エージェントコミュニティを利用したP2P型情報検索の精度評価

古後陽大[†] 峯 恒憲[††] 雨宮真人[††]

階層的なコミュニティの概念とエージェント間のピアツーピアな情報のやり取りを行う仕組みを合わせ持つマルチエージェントシステムを利用して，個人に特化した情報検索機能とコミュニティ所属のメンバーの知識を優先的に利用する仕組みを実現する P2P 型情報検索手法（ACP2P 法），各エージェントが検索のために行った通信履歴（検索履歴）を利用することで，効率のよい検索を実現する．ACP2P 法については，これまでのシミュレーション実験により，ユーザのクエリに関連する情報を検索する際に必要となるエージェント間通信量の削減効果，及び検索効率の向上の効果を確認した．本稿では，ACP2P 法の検索精度について行った評価実験の結果について報告する．

Agent-Community-based Peer-to-Peer Information Retrieval – an Evaluation

Akihiro Kogo,[†] Tsunenori Mine[††] and Makoto Amamiya [††]

The Agent-Community-based Peer-to-Peer Information Retrieval (ACP2P) method[7] uses agent communities to manage and look up information of interest to users. An agent works as a delegate of its user and searches for information that the user wants by communicating with other agents. The communication between agents is carried out in a peer-to-peer computing architecture. Retrieving information relevant to a user query is performed with content files which consist of original and retrieved documents, and two histories: a query/retrieved document history and a query/sender agent history. The ACP2P is implemented using the Multi-Agent Kodama framework.

In this paper, we present the overview of the ACP2P method and discuss the experimental results to illustrate the validity of this approach. The results show that two histories are more useful for reducing communication loads than a naive method employing 'multicast' techniques, and lead to a higher retrieval accuracy than the naive method.

1. はじめに

現在のインターネットユーザは，日々増大する情報の中から自分にとって必要な情報だけを取り出す作業に追われている．そのような作業の手間を省くため，ユーザにとって必要な情報だけを残す「情報フィルタリング」(e.g.[4]) や他のユーザの評価情報を利用してユーザにとって興味ある情報を推薦する「協調フィルタリング」(e.g.[13]) などといった研究が盛んに行われている．しかしこのような研究で開発されるシステムの多くはサーバ・クライアント型のモデルに基づいており，情報の集中制御を行う際に生じるボトルネックに悩まされている．そのため情報の共有機能をピアツーピア (以下 P2P と略記）型のモデルで実現する研究が現在盛んに行われている．しかしそのようなモデルの多くは，各ノードで行われる処理内容は画一的であり単純な内容であることが多い (e.g.[15)2)18)9)]) ．一方，自分の出した query に対する応答記録 (query caching) を利用して検索効率を上げる研究[14] や，分散情報検索やクラスタリングの技術を階層型の P2P ネットワークに適用し，高精度かつ高効率な検索を実現する研究[1)5)6)] もあるが，相手から送られてくる query 情報の利用を試みる研究はこれまでなかった．

このような背景から我々は，エージェントコミュニティを利用した P2P 型情報検索手法（ACP2P 法）を提案してきた[21)8)7)22)]．ACP2P 法では，各エージェントが持つデータの内容に対して検索を行うほか，他のエージェントから受けた検索履歴を基に情報の在処の特定や同じトピックに関心を持つエージェント同士のグルーピングを実現する．これにより，必要な検索結果を得るために行う通信の量を徐々に削減していくことができる．

† 九州大学大学院システム情報科学府
Graduate School of Information Science and Electrical Engineering, Kyushu University
†† 九州大学大学院システム情報科学研究院
Faculty of Information Science and Electrical Engineering, Kyushu University

これまでのシミュレーション実験によって，ACP2P 法が仮定していた「情報の通信量削減」効果と，クエリに多く答えられるエージェントほど自分の求める情報源へのパスを増やすことができ，その結果，検索効率が向上するという「give and take 」効果があることを示した[7)22)]．しかし，これらの実験では検索精度について触れていなかった．そこで本稿では，ACP2P 法の検索精度を測定する実験について述べ，その結果の評価および考察を行う．

以下第 2 節では，ACP2P 法の概要について説明する．第 3 節で実験方法とその結果について述べる．第 4 節で関連研究について議論し，最後にまとめと今後の課題について述べる．

