修 士 論 文

「情報地図」による情報獲得の支援に関する研究

指導教官　落水 浩一郎 教授

北陸先端科学技術大学院大学
情報科学研究科情報システム学専攻

川瀬 宏一郎

1998 年 2 月 13 日

要 旨

本研究では、既存の協調的情報フィルタリングシステムの問題点を概括し、さらにコンピュータ・ネットワーク環境のユーザがどのような形で情報獲得活動を行っているのかを考察した上で、ユーザにとって新しくかつ有用な情報の獲得を支援するためのシステムを提案する。そのシステムの設計方針は、個人が保有する情報体系を「情報地図」と呼ぶグラフ形式にて視覚化し、各ユーザの情報地図をシステムが合成することにより、他ユーザの知識を参照可能にすることである。そして、この設計方針に基づき、プロトタイプシステムの実装を行い、評価実験によってシステムの効果を確認する。

目 次

図 目 次

表 目 次

第 1 章

はじめに

1.1 本研究の背景と目的

コンピュータ・ネットワークの著しい発達に伴い、コンピュータ環境下での知的生産活動においては、ネットワークを通じて多種・大量の情報を手にすることが可能になった。例えば、本学の環境では 9,000 あまり[1]の Netnews のニュースグループを購読することが可能であるし、WWW で展開されている検索エンジンの goo [2] では、5,500 万件[3]のドキュメントをインデックス化しているという。しかし一方では、その情報の多種・大量性は爆発的に増大しているため、もはや個人がそれらの情報すべてを処理するのは、事実上不可能になってしまっている状況でもある。知的生産活動における情報の獲得は、ある最終的な目的 — 例えば本論文の作成 — へ至るための、過渡的ではあるが不可欠な作業であると言える。したがって、生産活動をより快適なものにしようとするならば、ネットワーク上の情報源から、自分にとって有用であると思われる情報を優先的に選択・発見する何らかの手段を講じる必要が生じてくる。

この要求に対応するために、情報検索と情報フィルタリングが研究され用いられてきた。実際に、現在の情報獲得活動では、サーチエンジン、キーワードマッチング方式のフィルタリング、学習型エージェントなどが具体的な方法として広く利用されている。しかし一方では、これらを単純に用いる場合は、ユーザ個人の知識・価値観に依存するとこ

[1] 1998 年 1 月の時点

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

[3] 1998 年 1 月の時点

ろが大であり、そのことによる問題点も指摘されている。それは、「ユーザ個人の知識・価値観によって限定された情報獲得を繰り返すことになってしまう」ということである。このことは、知識の拡がりの硬直化を招くことになる。

そこで、この問題に対処するために「協調的情報フィルタリング (collaborative filtering)」が研究され用いられてきた。この方法は、個人の知識・価値観のみならず、他人の知識・価値観も、情報の選別のためのフィルタとして取り込み利用するものである。代表的な協調的フィルタリングシステムとしては、例えば、他人が付けた注釈や評価を参考にして情報の選択を行う、あるいは、投票によってランキングし、高スコアがついた情報を優先的にピックアップするというものがある。

しかし、協調的フィルタリングにも問題点がある。注釈付けを行う場合はその煩わしさがついて回るし、フィルタリングが機能するためには、一定数以上の参加者が必要となる場合がある (クリティカル・マスの問題)。そして、一定数以上の参加者を必要とする場合、フィルタリングできるテーマ・領域は、沢山の人数が確保できるものに限られてしまう。また、システムが情報をユーザに提示する際の問題も存在している。例えば、有用な情報の羅列だけでは、その情報が持つ意義が分かりにくいし、カテゴリー分類による表示は、ある情報がユーザの思惑通りのカテゴリーに区分されるとは限らない。

本研究では、これらの背景と、人の情報獲得活動についての考察とを基にして、協調的フィルタリングの新たなシステムを提案する。そして、そのシステムの設計方針に基づいたプロトタイプシステムを構築し、評価を行う。

ここで、先に進む前に、本研究で使用する語句の意味合いを定義しておく。

「情報」とは、人間の手によって記された、あるいはコンピュータによって自動的に生成された、WWW コンテンツ、ニュース記事、テキストファイル等を指す。その情報をコンピュータのユーザが「獲得する」と言う場合には、単に情報を見付けるということだけではなく、その情報の意義・価値を判断した上で、ユーザ自身の知識として組み込まれることを意味している。

1.2 本論文の構成

第 2 章では、協調的情報フィルタリングの概要を示し、既存のシステムを概説する。そして、協調的フィルタリングのシステムを考える際に解決しなければならない問題点を述べる。

第 3 章では、本研究で提案する情報獲得の支援のためのシステムについて、基本的な設計方針を述べる。この設計方針は、コンピュータ・ネットワーク環境のユーザの情報獲得活動についての考察と、第 2 章で挙げる協調的フィルタリングの問題点に基づいている。

第 4 章では、第 3 章で述べた設計方針に従ったプロトタイプシステムの実装について説明する。

第 5 章では、適当な例題を設定した上でプロトタイプシステムを使用した実験を行い、効果の検証と問題点の洗い出しを行う。

第 6 章では、結びとして本研究のまとめを行い、今後の課題について述べる。

第 2 章

協調的情報フィルタリング

2.1 情報フィルタリング

第 1 章で述べたように、情報の氾濫とも言える状況に対処する具体的な技術として、情報検索と情報フィルタリング (Information Filtering) が研究され用いられてきた。情報フィルタリングの基本的な考え方を以下に記す。

1. ユーザのニーズ・興味・嗜好をプロファイルとして記述しておく。
2. そして、コンピュータ・ネットワークに流れている大量の情報に対して、そのプロファイルを用いてふるいにかける (即ちフィルタをかける)。
3. その結果、ユーザは、自分が必要とする情報だけを優先的に選択できる。

簡単な例を出すと、電子メールで用いる *kill file* がある。*kill file* に自分が読みたくない相手のメールアドレスを記入しておけば、メーラで電子メールを読む時には、その相手からのメールは表示されない。つまり、この例では、 *kill file* がプロファイルになっており、それを参照にしてフィルタリングが行われているわけである。

情報検索と情報フィルタリングの差異には、次のものがある。

- 検索の対象は静的なデータベースであり、フィルタリングの対象は動的なデータストリームである。
- 検索は単発の問い合わせ (query) を用い、フィルタリングは連続した興味からなるプロファイルを用いてふるいにかける (即ちフィルタをかける)。

川の流れのように絶え間なく流通しているネットワーク上の情報群から、新しくて必要な情報を日常的に取り出すには、逐一検索するというよりもフィルタによってふるいにかけるというアプローチが有効である [1]。とは言うものの、情報フィルタリングのために必要な技術と情報検索のためのそれとでは、共通する部分が多く、両システムの最終的な目的は本質的には等しい [1]。例えば、WWW は日々大量のコンテンツが生成されたり更新されるということにおいて、動的なデータストリームとも言えるが、実際に、WWW の有用なページを見つける場合は、検索というスタイルをとることが多い。情報獲得活動においては、情報フィルタリングと情報検索は相まって機能するものであると見做すべきである。

フィルタリング方法については、3 つの分類ができる (Malone らによる分類 [2])。この分類は、ユーザは如何にして情報を選択あるいはふるいにかけるのかについてなされたものである。

cognitive filtering

情報そのものの内容とユーザのニーズに基づいたプロファイルとをマッチングさせることで、ユーザに適切な情報を提供する。そのための方法としては、AND、OR、NOT のブーリアン操作子を使ったキーワード列からなるプロファイルを利用したり [2]、語彙間の関連性を空間ベクトルで表現したプロファイルを利用するものがある [3]。ユーザの振舞いをフィードバックすることで、自動的にプロファイルをアップデートするテクニックを持ったフィルタリングもある。たとえば、記事の注視時間の長さによってトピックスへの興味の度合を計り、それをプロファイルに反映させるものがある [4]。

social filtering

社会的な人間関係と個人の主観による判断に基づいてフィルタリングを行う。つまり、この人はこの道の権威なのでこの人が言ったことには信憑性がある、だとか、ボスから来たメールは要注意だ、などの判断を用いる。

economic filtering

ある情報を得ることに、どれだけコストが掛かるかによってふるいにかける。このコストとは、情報何バイトにつきいくら、といった明示的な課金として表されることもあるし、「この情報は大勢の人に向けられたものだから情報単価は安い」とするような表現もある。

2.1.1 情報フィルタリングの問題点

先に述べたように、一般に情報フィルタリングは、ユーザが必要とする情報を優先的に収集するために、そのユーザのニーズ・興味・嗜好を反映させたプロファイルに基づいて行なわれる。しかし、ユーザ個人のプロファイルのみに依存してしまい、ユーザ個人の価値観によって限定された情報収集を繰り返すことになる。このことは、知識の広がりの硬直化を招くことになる。そして、新しい知識に出くわす楽しみを味わったり、積極的な探求欲求を刺激したりするのを狭めてしまう危険性がある [4]。また、個人が自分にとって未知の分野に関する情報を入手しようとするとき、何をキーワードにして情報源に当たればよいのかはっきりしない場合が多々ある [1]。

2.2 協調的フィルタリングの研究動向

上記 2 つの問題点は、これまで情報フィルタリングの研究において言及されてきた。そこで打ち出されたのが、「協調的情報フィルタリング (Collaborative Filtering)」というアプローチである。その基本的立場は、次のようなものになる。

