントリが結合されるため，一つのキーワードに対応する Web 文書が複数存在することになる．これをマルチヒットと呼ぶことにする．

例えば，図 4 のような WIX ファイルを利用したアタッチを

```
<entry>
<kw>イチロー</kw>
<kw>鈴木一朗</kw>
<doc>http://web.51channel.tv/</doc>
</entry>
<entry>
<kw>イチロー</kw>
<kw>鈴木一朗</kw>
<doc>http://www.d-a-strategy.com/data/
                   baseball/ichiro.html</doc>
</entry>
```

図 4　マルチヒットとなるエントリ例

考える．このとき，Web 文書中に「イチロー」または「鈴木一朗」というキーワードがあれば，それらに対して 2 つのエントリが同時にヒットすることになる．これは同じ人物であるが，キーワードさえ同じであればどのような場合でも同様である．

WIX システムは，マルチヒットの場合に異なる対象を表すキーワードが含まれる可能性を本質的に避けられない．なぜなら，ユーザが明示的に利用すると宣言した WIX ファイル集合の中でのマルチヒットは，すべてユーザの求める Web 文書であると判断されるからである．例えば,「信号」とうキーワードにもつ 2 つ WIX ファイルを考える．一方は「信号機」を表す生活に密着した内容の WIX ファイルであり，他方は「制御信号」を表す工学的な内容の WIX ファイルであったとする．これら 2 つのファイルは扱う内容の方向性が異なるため，通常ならばこれら 2 つの WIX ファイルを同時に用いてアタッチするということは少ないと考えられる．しかし，ユーザがあえて両方使ってアタッチした場合には,「信号」との結合によってヒットしたエントリはすべてユーザが求めている Web 文書と判断することになるのである．

アタッチはキーワードベースで処理されるため，こうしたヒットはアタッチした Web 文書の内容やエントリが指す Web 文書の内容には一切関係なく結合されてしまう．ところで，閲覧ページに応じた WIX ファイルをアタッチの度に指定するというのは手間である．したがって，ブックマーク登録した WIX ファイルをすべて利用する，あるいは事前に利用する WIX ファイル固定しておき，それをアタッチの際に自動的に利用するといった方法を取るべきだろう．しかし，この方法ではさまざまな方向性の WIX ファイルを同時に利用することにつながるため，実際のコンテクストとは大きく異なる結合が起こる可能性がある．ユーザは結果のリンクしか得られないため，そのような結合が行われても認知できないという問題が生じる（マルチヒット問題）．

このような問題に対して，アタッチされたキーワード周辺の文脈に応じてランキングを行い，その順にユーザに提示することで解決を図る．ユーザは上位にあるリンクを見ることにより，求める情報により速く到達できるようになる．この考え方に基づいて，本研究では WIX システムにおけるランキング方法の提案を行い，どのようなランキング手法が最も有効に機能するかを比較・検証する．

### 3.3　あいまい検索

これまでの WIX システムでは，結合キーとなるキーワードは完全一致することが前提であった．しかし実際の文書中では，スペルミスや表記ゆれなどの影響により「わずかに異なる同じキーワード」が多く存在する．また WIX ファイルのキーワードにも作成者が異なる場合など，表記ゆれが入り込む可能性がある．表記ゆれには表 2 に示すようなパターンが存在する．

表 2　表記ゆれの例

| パターン | 例 |
|---|---|
| 漢字 | 國／国<br>沈殿／沈澱 |
| カタカナ語 | ダイアモンド／ダイヤモンド<br>プリンタ／プリンター<br>バイオリン／ヴァイオリン |
| 送り仮名 | 受付／受け付け<br>売場／売り場<br>付く／つく<br>引っ越／引越／引っ越し／引越し |

また，人物名などは「鈴木選手」「麻生首相」「〜氏」「〜様」のようにフルネームが存在せず，名字やそれに伴う敬称までしか文中に存在しない場合も多い．事前にそれらすべてをエントリのキーワードとして加えておくことで解決することも可能ではあるが，許容する候補を選出しそれらすべてをエントリのキーワードとして追加するという負担すべてを WIX 作成者が担うのでは，とても現実的とは言えない．