2. エージェントコミュニティを利用した P2P 型情報検索:ACP2P

ACP2P 法では，ユーザ毎にユーザインタフェースエージェント (**UIA**)，情報検索エージェント (**IRA**)，および履歴管理エージェント (**HMA**) の 3 種類のエージェントを 1 組として割り当てる．IRA は自分のユーザが所属するコミュニティ内の他の IRA との対話を中心に，自身のユーザの求める情報の探索を行う．もしそこで見つからない場合には，階層的に辿れる他のコミュニティ所属の IRA との対話を通して情報の探索を行う．

具体的には，IRA は自分のユーザが UIA に対して出したクエリを UIA から受け取ると，そのクエリを HMA に渡し，そのクエリの検索を依頼する他のユーザの IRA（検索対象 IRA と呼ぶ）を見つけさせる．その際 HMA は，**コンテンツファイル**と，**検索結果履歴** (**Q/RDH**) と**クエリ受信履歴** (**Q/SAH**) と呼ぶ 2 つの検索履歴を利用して，検索対象 IRA の検出を行う．その検出方法については，2.2 節で述べる．

ここでコンテンツファイルとは，自身のユーザが作成したドキュメントファイルと，検索により獲得したドキュメントファイルのことである．検索結果履歴は，IRA のユーザ自身が出したクエリと，その関連情報を返してきた IRA のアドレスの対からなる．また，クエリ受信履歴 はクエリと，そのクエリを送ってきた IRA のアドレスの対と，メッセージの受信形式からなる．

コンテンツファイルと検索結果履歴, クエリ受信履歴の形式を表 1 に示す．

ユーザから指定された数（N_R）の検索対象 IRA が見つからなかった場合には，コミュニティ内の全 IRA のアドレスを管理しているポータルエージェント（**PA**）に対してコミュニティ内の全 IRA にそのクエリをマルチキャストするように依頼する．クエリを受け取った IRA は，そのクエリと関連する情報があるか否かを 2.1 節の方法で調べ，その検索結果をクエリを送ってきた IRA（もしくは PA）に返す．コミュニティ内で N_R 個の IRA が見つからなかった場合，PA は，一つ上位の PA にマルチキャスト依頼を行う．

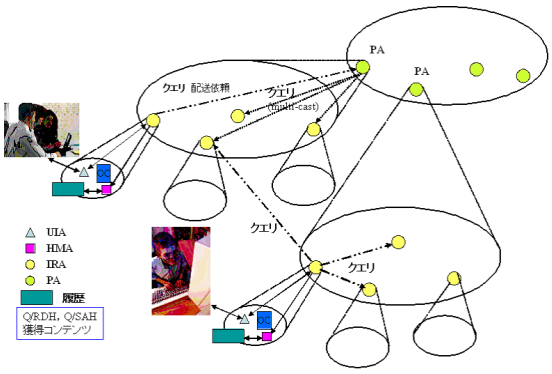


図 1 エージェントコミュニティの構造
Fig. 1 Example of agent community structure.

図 1 は，ACP2P 法が仮定しているエージェントコミュニティ構造の例を示している．PA は，コミュニティの代表であり，先に述べたようにコミュニティ内の全 IRA のアドレスを管理している．また PA は，上位のコミュニティのメンバエージェントでもあり，コミュニティが 1 エージェントとして扱われることにより，コミュニティの階層構造を実現する．本研究では ACP2P 法を Kodama[20)] を利用して実装した．Kodama の PA は，コミュニティ内のエージェントのアドレスのみを管理しているにすぎず，コミュニティ内のエージェントのコンテンツの管理等は行わない．

2.1 クエリとコンテンツとの類似度の計算方法

エージェントが他のエージェントからクエリ Q を受けた際，Q に関連するドキュメント D を求めるため，Q と D の類似度計算を行うが，それは情報検索において実績のある BM25[12)] を修正した式（1）において，$dl/avdl$ を 1 と近似した式によって算出する．

$$Sim_d(Q,D) = \sum_{T \in Q} w^{(1)} \frac{2tf}{\frac{dl}{avdl} + tf} \qquad (1)$$

ここで，T は Q に含まれる単語である．tf は D に含まれる T の数である．dl は D のドキュメント長（D に含まれる単語の数）である．$avdl$ は平均ドキュメント長である．$w^{(1)}$ は以下の式で表される T の重みである．

表 1 ACP2P 法で使用するコンテンツ，検索結果履歴，クエリ受信履歴の各ファイル
Table 1 Content file and two histories: Q/RDH and Q/SAH.