> ある情報に対して、それが有益であるのか、信頼性はあるのか、同僚にとっても役に立つのかなどの評価やコメントを、ユーザがつける。そして、その評価/コメントを仲間や同僚間で共有し参照できるようにしたり、評価結果を集計して順位付けの表示をしたりすることで、フィルタリングの機能を果たす。

つまり、このフィルタリング方式は、自分が情報を手に入れるときに他人の意見を参考にするという、我々が日常的によく行っている行為に立脚している。

協調的フィルタリングは、前述の cognitive filtering と social filtering とを組み合わせたものであるとも言える。キーワードマッチングなどの機械的な処理やヒューリスティックな評価/コメントは cognitive filtering として行われ、新たに情報を得る際に評価/コメントを参考にすることについては social filtering として行われることになる。

協調的フィルタリングを利用したシステムは、既に幾つか登場しており、それぞれ成果をあげている。以下に 4 つの例を挙げてみる。

Tapestry[5] 電子メールと Netnews のためのシステムである。

ユーザは読んだ記事に自由文の注釈 (annotation) を付けることができる。この注

釈は、記事本体と共にシステム内に保持される。そして、情報を欲するユーザは、query を発行することによって、記事内容だけでなく記事に付与されている注釈に基づいて記事を選択をすることがきる。

例えば、山田さんが fj.comp.music で何か面白そうな記事を探そうとしているとする。そこで、彼は query として 'midi' というキーワードを設定する。また、山田さんは、同僚の鈴木さんがこの手の分野に詳しくてニュースグループをいつもチェックしているということを知っているので、鈴木さんが注釈を付けた記事を探すための query も記述する。かくして山田さんは query に基づいてフィルタリングされた新たな記事を得ることが出来る。

この場合、query がプロファイルとして機能している。また、鈴木さんが注釈を付け、それを山田さんが参考にすることで協調的フィルタリングが機能していることになる。

このシステムは、query を記述する際に、SQL ライクな専用言語を使用する必要がある。そのため、何が知りたいのかを予めはっきりさせておくことが要求されてしまう。

Grouplens[6] Netnews のためのシステムである。

ユーザは記事に対してその価値を例えば 5 点満点で投票する。そして、そうして行われた投票の履歴から、似たような評価傾向にある人、つまり興味の方向性が似ている人を割出し、その人達同士による評価を参考にすることができるようにする。

例えば、木下さんがあるニュースグループの記事を Grouplens 用に改造されたニュースリーダで読んでいるとする。彼が行う採点は、Better Bit Bureaus と呼ばれるサーバに送られる。そのサーバでは、評価が集計され、どのユーザが似たような評価をしているのかが計算される。その結果、山本さんというユーザが浮かび上がった。そこで、木下さんが既に評価していて山本さんが未読であるような記事について、山本さんのためにその記事の予測スコアを BBB サーバは計算して配送する。予測スコアの計算は、木下さんと山本さんの類似度と木下さんの評価をもとになされる。その予測スコアは、改造ニュースリーダに表示され、山本さんにとってはその予測スコアが高いものほど有用な記事であろうことがわかる。木下さんが未読で山本さんが評価済みであるような記事に関しても同様である。なお、実際にこのシステムを利用するに当たっては、プライバシーの問題のため、実名ではなく匿名・変名を

用いる。

このシステムでは、評価履歴がプロファイルになっている。そして、評価を重ねていくことが、協調的フィルタリングを効果的に機能をさせる要となっている。

このシステムは、ユーザが大人数存在しなければ十分に機能しない。そのために、対象となる情報ソースが限られてしまう。また、フィルタリングの機能を享受するまでに、ある程度時間がかかる。

Firefly[7] ユーザの興味を引きそうな映画・音楽・本などに関する情報を提供するためのシステムである。これは WWW 上で展開されている。

ユーザは Firefly のページにて、例えば、自分が今まで観たり聴いたりした映画・音楽作品について、7 点満点の評価点を付ける。その評価傾向に基づき、そのユーザが好みそうな作品が予測され、表示される。評価の集計とその結果を用いた予測計算のアルゴリズムは、Grouplens と似たものになっている。また、具体的な作品名などを query として与えれば、それに付けられた評価点・コメント・傾向が似ている作品名といった情報を手に入れることができる。さらに、ユーザ各人はページを持っており、そこに好きな作品名・アーティスト名・評論などを記述することができる。それによって、自分と似たような興味を持つ人の見識を参考にすることができる。

このシステムでは、Grouplens と同じく評価履歴がプロファイルとなり、それによって協調的フィルタリングが機能する。また、自分でコメントを付けたり他人の意見を見たりすることもでき、そうした行為によってもフィルタリングの機能は増強される。

このシステムは、Grouplens と同様に、大人数のユーザが必要である。また、評価がほとんどなされていない対象については、フィルタリングの効果が期待できないという点も、Grouplens と同様である。

Pointers and Digests[8] 小規模グループ内で、有用な情報を効率よく共有するためのシステムである。このシステムは、Lotus Notes の環境上で機能する。

ユーザは、pointer なるファイルを作成する。この pointer は、a) 有用な情報本体へのハイパーテキストリンク b) その情報のタイトル、日付、保持されているデータベースの名前 c)pointer を作成した人によるコメント をパッケージしたものである。pointer は、同僚/仲間にメールとして送信することもできるし、Information Digest

なる形態で共有データベース化することもできる。
このシステムでは、「ある特定の人物から送られる pointer を受け取りたい」あるいは「自分はある特定の仲間に poiner を送る」という設定が、プロファイルとしての役割になっている。
このシステムを利用するためには、Lotus Notes を導入しなければならない。また、外部のネットワークに流れている情報群に対しての有効な機械的なフィルタリングをサポートしていない。

2.3 問題点

しかし、協調的フィルタリングにも解決すべき問題点が残っている。既存のシステムはいずれも、下記の問題点を何らかの形で有している。

- 情報に対する有用度の点数付け程度ならまだしも、自然言語による注釈付けをするとなると煩わしい。
- 点数付けや投票によって嗜好が近しい人同士を割出し、その人達同士で評価をあてにし合うといったタイプのシステムについては、利用者数が少ないと嗜好が類似した人が見付けられず、フィルタリングが機能しないという問題 (クリティカル・マス問題) がある。
- 評価が点数などで表される場合、最初になされた評価が低いと、その情報はつまらないものであると認識され、誰も手を付けないままにされてしまう可能性がある。
- 有用だと思われる情報の羅列を単に提示されるだけでは、その情報がユーザにとってどのような意義を持つのかが分かりづらい。
- 情報をカテゴリ分類して提示する方法もあるが、ある情報のカテゴリが一意に決まるとは限らないし、ユーザの思惑通りのカテゴリに区分されるとは限らない。

また、協調的情報フィルタリングのシステムを考えるとき、以下の 2 つの問題領域が存在する。

- 興味が類似しているユーザ同士をどのようにして割り出すか、あるいは、誰の推薦を受けることにするのか。

- そうして捜し出した他者からの知識をどのような形にして使うのか。

本研究では、上述の問題点の解決を図りつつ、後者の領域に焦点を当てたシステムの提案を行うことにする。

第 3 章

システムの設計

本研究の目的は、第 1 章でも述べたように、

> 協調的情報フィルタリングの新たなシステムを提案し、プロトタイプシステムによってその効果を検証する。

ということである。本章では、まず、新たなシステムに必要な方針を打ち出すために、コンピュータ・ネットワーク環境のユーザがどのような形で情報獲得活動を行っているのかを考察する。続く節で、システムの設計方針について述べる。

3.1 設計方針を見出すためのアンケート調査

「有用な情報を優先的に選択したい、かつ新たな発見的な情報獲得もしたい」という要求が存在する情報獲得活動を、ユーザはどのように行っているのかを概観するために、アンケート調査を行った。以下が調査のポイントである。

- 獲得した情報の所在をどのようにして知ったのか。
- どのような方法でその情報を獲得したのか (どこにその情報が存在していたのか)。

アンケートは、

> 無作為に選んだ本学情報研究科の学生 21 名を対象に、「日々の研究活動の中で、最近どのような情報獲得活動を行ったか」を、設問に沿って回答してもらう

というものである。自由文による質問と回答になっている。

3.1.1 結果と考察

表 3.1 に回答結果を示す。表の見方は、以下の通りである。

- 表上段の 「口頭、書籍 (論文等)、E-mail、Netnews、WWW」は、回答者が情報を得るきっかけとなった情報ソースのカテゴリ (種類) であり、同時に、その情報が存在していたソースのカテゴリでもある。

- 「獲得した情報の所在をどのようにして知ったのか」という点については、“ Q ” で表し、「どのような方法でその情報を獲得したのか」については、“ G ” で表している。

- 「他には?」という欄には、回答者がその情報を得る過程で、何か他にも有用な情報が得られたかどうかを記している。“ ○ ” は他の有用な情報が得られ、コピーや記録などの具体的なアクションを起こしたことを意味し、“ △ ” は他の有用な情報が得られたのだが、記憶に留めておく程度で何らアクションを起こさなかったことを意味している。“ × ” は、獲得した情報以外には有用な情報を得ていないことを意味している。

表に沿って回答者の挙動の例を挙げると、

> 回答者 G は、人から口頭で、有用と思われるある情報の示唆を受け、WWW を利用してその情報を手に入れた。その過程で、別の有用な情報も手に入れ、それについても何らかの処理を施した。

