人工知能学会 インタラクティブ
情報アクセスと可視化マイニング研究会(第10回)
SIG-AM-10-03

SOM を利用した Exploratory Search のためのユーザインタフェース開発

Development of the user interface for Exploratory Search using the SOM

徳永 秀和[1]　井上 雄翔[1]

Tokunaga Hidekazu[1] and Inoue Yusho[1]

[1] 香川高等専門学校

[1] National Institute of Technology, Kgawa College

The important thing in Exploratory Search is that a retrieving person clarifies the goal of search. For that purpose, first it is required to find the keyword which related to Search-word. Then, a retrieving person finds the related keyword that he is interested in. However,since the information acquired by search is huge, it is difficult to find the keyword which fulfills conditions from the information. Then, I thought that such a problem was solvable by developing the tool which extracts only required information from search results and displays the clustered result. In order to make a clustering result intelligible visually, a selforganization map is used, and information is arranged and displayed on a two-dimensional map. Moreover, in order to be able to reflect a user's idea in a clustering result, it enables it to change freely the parameter of the feature vector used by SOM. Finally, evaluating the usefulness of this tool by experiment.

１．はじめに

近年の高度情報化にともなってインターネット上の Web ページは急激に増加しており，現在は 1 兆ページを超えるといわれている[1]。この膨大な Web ページの中から必要な情報を得るために，検索の手法は多様化している。なかでも注目されている検索手法が Exploratory Search である。

Exploratory Search とは，情報のニーズが明確でない検索者が，検索で得られる情報を基に検索の目標を明確化しながら，新しい知識を獲得していく検索手法である[2]。検索の目標を明確化するときに重要となるのが，検索語と関連するキーワードである。検索で得られた情報の中から検索者が興味のあるキーワードを見つけ，そのキーワードを基に検索を繰り返すことが目標の明確化につながる。

インターネット検索を行う際の Web ページ滞在の調査によると，検索者が１ページに滞在する平均時間は約１分といわれている[3]。１ページあたりにかかる閲覧時間はそう長くないが，情報ニーズがあいまいで，検索キーワードに対する予備知識の少ない検索者が１ページずつ情報を探索していくと，検索に長い時間を要してしまう。さらに前述したように Web ページの数は膨大であるため，多くの情報の中から検索者にとって本当に有用なキーワードや Web ページを見つけるのは困難であると予想される。したがって，検索情報の中から必要な情報を抽出し，分類して検索者に提示するツールが必要であると考えられる。

そこで本研究では，Web ページから必要な情報を抽出して，それらをクラスタリングして表示することで，Exploratory Search の支援を行う GUI システムを開発することを目標とした。

２．目標達成の手段

Exploratory Search において検索目標を明確化するとき重要となるのが，検索キーワードに関連し，検索者の興味を引くキーワードを見つけることである。本システムでは検索者にそのようなキーワードを見つけやすくすることで，Exploratory Search を支援する。

検索者が特定のキーワードを見つけるためには，まず Web ページ内の情報を絞り込むことが必要であると考えられる。そこで本システムでは Web ページ中の名詞に注目し，それらを検索者の興味を引くキーワードの候補として抽出して，クラスタリングする。また，検索者によって興味を引くキーワードは異なるため，システムが独自に設定するパラメー

タによるクラスタリングの結果が必ずしも興味を引くキーワードの特定につながるとは限らない。そこで自己組織化マップと GUI を組み合わせ，検索者が独自の判断でパラメータを変更してクラスタリングを行うことでキーワードを絞り込むことのできるシステムを開発する[4]。