コンテンツファイル	id	コンテンツの識別子
	title	コンテンツのタイトル
	body	コンテンツ本文
	original	このコンテンツを最初に作成・発信した検索エージェントのアドレス
	range	流通範囲 (ALL, Community, Agent)
検索結果履歴	query	送信したクエリ
	from	この検索結果を返信した検索エージェントのアドレス
	contents	検索により取得したコンテンツ (上の段のコンテンツの形式に従う)
クエリ受信履歴	query	受信したクエリ
	from	このクエリを送信してきた IRA のアドレス
	attribute	クエリを受信した形式

$$w^{(1)} = \log \frac{N - n + 0.5}{n + 0.5} \tag{2}$$

ここで，N は各 IRA がコンテンツとして保持する全ドキュメント数である．n は，N 個のドキュメントのうち T を含むドキュメントの数である．

$Sim_d(Q, D)$ の値が 0 より大きい値となる D を Q と関連のあるドキュメントと判断する．

2.2 クエリと検索対象 IRA との間の類似度計算

クエリ Q と，IRA $agent_j$ $(j = 1 \cdots M)$ との類似度計算式 $Score(Q, agent_j)$ を，式 (3) に定義する．ただし M は，履歴中に登録されている IRA 数である．

$$\begin{aligned} Score(Q, agent_j) = & \\ & \sum_{i=1}^{k} \cos(Q, \vec{qh_{d_i}}) \\ & + \sum_{i=1}^{m} (\cos(Q, \vec{qh_{sa_i}}) + \varphi(i)) \\ & + \max_{1 \le i \le n} Sim_d(Q, doc_i) \end{aligned} \tag{3}$$

$$\varphi(i) = \begin{cases} \delta & qh_{sa_i} \text{が他の IRA から直接送られた場合} \\ 0 & \text{それ以外（PA から送られてきた場合）} \end{cases}$$

ここで，第 1 項は Q と $agent_j$ に出した k 個のクエリ qh_d とのスコア値であり，第 2 項は Q と $agent_j$ が送ってきた m 個のクエリ qh_{sa} とのスコア値である．また，$\varphi(i)$ は qh_{sa_i} が PA を経由せず他の IRA から直接送られた場合の重みであり，本実験では $\delta = 0.1$ とする．そして第 3 項は，$agent_j$ がコンテンツの *original* フィールドに登録されている n 個のドキュメントとのスコア値である．$Sim_d(Q, doc)$ は，クエリ Q と検索対象ファイル doc との類似度値を計算する式であり，2.1 節で述べた BM25 の簡易式の中の $\frac{dl}{avdl}$ を $\frac{dl}{avdl} = 1$ に更に簡略化した式を使用する．

検索対象 IRA(query の送り先の IRA) は，$Score(Q, agent_j)$ の上位から N_R 個を取り出した $agent_j$ とする．

3. 実　　験

文献[7)22)] では，ACP2P 法が仮定していた「通信量の削減」と「検索効率の向上」の効果があることを示した．また，クエリ受信履歴の効果から，IRA 間の「give and take」効果があること，および仮想的なエージェントコミュニティの創出についても効果があることを示した．しかしこれらでは，クエリと検索ドキュメント間の類似度式について単純な検索式を用いていたため，検索精度については何も述べていなかった．そこで本実験では，クエリと検索ドキュメント間の類似度計算式として情報検索で良く利用されている BM25[12)] の簡易式を利用した上で，通信量の削減効果と検索精度の向上効果について調べる．

3.1 準　　備

本実験でも，文献[7)22)] と同様に検索に利用するデータとして，Yahoo! JAPAN[19)] に登録されている Web サイトのコンテンツを利用した．Yahoo! JAPAN では，登録された Web サイトがカテゴリ分けされているが，ここでは,「動物」,「スポーツ」,「コンピュータ」,「医療」,「金融」の 5 つのカテゴリを使用した．実験では各カテゴリから登録数の多い順に 20 個（計 100 個）のサブカテゴリを選んで利用した．各々のサブカテゴリを仮想的なユーザと見なし，それぞれに 1 つの IRA を割り当てた．またそのカテゴリに登録されているサイトから収集した Web ページを，IRA のコンテンツ（ユーザの保持する情報）として利用した．次に IRA が利用するクエリとして，長さ 1 のもの（$QL = 1$）と長さ 2（$QL = 2$）の 2 つのセットを各 IRA 毎に用意した．各セットのクエリを作成するために，IRA に割り当てられたコンテンツから出現頻度の高い名詞 N 個 ($QL = 1$ の場合 10 個，$QL = 2$ の場合 5 個) を自動抽出した．$QL = 1$ の場合，抽出された各名詞をク