となる。

また、回答者 H は、ある書籍で有用と思われる情報の所在を知ったが、その有用な情報そのものを得たのは別の書籍からである。このように、“ Q ” と “ G ” が同じカテゴリ内に記されていたとしても、同一のソースを指しているとは限らない。同一のソースで “ Q ” と “ G ” がなされている場合のみ、“ (*) ” を記入しておく。

アンケートの回答結果から考察したことを以下にまとめる。

- 回答者は、様々な情報ソースを組み合わせて情報を探し出している。例えば、最終的には WWW で得た情報も、その URL は人から聞いたものであったり、Netnews で見付けたものであったりする。

回答者	口頭 (他者)	書籍 (論文等)	E-mail	Netnews	WWW	他には?
A	Q	Q			G	×
B	Q	Q, G				△
C		Q, G (*)				○
D	Q	Q, G				○
E		Q			G	○
F	Q			Q	Q, G	×
G	Q				Q, G	○
H		Q, G				△
I		Q, G				○
J				Q	G	○
K					Q, G	△
L		Q, G				○
M		Q, G				○
N	Q		Q	Q	Q, G	○
O	Q	G				×
P	Q				G	△
Q		Q, G (*)				○
R		Q, G				△
S				Q	G	○
T		Q, G (*)				×
U		Q, G				○

表 3.1: アンケート結果

- 回答者は、それぞれ自分なりの情報の体系をあらかじめ持っている。そうした情報は、その人にとっては一つ一つが独立して存在しているのではなく、その人なりの関連付けがなされている。そして、情報を探す時には、どこから当たれば良いのか、そのリンク付けされた情報体系を用いて、ある程度の見当を付けながら行っている。新たに情報を獲得すると、その情報はその人の情報体系の中に組み込まれていく。図 3.1は、この振舞いを表したものである。あるソースから得た情報が自分にとって有用なものとして感じるとき (図 3.1中で記すところの「刺激」を受けるとき)、その情報は、自分が既に保有している情報と何らかの関係を持つ。そして、そこで得た情報を切っ掛けにしてさらに新たな情報を探そうとするとき、関連する情報を足掛かりにして適当な情報ソースを探りに行く (図 3.1中で記すところの「獲得行動」)。

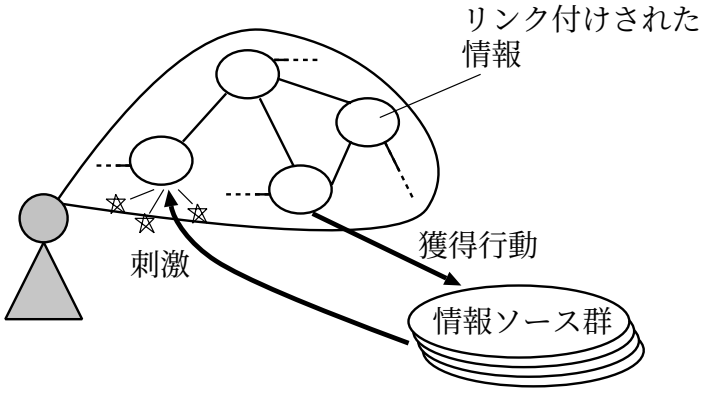

図 3.1: 情報獲得のモデル図

- 欲しかった情報を得る過程で、目的とは別の有用な情報を得ることも多い。

3.2 設計方針

本節では、情報獲得を支援するシステムの設計方針を述べる。それに先立ち、第 2 章で述べた協調的情報フィルタリングの問題点とアンケート結果からの考察をまとめておく。

1. 注釈付けは煩わしい。

2. 大人数での利用が必要条件となっている場合、クリティカル・マス問題が発生する。

3. 点数評価の場合、初期評価の影響力が大きい。

4. システムが情報をユーザに提示する際には、単なる羅列やカテゴリ表示では問題がある。

5. 人々は様々な情報ソース (例えば、WWW、NetNews) を組み合わせて、情報を獲得していると考えられる。

6. 人々は様々な形態の情報ソースからもたらされる様々な情報を、個人で勝手にリンクさせていると考えられる。そして、その情報体系を利用して、新たな情報を獲得する。

7. 欲しかった情報を得る過程で、別の有用な情報をも得ることが多い。

上記の点に鑑み、以下を設計方針とする。

方針 0 — 本システムの位置付け

ユーザは、日頃から各自の好みの検索/フィルタリングシステムを使いながら情報を獲得している。本システムとしては、そうした既存の便利なシステムの代替物としてではなく、それらを利用した情報獲得活動からこぼれ落ちてしまう情報や見付けられない情報を獲得するためのものとして利用されることを狙う。(上記 5., 7. に対応)

情報をフィルタにかけるという点については、協調的フィルタリングの方法論を踏まえ、他ユーザの知識を利用する。

また、ユーザ同士はある程度興味が共通する部分があるということが、本システムを利用する上での前提となる。自分と興味を全く異にする他者の知識は、フィルタとして利用することは考えにくいからである。したがって、本システムでは、第 2 章の最後で記したように、他者からもたらされる知識を如何に利用しやすくするかということに重きを置いている。

方針 1 あらゆる領域に対し、そこに興味を持っているユーザが 2 人以上存在すれば利用できるようにする。(上記 2. に対応)

方針 2 様々な情報ソースを扱えるシステムにする。(上記 5. に対応)

Netnews だけのため、WWW だけのため、というシステムにはしない。

方針 3 情報と情報とのつながりを示すことで、各情報が持つ意義をユーザに示唆してみる。(上記 1., 3., 4., 6. に対応)

システムを介して他ユーザから新しい情報が提供されるとき、自分が既に保有している自分なりの情報体系との関連をわかりやすい形で表すようにする (図 3.2)。この表示は、「提示された未知の情報は、自分にとってどのような意義を持つのか」の見当をつける指針として利用できると考えられる。

システムによって、自分が持っている情報体系に他ユーザが持っている情報体系が接ぎ木されていく。そうして作られる情報間のつながりは、自分にとって新しくかつ有用な情報へ行き着くための道標として示される。

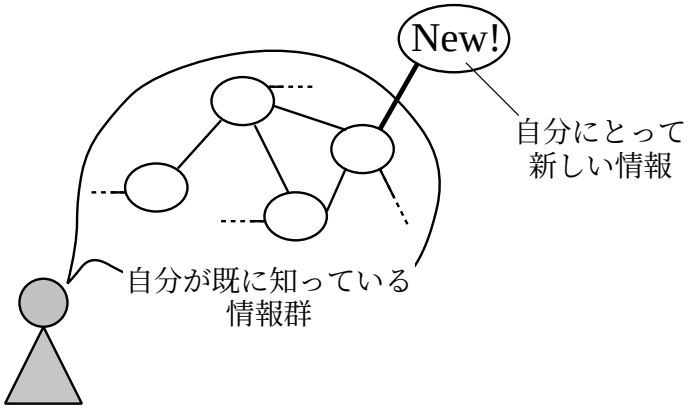

図 3.2: 「既知の情報」と「未知の情報」のつながり

この方針を具体化するために、本システムでは、各ユーザが持っている情報を、ある形態にして扱うことにする。これについて、次の節で詳しく述べる。

3.3 情報地図

人がある情報を獲得するとき、その人なりのリンク付けがされた情報体系を利用しているのではないかとこれまで述べてきたが、そのリンク付けされた情報体系は、新たな情報に行き着くための、いわば地図とも言える。そこで、この章で述べてきた「リンク付けされた情報体系」を視覚化したものを、今後「情報地図」と呼ぶ。

図 3.3は、情報地図の構造を表している。情報地図とは、個人が持っている情報間の関係

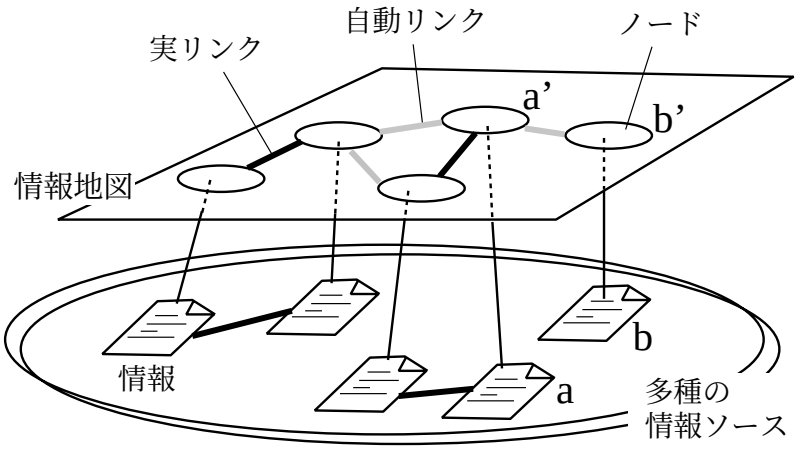

図 3.3: 情報地図の基本構造

をグラフによって明示的に視覚化したものである。グラフのノードは、情報を表し、エッジは、情報間に何らかの関係があることを示している。本論文では、そのエッジを、「リンク」と呼ぶことにする。

情報地図の特徴を以下に挙げる。

1. 情報地図のノード

   - 情報地図のノードは、情報本体ではなく、情報の在処を指し示す URL とその情報のタイトルから成るデータとする。つまり、ノードを介して、そのノードからの URL リンク先である情報本体にアクセスすることになる。
   - 「WWW のコンテンツ (図 3.3 中の a)」、「個人的なファイル (図 3.3 中の b)」といった、異なる情報ソースに存在している情報を一元的に表示する (図 3.3 中の a’, b’)。