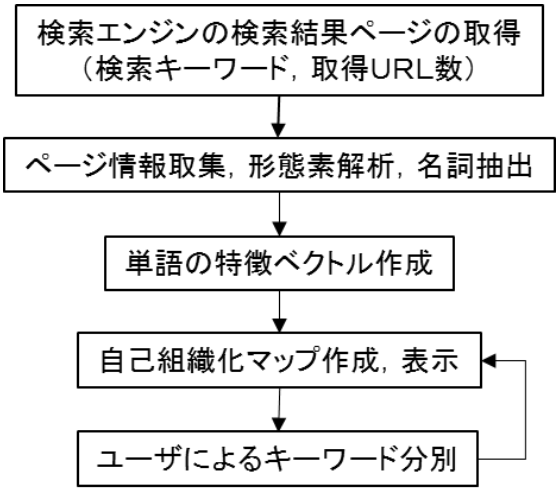

図１　システムの処理の流れ

３．システム構成

3.1 処理の流れ

システムの処理の流れを図１に示す。①キーワードと取得するホームページ数を指定し，検索エンジンより検索結果の HTML 文書を取得する。②HTML 文書より必要なテキスト情報を抽出し，形態素解析し，必要な名詞情報を抽出する。③抽出した名詞（キーワード）の特徴ベクトルを作成する。④キーワードの特徴ベクトルより自己組織化マップを作成し，表示する。⑤ユーザが自己組織化マップのノードを操作し，キーワードを選別する。⑥選別情報を基に，再び自己組織化マップを作成，表示する。⑦これ以降，⑤，⑥を繰り返し，興味を持つキーワードを探索する。

3.2 クラス構成

クラスの構成を図２に示す．SOMtest クラスで全体の流れを制御する。MakePagedata クラスにより，検索エンジンからの HTML 文書取得と名詞データの管理を行う。HTML 文書取得には Httpclient.jar，HTML 文書の処理には jericho-html.jar を使用する。形態素解析は jgo.jar を使用する。自己組織化マップの処理は，ExecSOM クラスが JRI.jar を使用し R の som ライブラリを利用する。SOMgui クラスによりキーワード選別と再自己組織化マップ作成を行う。

3.3 検索結果の取得

システムを開始すると検索キーワード，検索ページ数，また Web ページ本文とスニペットのどちらを

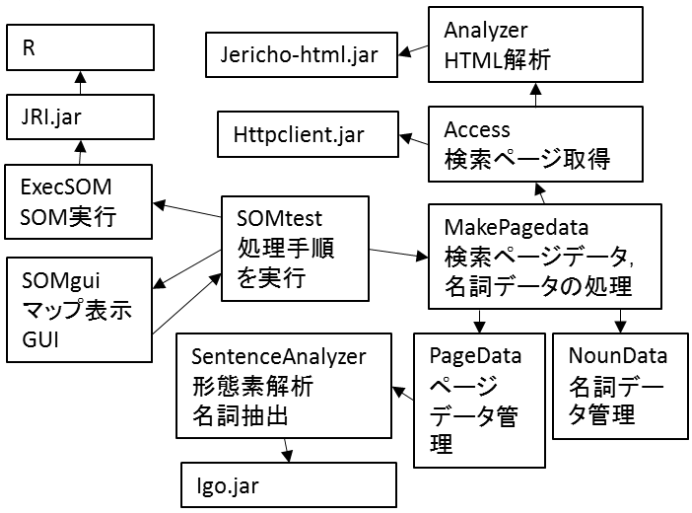

図２　クラス構成

使用するかを入力する画面が表示される。それぞれのデータを入力して実行ボタンをクリックすると，Google 検索エンジンから検索結果の HTML 文書を取得する。本システムでは Google 検索エンジンから検索結果を取得する際に使用する，HTTP ユーザーエージェントというパラメータを固定している。これにより，システム実行環境に依存せず検索結果を得ることができる。

3.4 形態素解析と名詞抽出

検索エンジンから得た検索結果を形態素解析し，形態素の中から名詞のみを，検索キーワードとの関連キーワードとして抽出する。形態素の品詞は階層構造で分類されており，単に名詞といっても数十種類に細かく分類される。本システムでは名詞の中でも特に単独で強い意味を持つことの多い「名詞，一般」と「名詞，固有名詞」を主として抽出する。また「ノンアルコール」などのように，単語として意味を成すが，「ノン」と「アルコール」という複数の形態素に分解されるような単語については，「ノンアルコール」というように一つの単語を関連キーワードとして抽出する。「環境汚染問題」のように複数の名詞が連続する複合名詞は，複合名詞をキーワードとする。

3.5 特徴ベクトル

特徴ベクトルとは関連キーワードの特徴を数値化して並べた多次元ベクトルのことである。本システムでは抽出した全ての関連キーワードについての特徴ベクトルを作成する。特徴ベクトルの属性は，「検索結果全体での出現回数」，「固有名詞であるか否か」，「キーワードの文字数」，「Web ページ１での出現回数」，・・・「Web ページ n での出現回数」である。