エリとし，$QL = 2$ の場合は，抽出された 5 個の名詞から 2 個を選択し，その組み合わせにより計 10 個のクエリを作成した．クエリの送信は，各 IRA とも出現頻度の高いものから順に送信する．

次の三つの手法について比較実験を行った．

(1) 検索対象 IRA を見つけるために検索結果履歴 とクエリ受信履歴の両方の履歴を利用し，N_R 個の検索対象 IRA を検出し検索を依頼する手法．これを**両履歴利用法** と呼ぶ．

(2) 検索対象 IRA を見つけるために検索結果履歴のみを利用し，N_R 個の検索対象 IRA を検出し検索を依頼する手法．これを**検索結果履歴利用法**と呼ぶ．

(3) 両方の履歴を利用せず，常に PA にマルチキャストを依頼する手法．**マルチキャスト法**と呼ぶ．
マルチキャストを PA に依頼する場合，PA はある一定時間内に早く返されたものから順に N_R 個選ぶ．(1)，(2) で N_R 個の IRA が見つからない場合には，PA にマルチキャストを依頼する．PA はコミュニティ内で N_R 個の IRA が見つからなかった場合，本来は更に上位の PA にマルチキャストを依頼するが，本実験では，一個のフラットなコミュニティのみを用いているため，そのまま見つかっただけの IRA のリストを返す．

3.2 検索精度の測定方法

P2P ネットワーク上で，各 Peer からすべてのコンテンツを集めて，それらをインデックス化するのは極めて困難である．そこで理想となる，すべてのドキュメントをインデックス化して行う従来の検索手法 (SDB(Single DB) 法と呼ぶ）と ACP2P 法とで検索結果を比較し，その類似度を ACP2P 法の近似的な検索精度の指標とした．本稿では，SDB 法として 2.1 節の式（1）を各エージェントの保持するすべてのコンテンツに適用する確率型の検索モデルを使用した．

ACP2P 法と SDB 法の検索精度を比較する指標として，以下の計算式を使用した．

$$\sum_{i=1}^{N_R} \frac{1}{r(i)} / \sum_{i=1}^{N_R} \frac{1}{i} \qquad (4)$$

ここで $r(i)$ は，ACP2P 法でクエリとの関連性が i 番目に高いと判断されたドキュメントの SDB 法における順位を表す．例えば，あるクエリとの類似度が，ACP2P 法において 3 番目であり，SDB 法においては 5 番目だった場合，$r(3) = 5$ ということになる．式（4）によって求められる値を **RRS(Reciprocal Rank Similarity)** と呼ぶことにし，両手法のドキュメントのランキング間の類似度を計算する．RRS 値は，N_R 個の中にランク上位のドキュメントが含まれているほど大きくなる．

例として，$N_R = 3$ の下で，ACP2P 法でのランキング上位 3 個のドキュメントが，SDB 法でそれぞれ 3，5，1 というランキングだったとすると，RRS 値は $\frac{1/3+1/5+1/1}{1/1+1/2+1/3} = 0.84$ と計算される．またランク 1 位が含まれず，SDB 法で 3, 5, 2 が返されたとすると，$RRS = 0.67$ と小さくなる☆．

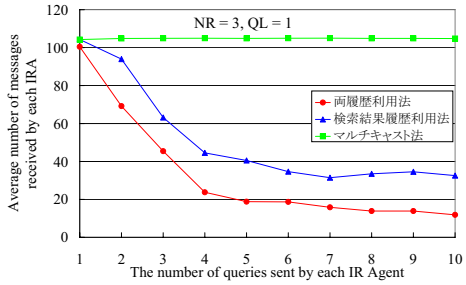
本実験では，RRS を全エージェントで平均したもの，すなわち $\frac{1}{N}\sum_i^N RRS(i)$ を比較指標として使用する．ここで，N は全エージェント数，$RRS(i)$ は i 番目のエージェントの RRS 値を表す．