2. 情報地図のリンク

   - WWW 上の URL によるリンクのように、現実世界で明示化されている情報の関係を反映する (図 3.3中の「実リンク」)。これは、httpd が記録するユーザのアクセスログを基に生成される。
   - 実際には関係付けられていない情報に対しても、その情報を解析することにより、自動的にリンクを張る (図 3.3中の「自動リンク」)。自動リンクの生成に

ついては、例えば、tf*idf 法とベクトル空間モデル [9] を利用して類似度を求め、その結果に基づいてリンク生成を行うという方法がある。

- ユーザの手によるリンクの編集も可能である。実リンク、自動リンク共に存在しない情報間にも、ユーザの主観によりリンクを張ることができる。また、生成された自動リンクを消去することもできる。

本システムは、各ユーザ毎に情報地図を生成し、合成する (図 3.4)。合成された情報地図をユーザが参照することで、自分にとって興味がある文脈を軸にして新たな情報を見付けることが出来るようにする。

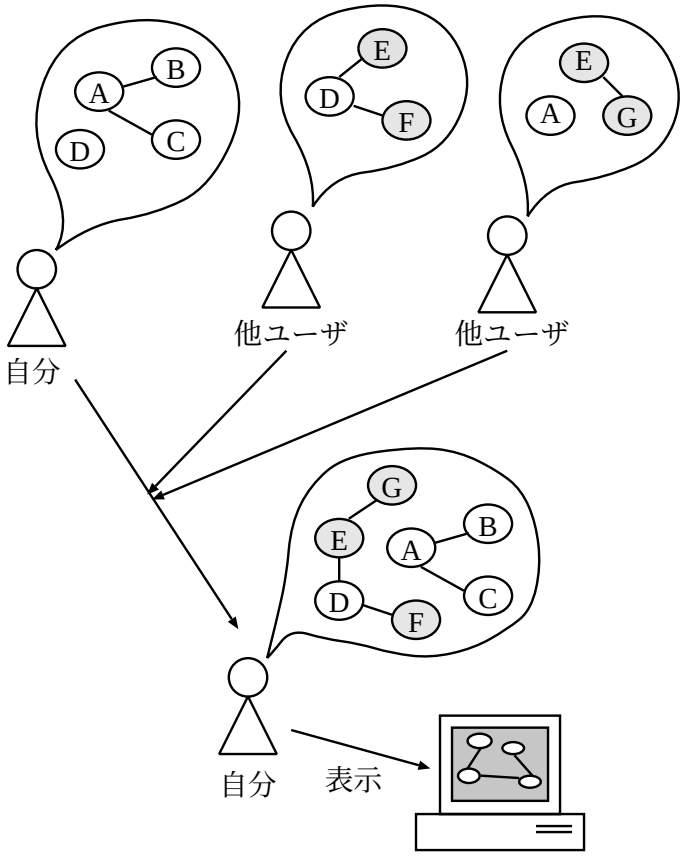

図 3.4: 情報地図の合成

先程述べた、システムの設計方針 3 を、「情報地図」という語を用いて言い換えると、次のようになる。

本システムでは、各ユーザが、自分の情報地図に他ユーザの情報地図を結合させた情報地図を参照できるようにすることで、他ユーザの知識を利用できるようにし、それによって発見的な情報獲得を促すことを目論む。

3.4 システムの要件

システムは、情報地図を表示する準備として、情報地図として表示したい情報を一旦取り込み、そして解析して、URL、タイトル、キーワード、などの情報の属性を得る必要がある。自動リンクの生成は、それらの属性を利用して行われるからである。したがって、ユーザは、システムを利用するときには、あらかじめどの情報を情報地図として表示させるのか決めておかなければならない。そのためには、システム用の設定ファイルを用意し、そこにターゲットとなる情報を記録しておき、そのファイルをシステムに読み込ませる、ということが必要になる。その際、プライバシーの関係上、情報地図として他ユーザに提供したくない情報もあると考えられるので、自分が持っている情報の内、他ユーザに提供しても構わない範囲を指定しておく。

また、実リンク生成のために、システムは、WWW の proxy サーバも兼ねたコンピュータに実装されることになる。システムのユーザは、日頃各自で行う情報獲得活動において、そのコンピュータを proxy サーバとして利用する。すると、ユーザのアクセス履歴より実際に張られている情報コンテンツ間の URL リンクを抽出できるので、その結果を情報地図の実リンクとして反映させる。

第 4 章

プロトタイプシステムの実装

本章では、前章で述べた設計方針に基づくプロトタイプシステムの実装に関する説明を行う。まず、システムの利用シナリオとシステム構成の概略を示す。そのあと順次、情報地図データの生成に用いられるデータ、情報地図データの生成部、そしてインタフェースの生成部についての詳細を述べる。

4.1 システムの概説

4.1.1 システム利用の概略

プロトタイプシステムの利用の大まかな流れを、以下に示す。先頭に振ってある各番号は、図 4.1中の番号に対応している。

1. 本システムのユーザは、システムが実装されているコンピュータを WWW の proxy サーバとして利用しながら、日頃から情報獲得活動を行っている。

2. 1. の活動の中で、自分が必要だと思った情報については、以下の処置がなされる。

   - 特定のディレクトリに情報をセーブする。
   - 特定のファイルに情報の URL を記録する。

   自分が所有しているシステム用の設定ファイルには、これらのディレクトリやファイルのパスが記録されている。また、他ユーザも各自こうしたディレクトリやファイル持っており、それらのパスも、設定ファイルに記録されている。

3. 情報地図を参照したいときは、自分の端末上でシステム用のコマンドを入力する。

4. すると、システムのあるモジュールが、自分の設定ファイルを読み込み、その内容に応じた情報地図用のデータを生成してくれる。2. でも述べた通り、設定ファイルには、他ユーザが保有している情報の所在も記録してあるので、生成される情報地図データは、自分の情報地図データと他ユーザの情報地図データをマージしたものになる。情報地図データが更新されたかどうかは、端末の画面上に表示される。

5. ユーザは、情報地図データが更新されたのを確認したら、WWW ブラウザの Netscape を起動し、情報地図システムを提供している URL を open する。すると、java アプレットがロードされる。そのアプレットには、情報地図が表示されている。

6. 彼にとって未知の情報として提示されているノードをダブルクリックすることによって、その情報にアクセスすることができる。また、自動リンクの修正を行うことができ、その操作によっても、情報地図データが更新される。