したがって，微妙な違いをシステムで吸収してヒットとみなすことが必要である．しかし扱う対象によって吸収の程度が大きく異なり，たとえば製品 ID などは逆に吸収してはいけない場合も存在する．本研究では人物名に対する WIX ファイルを扱うこととし，姓ないし名だけでもヒットとみなす程度に制約を緩めることとし，これ以上深い議論は行わない．

あいまい吸収は，本質的にマルチヒット問題を生じやすくするものである．そのため本研究では，マルチヒット問題に対する解決手法について重きを置く．

## 4.　システムの実装

### 4.1　概　　要

クライアント側は，Mozilla Firefox アドオンとして実装する．インストールすると次のような動作をする．Web 文書の読み込み時に，当研究室のサーバにある HTML 文書をインラインフレームとして埋め込む．コンテクスト・メニュー，ツールバー・メニューにフルアタッチおよびユーザ指定アタッチの項目が追加され，アタッチが実行できるようになる．

サーバ側の実装は Java 言語を用い，サーバは Apache Tomcat 6.0 を使用した．事前に作成した WIX ファイルから，まず

次の 2 つのインデックスを作成する．アタッチではこれらの辞書を用いてヒット判定やランキング処理を行う．

- あいまい検索用 N グラム・インデックス

　各 WIX ファイルがもつ各エントリのキーワードから作成する．

- スコアリング用インデックス

　各エントリが指す Web 文書を取得・解析して，スコアを決定する．

本研究におけるアタッチはユーザ指定アタッチが基本である．フルアタッチの場合にはヒットしたキーワードをマーキングし，クリックするとそのキーワードを選択したものとしてユーザー指定アタッチを実行する仕様である．

### 4.2 クライアントサイドの実装

#### 4.2.1 xul

XUL（読み：ズール）は Mozilla Firefox や Mozilla Thunderbird などの Mozilla アプリケーションを作成するためのユーザインタフェース・マークアップ言語である [6]．XUL を用いて，デフォルトのメニューにフルアタッチ用の「full attach」およびユーザ指定アタッチ用の「user attach」の項目を追加し，ショートカットキーをそれぞれ (f), (u) に設定した．そして，その項目が選択されたときに実行するロジックを JavaScript で関数として定義する．

#### 4.2.2 JavaScript : userattach()

ユーザ指定アタッチを実行した時に呼び出される関数である．

選択されたキーワードは *content.getSelection*() によって取得することが可能である．続いて，周辺文字列を取得するが，本研究の評価のため次の 3 種類を周辺文字列として検証する．

（1） 選択されたノードから直近のブロック要素以下の全テキスト

（2） 直近のブロック要素と同種のブロック要素以下の全テキスト

（3） HTML タグ以下の全テキスト

ただし，ブロック要素は HTML におけるタグ要素であり，表 3 のように定義した．

表 3　ブロック要素

| | | | | |
|---|---|---|---|---|
| BODY | P | DIV | TABLE | TBODY |
| TR | TD | UL | OL | LI |
| FORM | H1 | H2 | H3 | H4 |
| H5 | H6 | CENTER | PRE | BLOCKQUOTE |

これらのアタッチ情報（キーワードと周辺文字列）をサーバ側に送信するには，Ajax 通信を利用する．しかしアドオンから当研究室のサーバと通信するためには，クロスドメイン Ajax 通信が必要となる．そのため，アクセスした Web ページの中に当研究室のサーバにある HTML ファイルをインラインフレームを埋め込み，そのフレームを介して *postMessage*() メソッドを利用することでクロスドメイン Ajax 通信を実現した．

ランキングされた結果は，サーバ側にある HTML ファイルを動的に変更することによって，ユーザが閲覧しているインラインフレームの内容が変化し，ユーザに提示することができる．