3.6 自己組織化マップ

自己組織化マップとは，与えられた特徴ベクトルからそれぞれのキーワードの類似度をマップ上の距離で表現するものである。自己組織化マップ上では類似度の高いキーワードどうしは近くに，類似度の低いキーワードどうしは遠くに配置される。多次元データを持った関連キーワードを 2 次元マップ上に視覚的にわかりやすく表示できるため，多数の関連キーワードを分類し表示する必要のある本システムに適していると考えられる[5]。

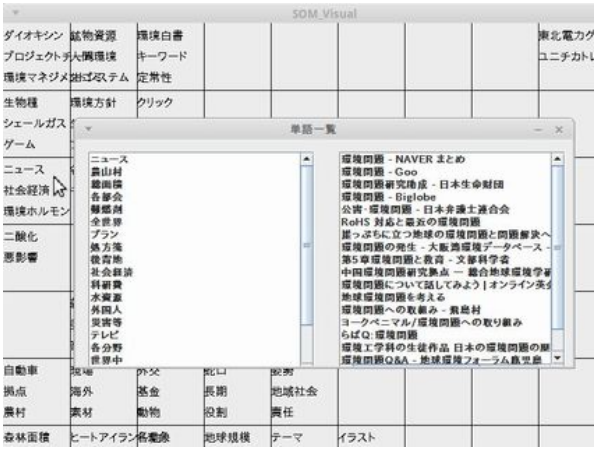

図３　自己組織化マップ

自己組織化マップは統計解析ソフト R によって作成する。本システムで作成・表示される自己組織化マップは，クラスタ数が 10×10，クラスタ形状が四角形のものである。自己組織化マップの作成と同時に，各関連キーワードの重要度の計算が行われる。関連キーワードの重要度は以下の式によって計算される。

重要度 ＝ A＊出現回数＋B＊文字数 ＋ C＊固有名詞
（固有名詞は，固有名詞なら１，違えば 0）

ここで A,B,C,は関連キーワードの各属性の係数であり，ユーザーが独自に設定できる値である。

自己組織化マップの作成と関連キーワード重要度の計算が終わると，図３の画面（単語一覧のポップアップは除く）が表示される。各ノードのマスごとで関連キーワード重要度の高いキーワードが最大３つまで表示される。また各ノードのマスを左クリックすると，図３（単語一覧のポップアップ）ようにクリックしたノード内の全ての関連キーワードと，それらのキーワードを含む Web ページのタイトル一覧を表示した画面が現れる。画面内左側にリスト表示された関連キーワードをクリックで選択すると，選択した関連キーワードが含まれる Web ページのみが右のリストに表示される。このとき関連キーワードは複数同時に選択することができる。

3.7 ノード選択と再マップ表示

自己組織化マップパネルを用いたクラスタリングでは，10×10 の各ノードに必要・普通・不要のいずれかの属性を割り当てて分類する。ノードに含まれる全ての関連キーワードは，ノードと同じ属性が割り当てられる。各ノードを右クリックすると属性を設定するためにポップアップメニューが現れ，メニューの中から属性を選択することでノードに属性を割り当てることができる。ノードに属性を割り当てると，ノードの背景色が必要属性ならば赤色に，不要属性なら青色に，普通属性なら灰色（元の色）に変化する。各ノードに属性を割り当てた時の例を図４に示す。