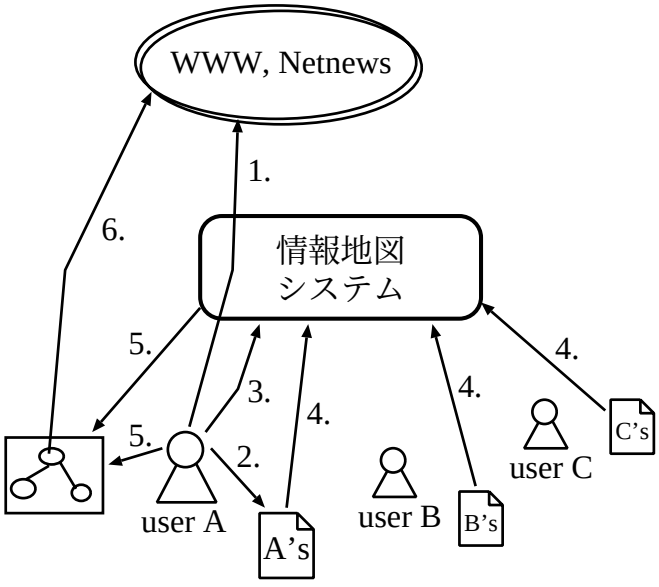

図 4.1: システム利用の概略

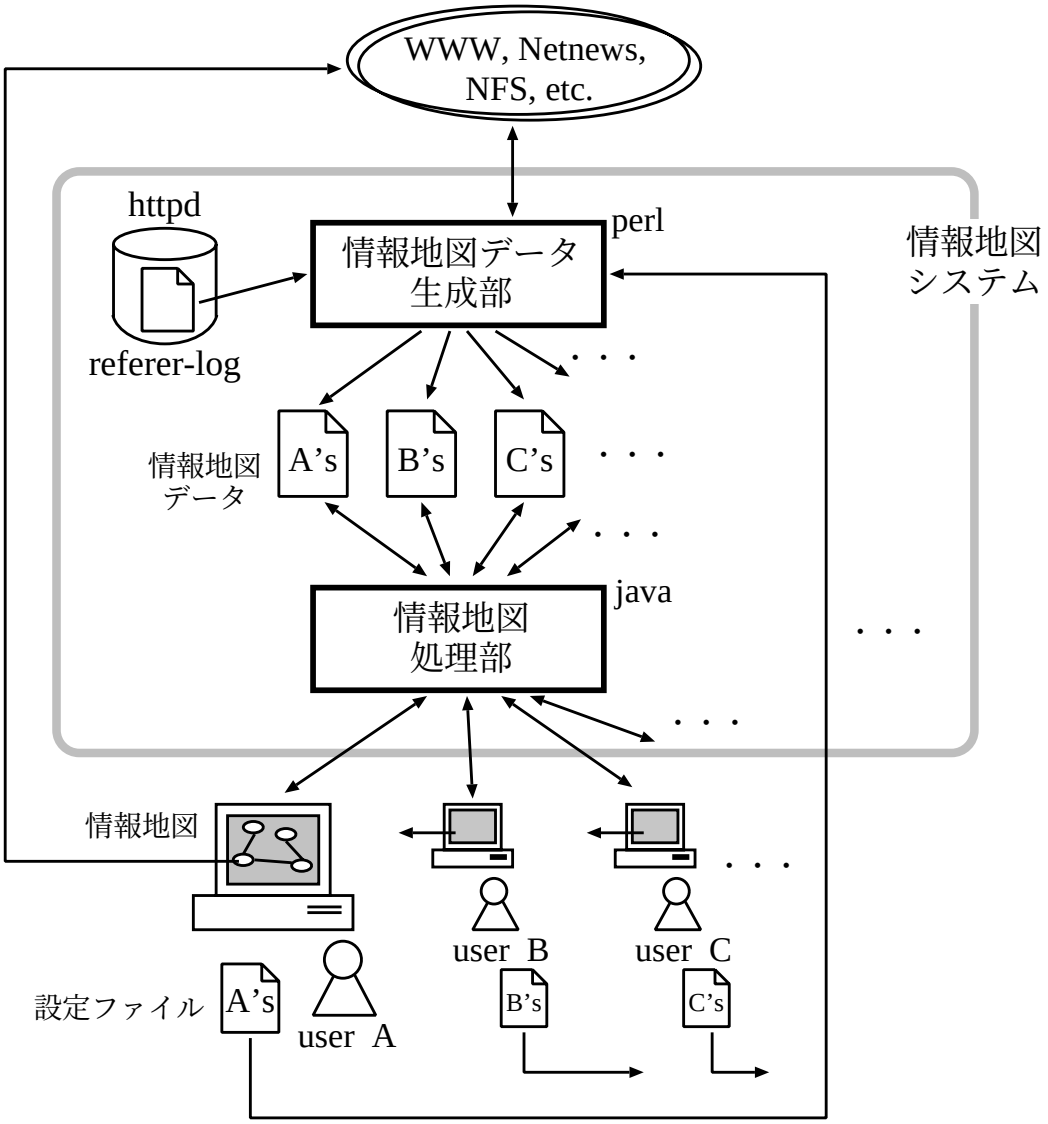

図 4.2: システム構成の概略図

4.1.2 システム構成の概略

図 4.2にシステム構成の概略を示す。図中の「情報地図データ生成部」は、情報地図用のデータを生成するためのバックエンドであり、プログラミング言語 Perl[1]で記述してある。上記の利用シナリオの 4. の部分を受け持っている。「情報地図処理部」は、情報地図データを処理し、グラフィカルな情報地図表示を行い、さらに、ユーザからの反応を処理するためのフロントエンドである。こちらは Java 言語[2]で記述してある。上記の利用シナリオの 5.、6. の部分を受け持っている。

本システムは、一台のコンピュータ上に実装されている。さらに、そのコンピュータは、WWW proxy サーバも兼ねている。本システムのユーザは、普段の WWW を利用する際にも、このコンピュータを proxy サーバとして利用することで、ユーザの WWW のアクセス履歴がシステム内に保存される。

4.2 情報地図データのためのデータ

本システムを利用するあるユーザが、本システムによって直接提示される以外の有用な情報を、自分で見つけ出したとする。その情報が例えばどこかの WWW サイトで公開されていたならば、その URL をシステム専用のファイルに記録しておく。この URL が記録されているファイルをここでは「URL リスト」と呼ぶことにする。あるいは、その情報が自分のディレクトリにセーブしておきたいものである場合、システムのバックエンドが読み出すことができるディレクトリ内にセーブしておく。このディレクトリをここでは「セーブ用ディレクトリ」と呼ぶことにする。

URL リストは 1 個のファイルであり、例を以下に示す。

```
http://www.foo.ac.jp/bar/
http://www.hoge.org/poo.html
        .....
```

セーブ用ディレクトリは、複数のディレクトリをユーザが任意に設定することができる。

[1] version 5.002

[2] Java Development Kit ver.1.1.3 を使用

システムの設定ファイルに、URL リストとセーブ用ディレクトリの絶対パスを書き込んでおく。また、他ユーザの URL ファイルとセーブ用ディレクトリも同様に設定ファイルに書き込んでおく。設定ファイルの記述例を以下に示す。

```
$MINE_FILE='/home/k-kawase/informap/target';          #自分の URL リスト
@MINE_DIR=(          #自分のセーブ用ディレクトリ
'/home/k-kawase/research/',
);
@OTHRES_FILE=(          #他ユーザの URL リスト
'/home/yamada/informap/target',
'/home/suzuki/misc/hoge/target',
);
@OTHRES_DIR=(          #他ユーザのセーブ用ディレクトリ
'/home/yamada/pub/',
'/home/suzuki/misc/pub/misc',
);
        .....
```

4.3 情報地図データ生成部

情報地図データの生成部では、以下のステップで情報地図用のデータを生成する。

step 1 設定用ファイルの読み込み

自分/他ユーザのセーブ用ディレクトリを走査しそこに置いてあるファイルのパス名を得、各パス名の先頭に `file:` を付けた文字列を生成する。それらを、URL リストとマージする。こうして、情報の置き場所全てが URL で表されているファイルを生成する。以下に例を示す。

```
http://www.foo.ac.jp/bar/
```

```
http://www.hoge.org/poo.html
file:/home/k-kawase/reseach/ooioo.txt
file:/home/yamada/pub/setting
        .....
```

なお、続く step 2 〜 step 4 は、1 つの URL 毎に行われる。

step 2 情報本体の読み込み

テキストベースの WWW ブラウザである lynx を用い、上記の URL が示している情報本体を読み込み、その結果を得る。つまり、ここでは

`lynx -source` *URL* > *output*

というルーチンが実行される。

また、自分が見つけてきた情報には “1”、他ユーザが見つけた情報には “0” を、読み込みが終った URL に付けておき、その URL を一時的なファイル (ここでは仮に tmp_file としておく) に保存する。

step 3 タイトル、サブジェクトの抽出

output のヘッダ、タグなどからタイトルやサブジェクトを抽出する。そして、ステップ 2 で生成されている tmp_file に付加する。

step 4 単語の抽出

output を形態素解析ソフトウェア「茶筌 (ChaSen)[10]」で処理し、単語を抽出する。その際、日本語については名詞のみを抽出する。ただし、ひらがなのみのものやカタカナで 1 文字のものは除去する。英語については、ChaSen は単語全てを名詞として扱ってしまう。したがって、意味を成さないと思われる単語 (例えば、the, what, make, ...) を、不要語リストを用いて除去する。この不要語リストは、[11] に掲載されていたものに、適当な単語を付け足したものを利用する。

step 5 単語の重み付けとランキング

step 1 で得られた全ての URL について、step 4 までの処理が終了したら、tf*idf 法により各単語の重み付けを行う (tf*idf 法については、このあとの節で述べる)。 そ

の後、ある値 W より低い重みの単語は除去し、一つの URL に付き、重みが大きい順に N 個の単語を並べる。この時点の tmp_file の例を以下に示す。

```
1**foo title**http://www.foo.ac.jp/barfile**音. 空中. キャンプ.
1**hoge title**http://www.hoge.org/poo.html**宇宙. 日本. 世田谷.
1**no title**file:/home/k-kawase/reseach/ooioo.txt**electoro.tabla.
0**no title**file:/home/yamada/pub/setting**sitar.guitar.
        .....
```

---

step 6 情報地図のノードの自動リンク付け

step 5 でランキングした単語を用いて、自動リンクデータの生成を行う。単語一つ一つについて、その単語を含む情報の組合わせを対応付けたデータを生成し、そのデータ中に、ある情報の組が C 回以上出現する時、その情報間に自動リンクを張ることにする。同時に、httpd の referer log を参照にして、実リンクデータも生成し、この 2 つのリンクデータをマージする。加えて、他ユーザの情報地図データも取り込み、マージする。

tmp_file は、以下のような内容を持つ情報地図データとなる。“+++” が自動リンクを表し、“===” が実リンクを表している。

```
1**foo title**http://www.foo.ac.jp/barfile**音. 空中. キャンプ.
===1**hoge title**http://www.hoge.org/poo.html**宇宙. 日本. 世田谷.
1**hoge title**http://www.hoge.org/poo.html**宇宙. 日本. 世田谷.
+++0**no title**file:/home/yamada/pub/setting**sitar.guitar.
1**no title**file:/home/k-kawase/reseach/ooioo.txt**electoro.tabla.
+++0**no title**file:/home/yamada/pub/setting**sitar.guitar.
        .....
```

4.3.1 単語の重み付け (tf*idf 法)

単語の重要度をスコアリングする方法としては、tf*idf 法 [9] が広く用いられている。tf, idf の各値の定義を以下に示す。

$$\begin{aligned} tf_{ij} &= \text{単語 } t_i \text{が文書 } d_j \text{に現れる割合} \\ &= \frac{t_i \text{の出現回数}}{d_j \text{の単語数}} \\ idf_i &= \text{単語 } t_i \text{の特殊性} \\ &= log(\frac{\text{全文書数}}{\text{単語 } t_i \text{が出現する文書数}}) \end{aligned}$$