#### 4.2.3 JavaScript : fullattach()

フルアタッチを実行した時に呼び出される関数である．

この関数では，フルアタッチ用に読み込ませたユーザからは見えない隠しインラインフレームを介して，閲覧中のソースをサーバに送信する．サーバではそのソースを全文解析して，WIX ファイルにあるキーワードがあればそのキーワードを次のように書き換える．

キーワード
↓
<a style=”省略” href=”Attach”
onclick=”atag_attach(this);return false;”>
キーワード < /a>

Web 文書のソース中にあるすべてのキーワードを書き換えたら，インラインフレームから閲覧しているページにそのソースを上書きする．これによって，ユーザが閲覧している文書において WIX ファイルとアタッチすることが可能なキーワードすべてにリンクが生成される．このリンクをクリックすると，そのキーワードを選択したものと解釈してユーザ指定アタッチが実行される．

### 4.3 サーバサイドの実装

#### 4.3.1 N グラム・インデックス

本研究では 1-gram によるあいまい検索を採用した．この他には 2-gram，3-gram や編集距離を用いたあいまい検索も考えられるが，人名に限定した場合には 2-gram や 3-gram では制限が厳しすぎるため，柔軟な対応が出来ない．また編集距離に関しては本研究では試していないが，対象によっては効果的であると考えられる．

具体的な 1-gram インデックスの例を表 4 に示す．

表 4　1-gram インデックスの例

| letter | kw |
|---|---|
| 鈴 | 鈴木亜美 |
| 木 | 鈴木亜美 |
| 亜 | 鈴木亜美 |
| 美 | 鈴木亜美 |

このような索引を，用意した WIX ファイルにある全キーワードから作成する．

#### 4.3.2 あいまい検索

あいまい検索の実装はすべて $KeywordSearch$ インターフェースの実装クラスとして与える．本稿では $UnigramSearch$ クラスがそれにあたる．

ヒット判定は次の定義に従う．アタッチの際にユーザが選択したキーワードの文字列の 1-gram 集合を $G_1$ とする．このとき，次の条件を同時に満たすキーワードをヒットとみなす．

- $\frac{|G_1|}{2}$ 以上のグラムを共有している
- キーワード長が $|G_1| * 2$ 以下

例えば，サッカー選手「中村俊輔」は「中村」や「中村選手」でもヒットとなる．

#### 4.3.3 スコアリング用インデックス

WIX ファイルのエントリが指す Web 文書を形態素解析エンジンによって解析し，抽出した単語についてインデックスを作成する．WIX ファイルのエントリにより指定される Web 文書から単語を抽出し，次の基準を満たした単語だけを扱う．

- HTML ソースからタグ部分をすべて消去し，テキスト部分を抽出する
- そのテキストから抽出された単語から，不要な要素を除去する

すべての WIX ファイル，すべてのエントリについて Web 文書の解析を行い，まず次のデータを収集する．WIX ファイル，エントリ（本稿では属性に id をもたせて特定する），単語，その Web 文書での出現回数 ($ltf$) およびそれを正規化した $nltf$ をデータベースに格納する．ここで，データベースにおいてスコアリング用インデックスのテーブルがもつ属性を表 5 に示す．

表 5　スコアリング用辞書テーブルの属性

| | | |
|---|---|---|
| file | WIX ファイル名 | WIX ファイルを特定する |
| eid | エントリ ID | WIX ファイル内での ID |
| kw | キーワード | 単語 |
| ltf | local tf | ローカルな単語出現頻度 |
| nltf | normalized local tf | 正規化したローカルな単語出現頻度 |
| ltfidf | local tf*idf | ローカルな TF*IDF 値 |
| gtf | global tf | グローバルな単語出現頻度 |
| ngtf | normalized global tf | 正規化したグローバルな単語出現頻度 |
| gtfidf | global tf*idf | グローバルな TF*IDF 値 |