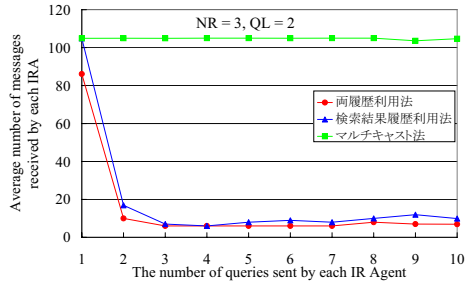
3.3 実験結果

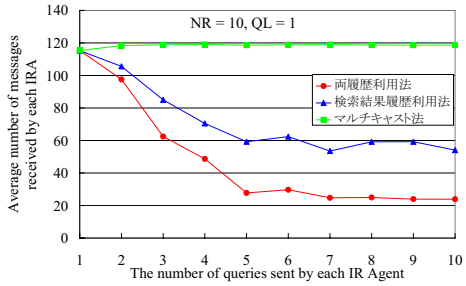
今回は，クエリとドキュメントとの類似度計算式を変えたため，これまでの実験との比較の意味で，1 つのコミュニティを利用した場合についてメッセージ数の変化と RRS 値の変化について調べた．その結果を図 2,3 に示す．文献[22] で述べたのと同様に，両履歴利用法と検索結果履歴利用法では $QL = 1$ と $QL = 2$ のどちらの場合でも，クエリが投入されるに従って平均メッセージ数は減少していく．また検索結果履歴利用法では，N_R が大きくなるとグラフがマルチキャスト法のものに近づくことが分かる．この結果から，クエリ受信履歴がクエリに関連するエージェントの発見に有効であるということができる．また，図 2,3 より，$QL = 2$ の場合のほうが $QL = 1$ の場合よりも受信メッセージ数の減少が速いことが分かる．これはクエリにヒットするドキュメントの数が $QL = 2$ の場合は，$QL = 1$ の場合よりも多くなることで，履歴の効果が初期の検索の段階でも現れやすいからだと思われる．

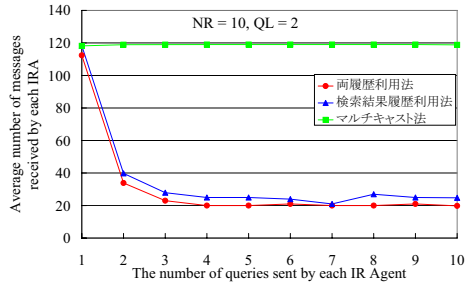
次に，同じ条件の下で，3 つの手法の RRS の比較を行った．その結果を図 4,5 に示す．まず，QL の値に関わらず N_R が大きくなると，RRS 値は増加しその増加度は緩やかになっている．最大値は $QL = 2$ の場合でも 0.8 程度である．これは出現頻度の高いクエリから順番に送信していることにより，そのクエリにヒットするドキュメントの数が多くなり，履歴の効果が早く現れているためだと考えられる．またクエリ送信回数が増えるにつれてマルチキャスト法の RRS 値が大きくなるのは，検索を重ねることでコンテンツが

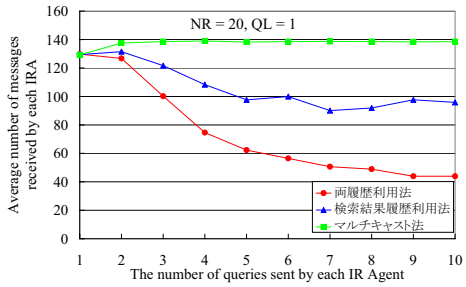
☆ RRS と類似した評価手法として，情報検索や情報抽出の評価手法としてよく利用される平均逆順位 (MRR : Mean Reciprocal Rank) がある．MRR では，正解の逆順位の総和を正解数で割るが，SDB 法との類似性という観点から考えて，RRS を定義した．











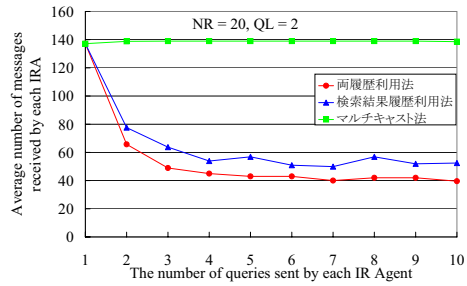


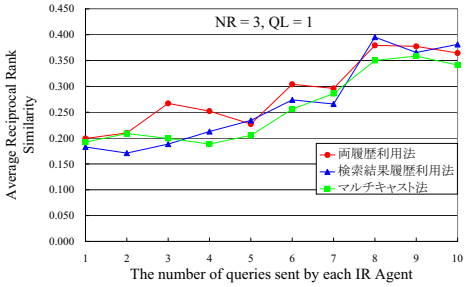
図 2 各方式における平均受信メッセージ数の比較（$QL = 1$，上から $N_R = 3$，$N_R = 10$，$N_R = 20$）