文書 d_j における 単語 t_i の 重要度 W_{ij} を、以下の式で求める。

$$W_{ij} = tf_{ij} \cdot idf_i$$

プロトタイプシステムでは、4.3節の step 5 にてこの tf*idf 法によるスコアリングを行い、キーワードを抽出する。

4.4 情報地図処理部

情報地図処理部では、以下のステップで Java アプレットとして情報地図を生成する。同時に、そのアプレットには、情報のタイトル、URL、キーワードを表示するためのテキストエリアと、ノードに対する操作を行うボタンを埋め込む。

step 1 情報地図用データの読み込み

読み込まれるデータは、4.3節で述べた処理によって生成されたものである。すなわち、4.3節の step 6 で記したものと同様のデータを読み込むことになる。

step 2 ノードとリンクのオブジェクトの生成

以下の属性を持つノードオブジェクトを生成する。

- 自分が探してきた情報なのか、他ユーザからもたらされた情報なのかを区別するフラグ
- 情報のタイトル
- 情報の URL

- 情報に含まれるキーワード

さらに、ノードから情報本体へのアクセスを可能にするために、各ノードから URL オブジェクトを生成する。
また、以下の属性を持つリンクオブジェクトを生成する。

- リンクを張る対象となるノードの URL
- 自動リンク
- 実リンク

step 3 情報地図、操作用ボタン、テキストエリアの生成
グラフ生成のアルゴリズムについては、JDK(Java Development Kit) にサンプルプログラムとして付随している Graph Layout アプレットのものを利用する。
情報地図アプレットは、以下の要素から成る。

- 情報地図
- 自動リンクを修正するためのボタン
- 不要なノードを消すためのボタン
- 情報のタイトル、URL、キーワードを表示させるためのテキストエリア

step 4 各種操作の反映
ユーザは情報地図アプレットのボタンを利用して、情報地図の修正を行うことができる。その結果は、CGI(Common Gateway Interface) によって送られ、情報地図データが書き換えられる。

4.5 機能

プロトタイプシステムで実装した機能を以下にまとめる。

ノードとリンク

- ノードをマウスでダブルクリックすることによって、情報にアクセスできる。

- ノード内に情報のタイトルやサブジェクトを表示できる。
  この機能により、ノードがどのような情報を指しているのかの判断材料を提供する。
- ノードの色の違いにより、自分にとって未知の情報と既知の情報を区別することができる。
- リンクの色の違いにより、自動リンクと実リンクを区別することができる。
  この機能により、自分が既に知っている情報から実リンクが張られていている未知の情報を優先的にチェックしてみる、などの判断材料を提供する。

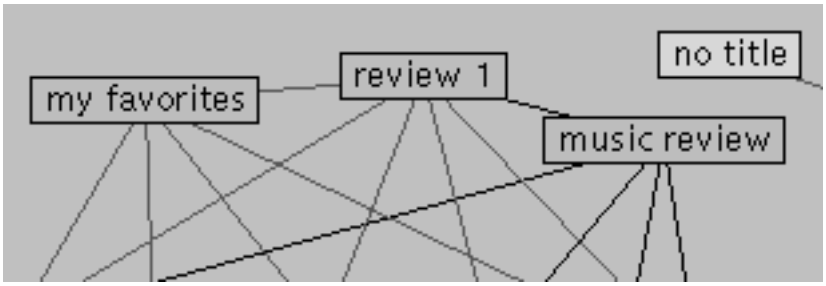

図 4.3: ノードとリンク

各種ボタン

- 情報地図のノードをマウスでドラッグすることにより、レイアウトを変えることができる。
  この機能は、以下のボタンと組み合わせて利用する。
  - “Fix” ボタン—情報地図を固定する
  - “Relax” ボタン—情報地図の固定を解除する

  “Fix” してからであれば、一つずつのノードを動かすことができる。
  “Relax” してから一つのノードをドラッグすれば、そのノードにリンクを張っているノードも追随させて動かすことができる。
- “BOO!” ボタンによって、不要なノードを消去することができる。その結果は、CGIを介して情報地図データに反映される。

- “LINK”、“UNLINK” ボタンによって、自分が望むように自動リンクの修正を行うことができる。その結果は、CGI を介して情報地図データに反映される。

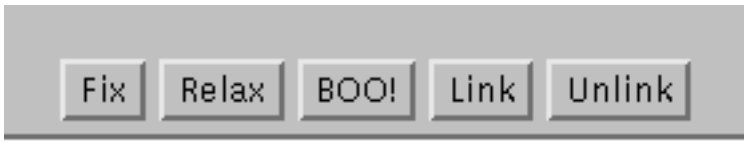

図 4.4: 各種ボタン

テキストエリア

ノードをマウスでシングルクリックすることによって、そのノードが指す情報の URL、タイトル、キーワードがテキストエリアに表示される。この機能により、ノードがどのような情報を指しているのかの判断材料を提供する。

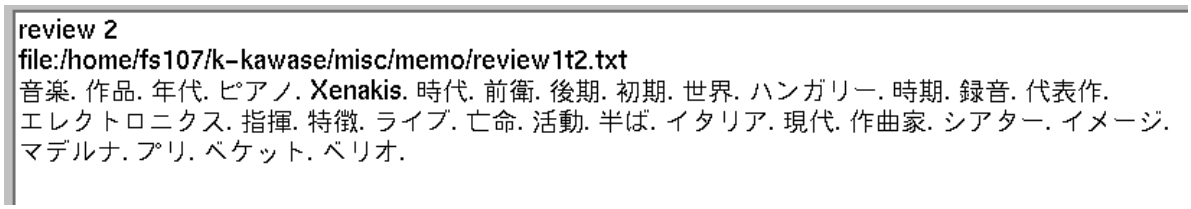

図 4.5: テキストエリア

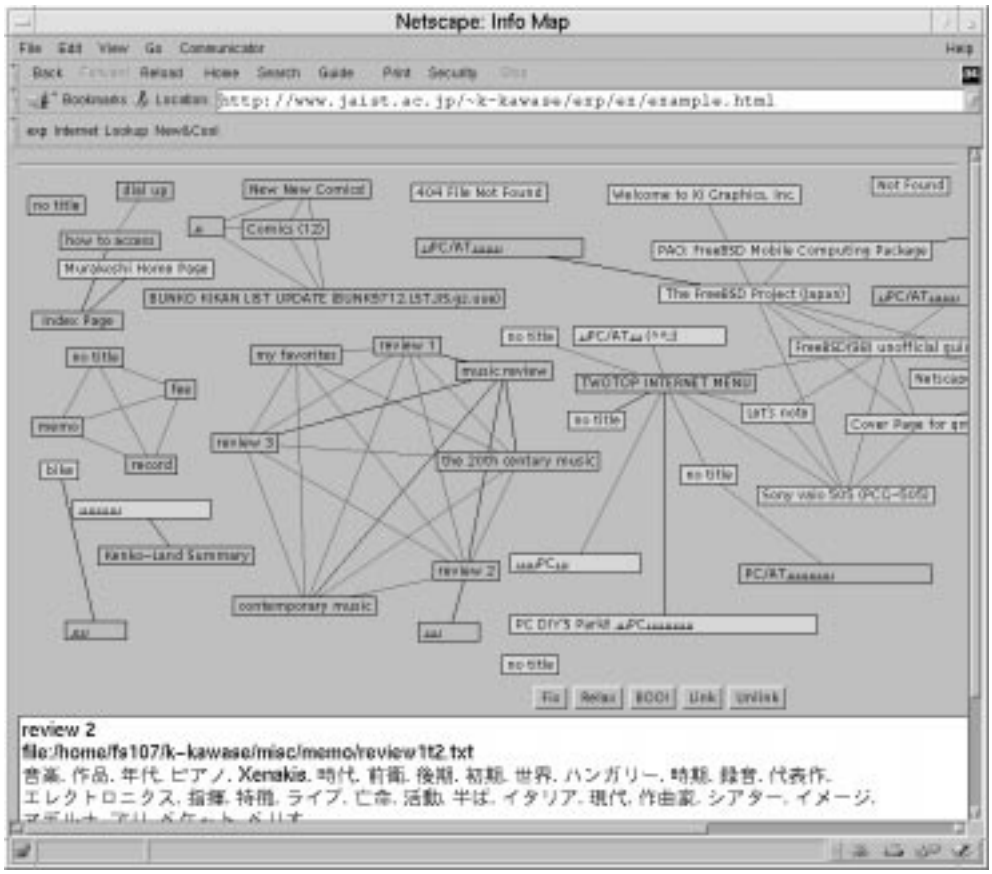

図 4.6: 情報地図アプレット全体画面

第 5 章

評価と考察

5.1 目的

本研究で提案するシステムの有効性を確認するために、プロトタイプシステムを用いた評価実験を行った。評価の目標は、以下の 2 点を確認することにあった。

評価目標 1 他ユーザの情報地図を合成することで有用な情報が得られるか?

評価目標 2 リンクを用いたグラフ表示は、有用な情報を得るのに有効か?