正規化は注目する単語の出現回数を，その文書から抽出された単語総数で割ることで行う．これによって，文書長における差異をなくすことができる [11]．

一つの WIX ファイルについてデータを収集し終えるごとに ltfidf を計算する．TF*IDF はある文書に出現した単語が，WIX ファイルで扱っている文書集合の中でどれだけの価値があるかを表す．1 つの WIX ファイル中で，その単語どれだけの価値があるかを表すのが ltfidf である．

この ltfidf から計算できるスコアはエントリのスコアであり，WIX ファイルそのもののスコアではない．WIX ファイルは通常，内容に一定の方向性をもって作成されるものであるから，それをスコアリングすることには意味がある．よって，WIX ファイル単位での TF*IDF，すなわち gtfidf を計算する．

WIX ファイル単位で各単語の ltf の和を取り，それを gtf とする．さらに注目 WIX ファイル中の ltf の総和で正規化する ($ngtf$)．そして gtfidf を求める．

#### 4.3.4 ランキング

アタッチ情報から得られた周辺文字列に対して，Sen を用いて Web 文書の解析と同様に単語を抽出することにより単語集合を取得する．ここで，先の辞書から次の 2 つのスコアを計算する．

| | |
|---|---|
| $SCORE(wix)$ | WIX ファイルのスコア |
| $SCORE(entry)$ | エントリのスコア |

最終的なエントリのスコアは，各エントリのスコアはそれらの積として求める．このスコアを降順に並べた結果がランキング結果である．

ここで，これら 2 つのスコアについて 3 つの定義を検証する．

| | |
|---|---|
| Simple | アタッチ情報と共有する単語の数をスコアとする定義 |
| TF | 共有単語の出現数の和をスコアとする定義 |
| TF*IDF | 共有単語の TF*IDF 値の和をスコアとする定義 |

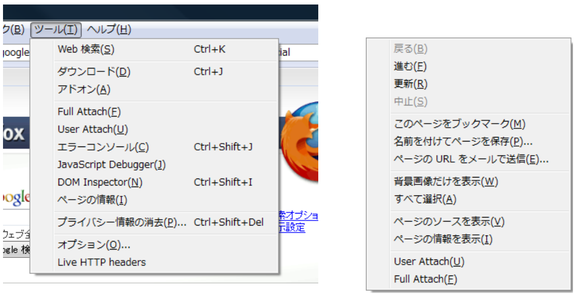
すなわち，$SCORE(wix)$ は，共有単語の個数または global tf の和，または global tfidf の和により計算する．$SCORE(entry)$ は，共有単語の個数または local tf の和，または local tfidf の和により計算する．

ここに示したスコアリング定義の組合せにより，9 種類のランキング結果が得られる．さらに，3 種類の周辺文字列の範囲定義によって得られるランキング結果が異なるため，計 27 種類のランキング結果を検証する．

### 4.4 実　行　例

#### 4.4.1 ユーザインターフェース

アタッチの実行はツールバー・メニューまたはコンテクスト・メニューに追加された項目から行う．その様子が図 4.4.1(a)(b) である．ショートカットキーを定義してあるので，そのキーも表示されている．




(a) ツールバー・メニュー　　(b) コンテクスト・メニュー

#### 4.4.2 ユーザ指定アタッチ

ユーザ指定アタッチの様子を図示する．図 5 がユーザ指定アタッチを行った様子である．図中では Web 文書中にある「イチロー」という文字列を選択し，コンテクスト・メニューからユーザ指定アタッチを実行したところである．実行すると，その周辺情報を考慮したランキング結果が画面下方のインラインフレームに表示される．

#### 4.4.3 フルアタッチ

フルアタッチの実行前後の画面を図 6 および図 7 に示す．フルアタッチの実行もツールバー・メニューあるいはコンテクスト・メニューから簡単に実行できる．実行すると，閲覧中の Web 文書と WIX ファイルが結合され，それらのキーワードがハイパーリンクに変換される．画面では，変換された部分を赤い背景の太字にして強調表示している．