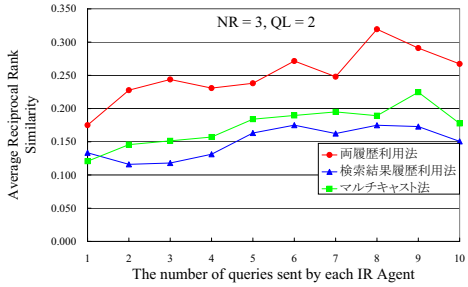
Fig. 2 The comparison of average number of messages received by each IR agent for every query input, using 3 diifferent N_R values : $N_R = 3$ (TOP), $N_R = 10$ (MID) and $N_R = 20$ (BTM) and QL=1

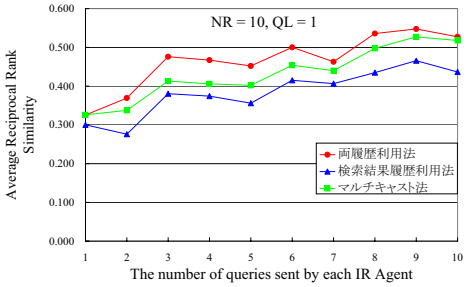
図 3 各方式における平均受信メッセージ数の比較（$QL = 2$，上から $N_R = 3$，$N_R = 10$，$N_R = 20$）

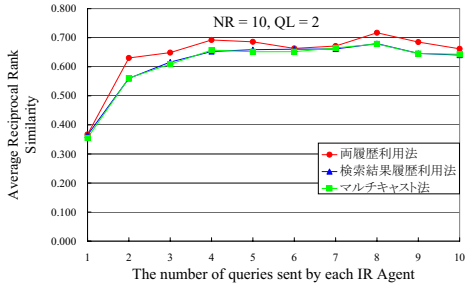
Fig. 3 The comparison of average number of messages received by each IR agent for every query input, using 3 diifferent N_R values : $N_R = 3$ (TOP), $N_R = 10$ (MID) and $N_R = 20$ (BTM) and QL=2

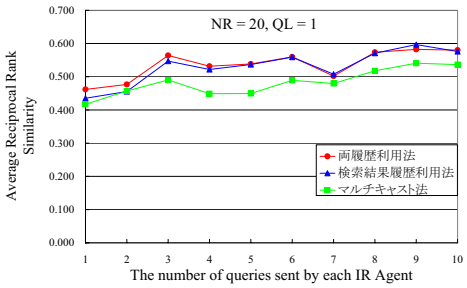
コミュニティ内のエージェント間に分散し，クエリとの類似度の高いドキュメントが検索結果として返される確率が上昇するためだと考えられる．次に $QL = 1$ の場合，両履歴利用法の RRS 値が他の 2 手法よりも高い数値を示しているものの，その値がそれほど高くない．この原因は，初期の検索の段階で蓄積されるコンテンツファイルおよび検索履歴は，PA がマルチキャストを行い早く返された順に N_R 個選んだことにより得られるものであるためであり，それゆえ検索回数を重ねても，SDB 法との類似度があまり上昇しなくなると考えられる．さらに，$QL = 1$ で $N_R = 3$ の時，3 つの手法の間で特に差が見られないのは，検索で獲得されるドキュメント数が少ないために履歴の効果が現れにくいためだと考えられる．$N_R = 20$ の時でも同











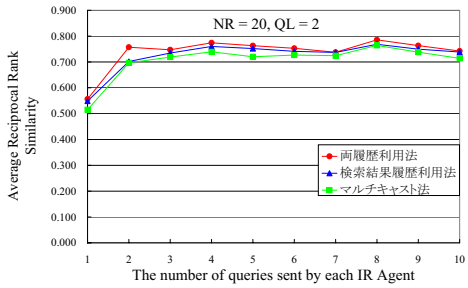


図 4 各方式における平均 RRS 値の比較（$QL = 1$，上から $N_R = 3$，$N_R = 10$，$N_R = 20$）

Fig. 4 The comparison of average number of messages received by each IR agent for every query input, using 3 diifferent N_R values : $N_R = 3$ (TOP), $N_R = 10$ (MID) and $N_R = 20$ (BTM) and QL=1