5.2 実験内容

5.2.1 例題設定

被験者は、本大学院の院生 3 名である。被験者たちは、各自のコンピュータ環境を自分なりに構築しカスタマイズすることに興味を持っている。そこで、インターネット上に散見する使えそうな情報、例えば、自作パーソナル・コンピュータに関する情報、各種ハードウェアにまつわる話、OS の設定に関する情報、ネットワークまわりの設定に関するノウハウといったものを、日々チェックしている。そこで、そういった情報を、プロトタイプシステムを利用しながら獲得する。

5.2.2 実験の手順

実験の流れを以下に示す。

1. **被験者:** 各自が普段行っている方法によって情報を収集する。これは随時行われる。
   実験者: 被験者に、情報地図を利用してもらう時間帯を案内しておく。

2. **被験者:** 各自が自分で手に入れた情報を実験者に知らせる。その後、情報地図を呼出し、参照する。
   実験者: 被験者からの情報を受取り、被験者が情報地図を呼び出す前に、情報地図データを更新しておく。

3. **被験者:** 各自が参照している情報地図から得られる情報を評価する。
   実験者: 各被験者が行った評価データを収集し整理する。

4. **被験者:** 3. の作業が終ったら、1. の作業に戻る。
   実験者: 被験者に 1. の作業に戻るように知らせる。

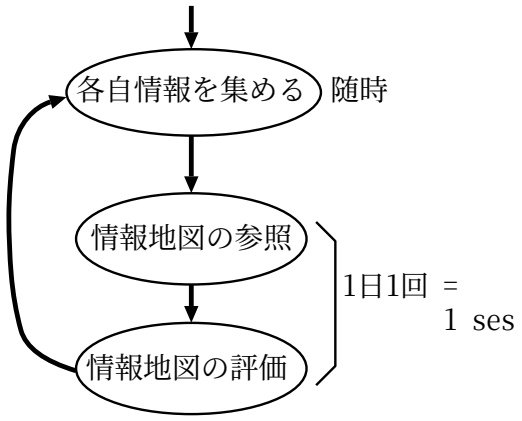

図 5.1: 被験者の作業の流れ

2., 3. は 1 日 1 回連続して行われる。これを 1 セッションとしてカウントする。また、1. での情報の増加が停止することによって 2. で参照される情報地図が更新されなくなる、という状況が 2 回連続したら、実験を終了することにした。

5.2.3 被験者によるプロトタイプシステムの使用

以下は、この実験に限った、システムの使用シナリオである。

1. 被験者 A は Web ブラウザを起動し、情報地図アプレットをロードする。
2. 彼は自分にとって未知と思われる情報を見に行く。どの情報を見に行くかは、彼の自由である。
3. 見に行った情報に対し、以下の評価を行う。
   - 未知であり、かつ有用であると思えたら、アプレットの “FINE!” ボタンを押し、対象ノードを選択する。
   - 未知であり、かつ不要であると思えたら、“BOO!” ボタンを押し、対象ノードを選択する。
   - 未知であり、かつ有用なのか不要なのかよくわからないと思えたら、“HUUM...” ボタンを押し、対象ノードを選択する。
   - 実は既知であり、かつわざわざ記録しておらず、しかし改めて記録しておこうという気になった場合は、 “KNOWN_NEED” ボタンを押し、対象ノードを選択する。
   - 実は既知であり、かつわざわざ記録しておらず、改めて見てもやはり記録しておく気にはならない場合は、 “KNOWN_UN” ボタンを押し、対象ノードを選択する。
4. また、自動リンクに対し、以下の評価を行う。
   - 情報地図内で、自動リンクが張られるべきだと思えるところがあれば、 “link” ボタンを押し、対象ノード 2 つを選択する。
   - 情報地図内の自動リンクで、不適切だと思えるものがあれば、“unlink” ボタンを押し、対象ノード 2 つを選択する。
5. 2.〜 4. を順不同で繰り返す。但し、1 セッション毎に全ての情報を見に行く必要はなく、どうするかは彼の自由である。
6. 見に行くべき情報が無くなったら作業を終了する。

このシナリオ中の被験者 A の各アクションの結果は、step 2 で用いるデータとして記録される。

5.3 結果と考察

被験者たちの作業は、4 日間にわたって行われた。つまり、各自が探してくる情報の増加は 3 回目のセッションからストップし、実験は 4 セッションで終了した。最終的には、各被験者は、自分にとって新しいものとして提示された情報を全てチェックした。

5.3.1 情報地図の合成による効果

第 5.1節で述べた評価目標 1「他ユーザの情報地図を合成することで有用な情報が得られるか?」について、2 つの観測を基に考察する。

新しくかつ不要でない情報の累積結果

第 5.2.2節で述べた 1. での作業による情報の累積数と、各自が “FINE!” あるいは “HUMM...” の評価をした情報の累積数を図 5.2に示す。図中の ‘A_self’ は図中の ‘A_self’ は、被験者 A が自分で見つけてきた情報の累積を表し、‘A_fine_humm’ は、彼にとって「新しくかつ不要でない情報」の累積を表している。

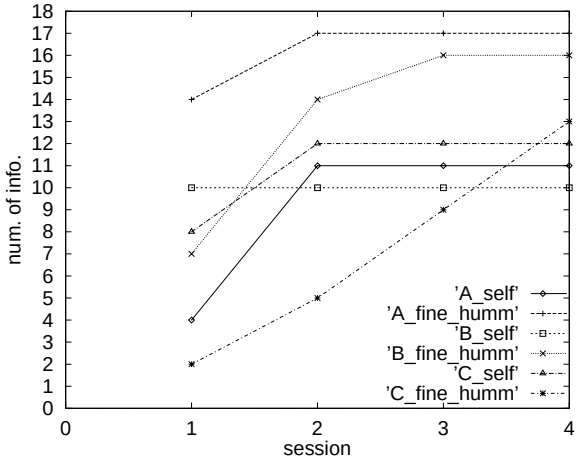

図 5.2: 情報の累積結果

この結果から、システムの利用によって、各被験者は、「新しくかつ不要ではない情報」を増やしていることが定性的に言える。

情報の有用度と不要度

被験者が新たに発見した情報のうち、自分にとって有用だった情報の割合と、不要だった情報の割合を求める。これらの数値は、以下の式で求める。

$$
\begin{aligned}
\text{有用度} &= \frac{\text{被験者 A にとって、新しくかつ有用な情報数}}{\text{被験者 A が新たに発見した情報数}} \\
&= \frac{\text{被験者 A が“FINE!”の評価を与えたノードの数}}{\text{被験者 A がダブルクリックした黄色いノードの数}} \\
\text{不要度} &= \frac{\text{被験者 A にとって、不要な情報数}}{\text{被験者 A が新たに発見した情報数}} \\
&= \frac{\text{被験者 A が“BOO!”の評価を与えたノードの数}}{\text{被験者 A がダブルクリックした黄色いノードの数}}
\end{aligned}
$$

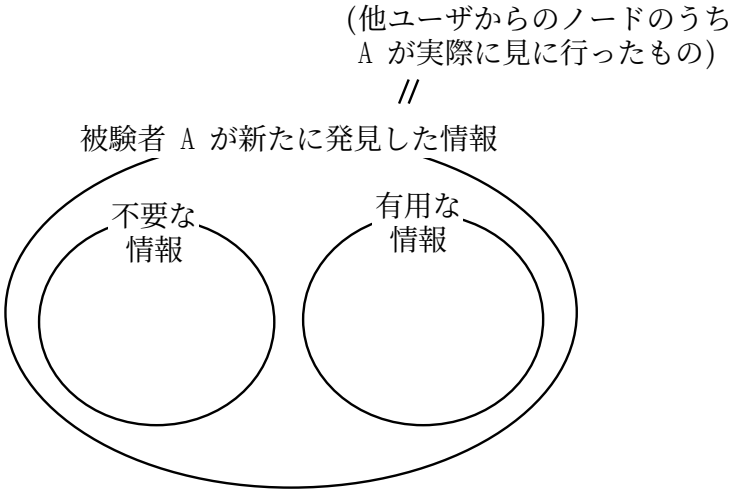

図 5.3: 有用な情報と不要な情報のモデル

第 4 章でも記したように、被験者 A にとって新しいと思われる情報を指し示すノードには、黄色い色を付けてある。そして、情報地図のノードをダブルクリックすることによって、情報本体にアクセスすることになる。したがって、「被験者 A が新たに発見した情報の数」は、「彼がダブルクリックした黄色のノードの個数」となる。

各被験者におけるセッション毎の有用度と不要度を表 5.1 に示す。数値は、「有用度, 不要度」の順序で記してある。「—」の表記は、被験者がシステムから提示される情報は既にすべてチェックし終って、もうノードをダブルクリックする必要がなくなったことを意味している。また、最終的な結果を円グラフとして図 5.4に示す。

被験者	1	2	3	4	total
A	0.35, 0.18	0.60, 0.40	—, —	—, —	0.41, 0.23
B	0.46, 0.36	0.70, 0.30	0.50, 0.00	—, —	0.57, 0.30
C	0.67, 0.33	0.50, 0.50	0.00, 0.33	0.50, 0.33	0.38, 0.33

表 5.1: 有用度と不要度

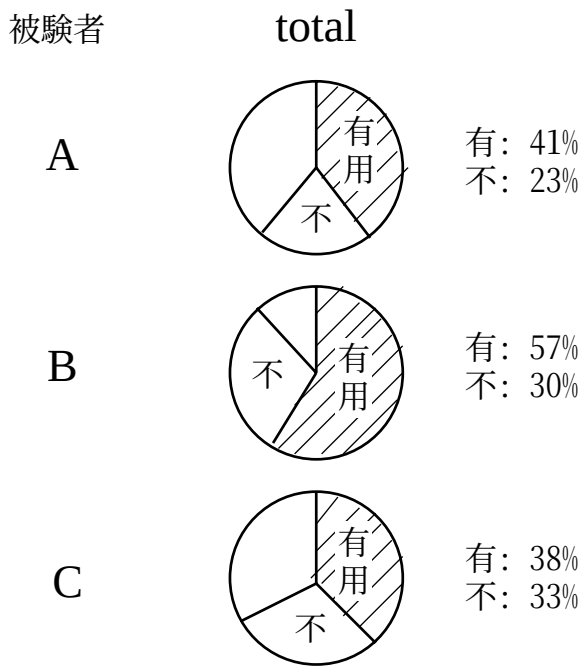

図 5.4: 有用度と不要度 (円グラフ)