そして，そのハイパーリンクをクリックすることにより，そのキーワードを選択したものとしてユーザ指定アタッチが実行される．

図 5　ユーザ指定アタッチ

図 6　フルアタッチ実行前

図 7　フルアタッチ実行後

## 5. 評　　価

### 5.1 評価方法

ユーザ指定アタッチを行うことにより，そのアタッチ情報に応じたランキング結果を得ることができる．周辺文字列の定義とスコア定義の組合せによって 27 種類のランキング結果が生成される．

アタッチ情報として，

- キーワード
  ユーザ指定アタッチのために指定された，選択文字列．
- 周辺文字列
  選択文字列の周辺文字列であり，定義により 3 種類ある．
- 文脈が意味する人物名
  評価のために，文脈が意味する人物をユーザに選択してもらう（図 8）．

を保存しておき，それぞれについて 27 種のランキング結果すべてについて精度を求める．

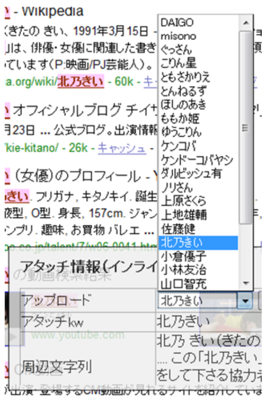


図 8　文脈が意味する人物の選択

各アタッチ情報に対応するランキング結果の精度は次のように評価する．アタッチの結合キーとなるキーワードを $kw$ とし，アタッチ後のランキング結果を $Ranking(kw)$ とする．また，アタッチされた文脈が意味する人物を表すエントリ集合を $Target(kw)$ とする．文脈中のキーワード（人物）を表すエントリは，人物ごとに事前に定義しておきそれを用いて取得する．

ここで，ランキング結果には文脈にそぐわない人物も同時にヒットしているとすると

$$|Target(kw)| < |Ranking(kw)|$$

が成立する．ここで | エントリ集合 | は集合の要素数を表す．

よって，ランキング結果の上位 $|Target(kw)|$ 件の中に，どの程度 $Target(kw)$ の要素が含まれるかを精度として求めることができる．すなわち，

$$\text{精度} = \frac{|Target(kw) \cap Ranking(kw)|}{|Target(kw)|} \times 100(\%)$$

である．

評価には「鈴木一朗」「鈴木亜美」「佐藤琢磨」の 3 名を取り上げ，アタッチは名字だけで実行した．「鈴木」「佐藤」の名前をもつ人物は多いため，マルチヒットが起こりやすいためである．本実験で利用した WIX ファイルでは，「鈴木」では 30 件，「佐藤」では 8 件のマルチヒットが生じる．各人物に対するアタッチはそれぞれ 15 回行い，各評価手法の評価結果の平均と分散を求めた．以下ではこの結果を用いて，比較検討を行っていく．

### 5.2 結果および考察

グラフは 3 名それぞれの評価結果と，全体における評価結果を示した．各グラフの縦軸は精度 (%)，横軸はスコアリング手法の組合せを表し，「$SCORE(wix)$ の定義 * $SCORE(entry)$ の定義」となっている．また各手法にある 1, 2, 3 の項目は周辺文字列の定義を表しており，4 章で示した数字に対応する．

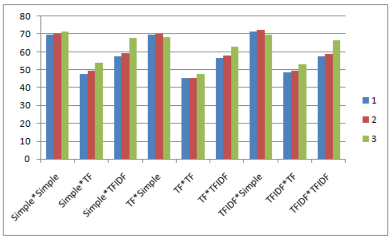



図 9　全体の評価平均

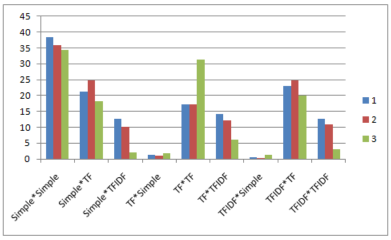



図 10　全体の評価分散

#### 5.2.1　全　　体

3 名に関する全アタッチ情報の評価結果のグラフを図 9 と図 10 に示す．平均値を見ると高くても 70%となっており，低いものは 50%を割るほどとなっている．比較的精度がよいと思われるものは，$Simple \times Simple$，$TF \times Simple$，$TFIDF \times Simple$ である．しかし，$Simple \times Simple$ は分散が大きく精度が極端に悪くなる場合が存在している．