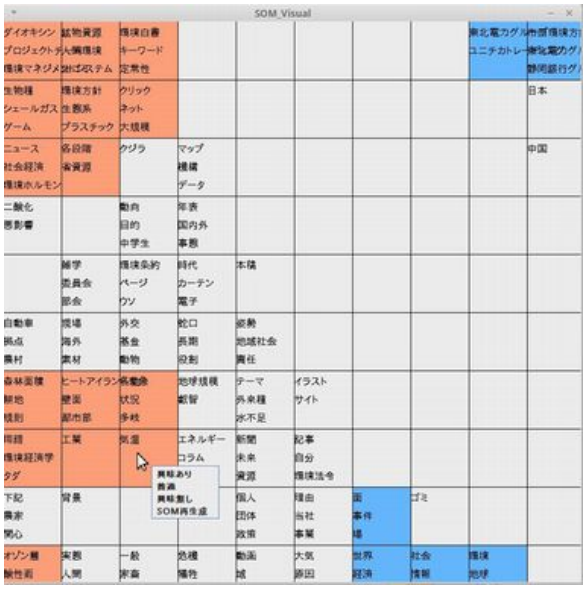

図４　ノードへの属性設定

関連キーワードの分類が完了したら最後に自己組織化マップの再作成を行う。再作成は右クリックで現れるポップアップメニューの最下部にある「再作成」を選択することで実行される。再作成が実行されると不要属性が割り当てられた関連キーワードが削除され，必要・普通属性の関連キーワードだけで再度特徴ベクトルが作られる。このとき特徴ベクトルに新たな属性が 1 つ追加される。追加された属性の値は，必要属性であれば 300，普通属性であれば 0 となる。新たな特徴ベクトルを基に再び自己組織化マップを作成して表示する。このように自己組織化マップの作成とユーザーによる関連キーワードの選別を繰り返し行うことで，ユーザーが興味を持つ関連キーワードや Web ページを絞り込んでいくことができる。

４．実験と考察

4.1 実験方法

本システムにおけるユーザーの関連キーワードの選別から自己組織化マップの再作成・表示までを 1 サイクルと考えたとき，1 サイクルで検索結果から抽出した関連キーワードと，それらを含む Web ページの数をどの程度絞り込んでいくことができるかを調べる。これによりシステムの ExploratorySearch 支援の性能を評価する。実験の条件として，検索数を 100 ページ，使用する Web ページの情報をスニペットとする。以下の表 1 に実験時の検索キーワードと，検索結果の中で必要とした話題の基準，関連キーワードの例を示す。

表 1 実験の ExploratorySearch

検索 キーワード	必要な 話題	必要な キーワード例
環境問題	環境問題の種類	地球温暖化
宇宙	航空技術	ロケット
プログラミング	言語の種類	java
国内旅行	観光地	北海道
海外旅行	観光地	台湾

4.2 実験結果

表 1 に示す検索キーワードで実験を行ったときの，初期の検索結果のキーワード数と Web ページ数を表 2 に示す。ここで，Web ページの数が 100 以下のものがあるが，これは抽出するべき関連キーワードをスニペット中に含まない Web ページが存在したためである。

表 2　初期の数

検索キーワード	キーワード数	Web ページ数
環境問題	482	100
宇宙	585	99
プログラミング	410	100
国内旅行	443	100
海外旅行	436	99
合計	2356	498

1 サイクル，2 サイクル後のキーワード減少率とページ減少率を表 3，表 4 に示す。 キーワード減少率とページ減少率は，以下の式で定義した。

$$\text{キーワード減少率} = \left(1 - \frac{\textit{現在キーワード数}}{\textit{初期キーワード数}}\right) \times 100$$

$$\text{ページ減少率} = \left(1 - \frac{\textit{現在ページ数}}{\textit{初期ページ数}}\right) \times 100$$

表 3　1 サイクル後の減少率

検索 キーワード	キーワード 減少率	ページ 減少率
環境問題	21	0
宇宙	41	2
プログラミング	29	3
国内旅行	87	40
海外旅行	58	7
平均	49	11

表 4　2 サイクル後の減少率

検索 キーワード	キーワード 減少率	ページ 減少率
環境問題	56	11
宇宙	69	13
プログラミング	55	23
国内旅行	90	43
海外旅行	75	43
平均	69	25

4.3 考察

表 3 を見ると 1 サイクル後の自己組織化マップでキーワード減少率の平均は 49%である。1 サイクル目のキーワード選別では必要・普通属性に割り当てられたノード数の合計が 116，不要属性に割り当てられたノード数の合計が 142 とおおよそ同数である。表 4 を見ると，2 サイクル後の自己組織化マップではキーワード減少率の平均が 69%と 1 サイクル後からさらに半減近く減少している。2 サイクル目のキーワード選別では必要・普通属性に割り当てられたノード数の合計が 54，不要属性に割り当てられたノード数の合計が 39 であり，こちらもおおよそ同数である。自己組織化マップの再作成では，不要属性の関連キーワードが削除される。表 3，表 4 の関連キーワードの減少率の平均と，不要属性に割り当てられ削除されたノードの比率がほぼ同じであることから，自己組織化マップの各ノードには関連キーワードがほぼ均等に配置されており，関連キーワードの減少数は不要属性に割り当てるノードの数に比例すると考えられる。したがって，本システムは関連キーワードの絞り込みについては非常に効率よく行うことができると考えられる。