被験者 C の 3 回目のセッションで、唯一 有用度 < 不要度 となってしまったが、どの被験者においても、最終的には、有用度 > 不要度 となっている。つまり、自分にとって新しいと思われる情報を見に行くとき、それが有用である場合の方が多く、不要な情報で

ある割合が 3 割前後で済むことが確認できる。

以上 2 つの観測結果から、本システムの利用により、ユーザは自分にとって新しくかつ有用な情報を、確実にしかも効率よく獲得することができると言える。

5.3.2 グラフ表示の有効性

第 5.1節で述べた評価目標 2「自動リンクを用いたグラフ表示は有用な情報を得るのに有効か?」について考察する。そのために、図 5.5に示す数値 F と B の比較を考えた。

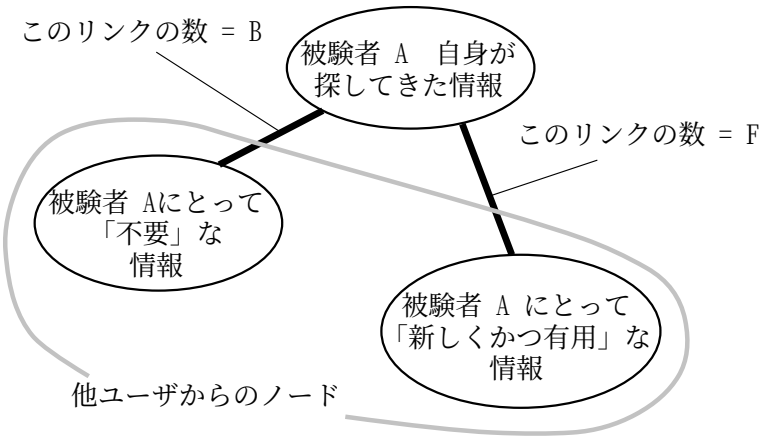

図 5.5: 有用な情報へのリンク数と不要な情報へのリンク数

しかし、各ノードについて、$F > B$ となるか $F < B$ となるかは、一概には言えず、有意な傾向は見出せなかった。その理由を考察するために、被験者に対し聞き取り調査を行ったところ、以下の指摘があった。

- リンクの数が多いので、リンクの意義を掴みづらい。
- リンクの表示が煩雑になり、手軽に利用できない。

この指摘から、現在のところ、自動生成されるリンクの確度が低いものになっているために、リンクの有効性が不明確になっていると考えられる。リンクを、新たな情報を探る時の有効な指針として提供するためには、リンクの自動生成の条件をより厳しくするための手立てが必要であり、その改良よって、セッション毎の有用度が向上するものと考えられる。

第 6 章

おわりに

6.1 まとめ

本研究では、既存の協調的情報フィルタリングシステムの問題点を概括し、さらにコンピュータ・ネットワーク環境のユーザがどのような形で情報獲得活動を行っているのかを考察した上で、ユーザにとって新しくかつ有用な情報の獲得を支援するためのシステムを提案した。そのシステムの設計方針は、個人が保有する情報体系を「情報地図」と呼ぶグラフ形式にて視覚化し、各ユーザの情報地図をシステムが合成することにより、他ユーザの知識を参照可能にすることである。そして、この設計方針に基づき、プロトタイプシステムの実装を行い、評価実験を行った。

本研究の成果を以下にまとめる。

- 以下の特徴を持つシステムを設計し、プロトタイプシステムを実装した。
  - あらゆる興味の領域に対し、そこに興味を持っているユーザが 2 人以上存在すれば協調的フィルタリングが機能する。
  - Netnews、WWW など様々なソースからの情報を一元的に扱える。
  - 各ユーザが保有する情報体系を、情報地図と呼ぶグラフ形式で視覚化する。
  - ユーザ同士の情報地図を合成し、その結果を各ユーザに提示することで、自身にとって未知の情報の意義を示唆する。
- 評価実験では、本研究で提案したシステムの利用により、ユーザ単独で行う情報獲得よりも効率良く新しくかつ有用な情報を獲得できることを確認した。

一方、評価実験により、リンクの有効性について以下の問題点が明らかになった。

- リンクの数が多すぎる場合は、リンク一つ一つの意義が薄れてしまう。
- ユーザインタフェースの不備により、リンクが利用しづらい。

6.2 今後の課題

評価実験によって明らかになった問題点に対処し、以下の改良を進めることで、情報獲得の支援がより一層確実なものになると考えられる。

- より意味のあるキーワードの抽出
  プロトタイプシステムでは、重み付けしたキーワードを基に情報間のリンクを自動生成している。したがって、情報地図を生成する際、英単語の正規化や強力な辞書の使用などによって、より意味のあるキーワードを抽出することは、重要な改良要素である。
  また、この改良により、インタフェース部のテキストエリアに表示するキーワード列の質も向上する。そのテキストエリアは、ユーザが情報の意義を見当付けるための一つの指針として、より有効なものになると考えられる。

- ベクトル空間モデル等の情報間の類似度計算法の導入
  プロトタイプシステムでは、各情報が一定以上の重みがあるキーワードをどれだけ共有しているかによってリンク生成の判定を行っているが、自動リンクの確度を向上させるためには、より効果的なリンク生成アルゴリズムが必要である。例えば、情報検索の分野で広く利用されているベクトル空間モデル [9] を導入することが考えられる。ベクトル空間モデルとは、各文書を多次元空間上のベクトルとして表現し、2 つの文書の類似度を、各ベクトルを比較することによって求めるものである。本システムでは、ベクトルの各次元にはキーワードを、各成分には重みを割り当てることになる。

- 自動リンクにキーワードを付加する等の表示方法の工夫
  プロトタイプシステムでは、関係があると思われる情報間に単にリンクを張るだけであった。そこで、リンクが張られている情報間で共有しているキーワードを、リ

ンク表示の際に付加することにより、リンクの意義をより明らかに示せるかもしれない。

さらに、以下のようなドメインにおけるシステムの有効性を評価したい。そのためには、人数、利用目的、利用期間の各パラメータを変えた運用実験を実施する必要がある。

- メーリングリスト参加者間での情報交換
- Netnews の記事を対象にした情報収集
- ソフトウェアプロジェクト全体またはサブグループにおける情報共有

謝 辞

本研究を進めるにあたり、落水浩一郎教授、篠田陽一助教授からは終始変らぬ御指導を頂きました。心から感謝致します。

海谷治彦助手をはじめ、落水研究室の諸先輩方からは、多大なる助言を頂きました。深く感謝致します。

落水研究室、篠田研究室の方々からは、日頃からさまざまな刺激を頂きました。深く感謝致します。

猪俣敦夫氏、柏瀬秀行氏、高瀬泰宏氏、宇多仁氏には、実験に参加して頂きました。深く感謝致します。

アンケート調査で回答を寄せてくださった学生の方々に感謝の意を表します。

参考文献

[1] Nicholas J. Belkin and W. Bruce Croft. Information filtering and information retrieval: Two sides of the same coin? *Communications of the ACM*, Vol. 35, No. 12, pp. 29–38, Dec. 1992.

[2] Thomas W. Malone, Kenneth R. Grant, Franklyn A. Turbak, Stephen A. Brobst, and Michael D. Cohen. Intelligent information-sharing system. *Communications of the ACM*, Vol. 30, No. 5, pp. 390–402, May. 1987.

[3] Peter W. Foltz and Susan T. Dumais. Personalized information delivery: An analysis of information filtering methods. *Communications of the ACM*, Vol. 35, No. 12, pp. 51–60, Dec. 1992.

[4] 森田昌宏, 篠田陽一. 情報に対する行動の分析と情報フィルタリングへの適用に関 する研究. Master's thesis, 北陸先端科学技術大学院大学, Feb. 1994.

[5] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, Vol. 35, No. 12, pp. 61–70, Dec. 1992.

[6] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. Gropulens: An open architecture for collaborative filtering of netnews. In *Proceeding of the ACM 1994 Conference on Computer Supported Cooperative Work*, pp. 175–186, 1994.

[7] Upendra Shardanand and Pattie Maes. Social information filtering: Algorithms for automating "word of mouth". In *Proceeding of the ACM Conference on Human Factors in Computing Systems, CHI'95*, pp. 210–217, 1995.

[8] David Maltz and Kate Ehrlich. Pointing the way: Active collaborative filtering filtering of netnews. In *Proceeding of the ACM Conference on Human Factors in Computing Systems, CHI'95*, pp. 202–209, 1995.

[9] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexig. *Communications of the ACM*, Vol. 18, No. 11, pp. 613–620, Nov. 1975.

[10] 松本裕治, 北内啓, 山下達雄, 平野善隆, 今一修, 今村友明. 日本語形態素解析システム『茶筌』version 1.5. 奈良先端科学技術大学院大学 松本研究室, Jul. 1997. http://cactus.aist-nara.ac.jp/lab/nlt/chasen.html.

[11] W. B. Frakes and R. Baeza-Yates, editors. *Information retrieval:data strauctures and algorithms*. Prentice Hall, 1992.