周辺文字列による情報が増加すると判断要素が増加する．そのため基本的には精度は上昇することが予想でき，図 9 の評価平均から，その傾向は実際に確かめられる．しかし一方で，判断を誤らせるノイズも増加すると考えられるが，図 10 を見ると，むしろノイズ除去に働いていることが分かる．つまり，どのようなアタッチ情報に対しても安定した精度を示し，バラツキが小さくなるということである．

$TF \times Simple$，$TFIDF \times Simple$ に関しては分散の傾向は明確には見られない．よって，最も高い精度を示していること，周辺文字列の定義は 1 が最も処理負荷が小さいことから，この 2 つの定義手法が最適であると判断できる．

#### 5.2.2　ま　と　め

- $TFIDF \times Simple$ が全体を通して最もよい結果を示す．
- $SCORE(entry)$ の定義には $Simple$ を採用する方が高い精度を示す結果となる．
- $SCORE(wix)$ の定義に $TF$，$TFIDF$ のいずれかが最も精度が高く，分散が小さくなる．
- 周辺文字列の定義は 1 を利用すれば，他と同程度の結果が期待できる．

## 6.　結論、今後の課題

### 6.1　結　　論

クライアント側とサーバ側で処理を連携することにより，ユーザは容易に WIX システムを導入でき，かつ将来サーバ側で提供されるであろうさまざまなサービスを享受することができるようになった．また，あいまい一致を実現することによってより柔軟なアタッチが可能となった．さらに，マルチヒットしたとき文脈に応じた順に提示することによって，アタッチする WIX ファイルがさまざまな範囲を扱う場合でも，文脈に即したエントリを優先してユーザに提供することが可能となった．

ランキング手法としては，周辺文字列はアタッチされた位置から直近のブロック要素に含まれるテキストを用いれば十分であり，スコアリングは $TFIDF \times Simple$ が最適であることが分かった．

### 6.2　今後の課題

以下に挙げる課題を解決することにより，WIX システムを一般ユーザが触れることが可能になる．いずれも実現可能であると思われ，WIX システムの有用性の向上につながると考えられる．ぜひ今後取り組んでいきたい課題である．

- 公開された WIX ファイルの利用
- ランキング手法の改良
- あいまい検索の研究
- Web 文書の解析

## 文　　献

[1] mixi，http://mixi.jp/
[2] Ameba Blog，http://www.ameba.jp/
[3] Wikipedia，http://ja.wikipedia.org/wiki/メインページ
[4] はてなダイアリー，http://d.hatena.ne.jp/
[5] 形態素解析システム Sen，http://www.mlab.im.dendai.ac.jp/ yamada/ir/MorphologicalAnalyzer/Se
[6] XUL チュートリアル，https://developer.mozilla.org/ja/XUL
[7] 高橋　健太郎，“ SuperSQL による A-doc ファイルの生成 ”，データ工学ワークショップ，DEWS2007，2007
[8] 佐藤　裕紀，“ A-doc ファイルのアタッチの機能を持つ専用ブラウザの試作 ”，データ工学ワークショップ，DEWS2007，2007
[9] 市東　隼，“ A-doc に基づく WEB リンク集再利用化ツールの試作 ”，データ工学ワークショップ，DEWS2008，2008
[10] 朱　成敏，遠山　元道，“ A-doc に基づく閲覧者主導のハイパーリンク支援 ”，データ工学ワークショップ，DEWS2008，2008
[11] G. Li, B. C. Ooi, J. Feng, J. Wang, L. Zhou: “EASE: An Effective 3-in-1 Keyword Search Method for Unstructured, Semi-structured and Structured Data”, *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pp. 903-914, 2008