表 3 を見ると 1 サイクル後の Web ページの減少率の平均は 11%であり，キーワードの減少率と比較すると非常に低いことが分かる。また減少率の平均が 11%となったのは検索キーワード「国内旅行」で特

に減少率が多かったためであり，本来の１サイクル後の Web ページ減少率は 11%よりも低い数字だと予想される。表４を見ると２サイクル後でも初期表示からの Web ページ減少率は 25%であり，キーワードの初期表示からの減少率 69%と比較しても低いことが分かる。したがって本システムではキーワード選別によるクラスタリングのサイクルを繰り返しても，キーワードを含む Web ページを絞り込むのは難しいと考えられる。このような結果となった原因は，キーワードに一般的に使用される単語が含まれてしまったことや，必要とした話題の関連キーワードが，検索キーワードを説明する際に一般的に使われるキーワードであったためなどが考えられる。本システムが効率的に Web ページを絞り込めるようにするには，一般的に使用されるキーワードを削除することや，必要とする話題を検索キーワードの中でもマイナーなものにする必要があると考えられる。

５．おわりに

本研究では，Exploratory Search を支援するために検索結果のからの関連キーワード抽出と，ユーザによるキーワード選別およびクラスタリング機能を備えた提案システムの開発，および提案システムの性能を検証するための評価実験を行った。実験の結果から，本システムは検索キーワードに関連した特定の話題のキーワードの絞り込みについては効率よく行うことができるが，それらの話題について詳しく調べるために閲覧する，関連キーワードを含む Web ページの数はクラスタリングを繰り返し行っても大きく減少することはなく，効率よく絞り込むのは難しいことが分かった。Web ページを効率よく絞り込んでいくには，関連キーワードのクラスタリング方法や，必要とする話題の選択を工夫する必要がある。

本研究の今後の課題としては，それぞれ異なるユーザーにとって最適な検索およびクラスタリング結果を得られるようにするために，システムの設定をユーザーが自由に変更できるようにすることが挙げられる。本システムはプログラム側がユーザーに一方的にクラスタリングの結果を提供するのではなく，ユーザー側が主体となって独自の基準でクラスタリングを行うことで Exploratory Search を進めることを目標としている。今回説明したシステムの内容ではユーザーが決定できるのは，検索キーワード，検索ページ数，解析に使用する情報の種類のみであり，それ以外の設定は変更することができない。しかし本来は，特徴ベクトルの属性内容や値，キーワードの重要度の決定方法や計算式の係数の重み，抽出する関連キーワードの品詞の種類，自己組織化マップパネルのノード数など多くの項目をユーザーが自由に変更することで，ユーザー独自のクラスタリングを提供するシステムが理想である。今後はこのような課題を解決するためにシステムの改良を行う必要がある。

参考文献

[１] Jesse Alpert，Nissan Hajaj：We knew web was big…OfficialGoogleBlog-http://www.googleblog.blogspot.jp/2008/07/weknew-web-was-big.html.

[２] RyenW.White,ResaA.Roth:ExploratorySearch:BeyondtheQuery-ResponseParadigm-http://www.morganclaypool.com/doi/abs/10.2200/s00174ed1v200901icr003.

[３] JAKOB NIELSEN:How Long Do User Stay Web Pages?NielsenNormanBlog-http://www.nngroup.com/articles/howlong-do-users-stay-on-web-pages/.

[４] 梶並知記，高間康史：ユーザ意図を強調したキーワード配置支援機能を備えたインタラクティブなキーワードマップ，情報処理学会論文誌，Vol.48，No3，pp.1176-1185，2007.

[５] 津高新一郎：自己組織化マップを用いたテキスト自動分類の試み，情報処理学会 第 46 回全国大会講演論文集，pp.187-188，1993.