図 5 各方式における平均 RRS 値の比較（$QL = 2$，上から $N_R = 3$，$N_R = 10$，$N_R = 20$）

Fig. 5 The comparison of average number of messages received by each IR agent for every query input, using 3 diifferent N_R values : $N_R = 3$ (TOP), $N_R = 10$ (MID) and $N_R = 20$ (BTM) and QL=2

様なのは，早い段階でドキュメントがネットワーク全体に分散し，エージェント間で保有するドキュメントにあまり違いが生じなくなるため，履歴を使用してもマルチキャストと同程度の精度にしかならないためだと考えられる．また，$N_R = 3$ の場合を除き，$QL = 2$ では，$QL = 1$ の時よりも 3 手法共に高い RRS 値を示している．全体として QL, N_R の値に関わらず，両履歴利用法は他の手法よりも高い RRS 値を記録していることから，クエリ受信履歴は検索精度の向上にも効果があると言える．さらに，式（1）によって求められるクエリとコンテンツとの類似度の値は，各 IRA の持つドキュメントの個数である N および n に左右されるが，これらはネットワーク全体から集めた SDB 法で利用する N や n と比較すると大きく異なってい

るため，ACP2P 法と SDB 法のランキングに異なる結果が出やすくなり，RRS 値は 1 よりも低い値を示す原因となっていると考えられる．

4. 関 連 研 究

P2P 型のファイル検索システム (ファイル共有システム) としては，Gnutella[18]，Kazaa[17]，CAN[2]，Chord[15]，pSearch[16] など，多数が提案されている．CAN や Chord，pSearch が利用している DHT (Distributed Hash Table) は，ユーザの検索要求と検索対象との効率的なマッピングを提供する．しかし，これらの構造型 P2P ネットワークでは，動的で多様な P2P ネットワークにどれだけ対応できるかは不明であり[10]，柔軟な検索やメッセージ交換を行うには，Gnutella や Kazaa のようなメッセージパッシング型の方が望ましい[5]．ACP2P 法も，Gnutella v0.6 や Kazaa のような階層型の P2P ネットワーク構造を仮定するが，ACP2P 法では各エージェントの通信は，他のコミュニティのエージェントと，直接，ピアツーピアで通信を行う．

コンテンツ情報を利用した P2P 型の検索に Lu ら[5][6] や Bawa ら[1] の研究がある．Lu らは，Gnutella のような階層型の P2P ネットワーク上で，各 Peer の持つコンテンツを予めクラスタリングし，各クラスタ内の葉ノードにあたる Peer に対する directory 機能をもつ Directory Peer (DP) 間の routing を，隣接する DP のコンテンツ情報を基に行うことで，効率の良い高精度な検索ができることをシミュレーションにより示している．Bawa らは同様にコンテンツをトピック毎にクラスタリングし，ハブ間や，クラスター内での routing を決めることで，効率良い検索ができることを，再現率を基準に，様々な条件でシミュレーションで示している．いずれも，構造型 P2P ネットワークと同様に，全てのピアの情報源記述 (resource description) が利用できることを仮定している．

本稿での ACP2P 法を用いた実験では，検索過程を通してのみ各エージェントがもつ情報にアクセスできるという条件の下で，徐々に獲得していったコンテンツや検索履歴を利用した際の検索精度について議論している．今回利用した検索要求履歴 (Q/RDH) は，Sripanidkulchai ら[14] の Short Cut と同様なものである．Sripanidkulchai らは，それを利用することで効率の良い検索が可能となることを実験により示し，また協調フィルタリングと同様な効果が得られることを指摘している[14] が，クエリ受信履歴 (Q/SAH) のような，相手側から来る検索要求情報の利用については触れられていない．

5. お わ り に

本稿では，エージェントコミュニティを利用した P2P 型情報検索手法（ACP2P 法）の検索精度と検索に必要となるメッセージ数の変化についての実験結果について報告した．クエリと検索ドキュメント間の類似度計算では，情報検索で良く利用される BM25[12] を利用した．実験の結果，相手から送られてくるクエリ情報を利用することで，自分の出した検索結果履歴を利用するだけの場合よりも，少ないメッセージ数で，しかも高い検索精度を得られることが分かった．

しかしながら，集中型のドキュメントデータベースを利用した SDB 法との比較値である RRS 値は，十分に高いものではなかった．その理由として，PA によるマルチキャストの際に，返された順に検索対象 IRA を選択したこと，それによって蓄積されたコンテンツファイルおよび検索履歴が，次回の検索に悪影響をしていることが考えられる．つまり PA は検索依頼に対して，単純に YES と返答してきた順にエージェントをアドレスリストに追加するので，このままではこれ以上の精度は期待できない．そこで，現在，検索結果のクエリとの類似度について何も設けていない閾値について考慮し，関連性の低い検索結果を排除することや，PA に直接依頼する代りに，検索を依頼したエージェントに対して，そのエージェントが知っている検索要求と関連性の強いエージェントに検索要求を転送してもらうことなどを検討している．後者を利用することで，Routing Indices[3] のような効果が期待できる．また，各 IRA の持つドキュメント数とネットワーク全体のドキュメント数との大きな差異がクエリとコンテンツの類似度の計算に影響を及ぼし，SDB 法と異なるランキングになることも RRS 値が低くなる原因と考える．この解決策として，ネットワーク全体のドキュメント数を推測する方法を現在検討中である．

なお，本稿では報告できなかったが，現在，複数のコミュニティを利用した場合との比較実験も行っており，これについては，近々報告する予定である．

今後の課題としては，各ピアがコミュニティの入退出を行った場合での実験[11]，更には各ピアが持つコンテンツ内容が変化する場合での実験を行うことが上げられる．これらの実験を行う際には，実験のデータ規模を大きくして，拡張性についても調査する予定である．その他の課題として，ユーザの検索結果に対する評価情報を次回の検索に利用する手法の検討などが挙げられる．

謝辞 本研究の一部は，科学研究費 基盤研究 (C)(2)（課題番号 16500082）ならびに戦略的情報通信研究開発推進制度 SCOPE-C （課題番号 052310008）の支援を受けて行われた．

参 考 文 献

1) Mayank Bawa, Gurmeet Singh Manku, and Prabhakar Raghavan. Sets: search enhanced by topic segmentation. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 306 – 313, 2003.
2) Ian Clarke, Oskar Sandberg, Brandon Wiley, and Theodore W. Hong. Freenet: A distributed anonymous information storage and retrieval system. *Designing Privacy Enhancing Technologies: International Workshop on Design Issues in Anonymity and Unobservability, http://www.doc.ic.ac.uk/˜twh1/academic/*, 2001.
3) Arturo Crespo and Hector Garcia-Molina. Routing indices for peer-to-peer systems. In *the 28th International Conference on Distributed Computing Systems*, 7 2002.
4) Ken Lang. NewsWeeder: learning to filter netnews. In *Proceedings of the 12th International Conference on Machine Learning*, pages 331–339. Morgan Kaufmann publishers Inc.: San Mateo, CA, USA, 1995.
5) Jie Lu and Jamie Callan. Content-based retrieval in hybrid peer-to-peer networks. In *Proceedings of the twelfth international conference on Information and knowledge management*, pages 199–206, 2003.
6) Jie Lu and Jamie Callan. Federated search of text-based digital libraries in hierarchical peer-to-peer networks. In *Proceedings of the Twenty-Seventh European Conference on Information Retrieval Research (ECIR'05)*, 2005.
7) Tsunenori Mine, Daisuke Matsuno, Akihiro Kogo, and Makoto Amamiya. Design and implementation of agent community based peer-to-peer information retrieval method. In *CIA 2004, LNAI 3191*, pages 31–46, 9 2004.
8) Tsunenori Mine, Daisuke Matsuno, Koichiro Takaki, and Makoto Amamiya. Agent community based peer-to-peer information retrieval. In *the third international joint conference on Autonomous Agents and Multi Agent Systems (AAMAS)*, pages 1484–1485, 7 2004. poster.
9) Napster. http://www.napster.com/, 2000.
10) Sylvia Ratnasamy, Scott Shenker, and Ion Stoica. Routing algorithms for dhts: Some open questions. In *First International Workshop on Peer-to-Peer Systems (IPTPS)*, 2002.
11) M. Elena Renda and Jamie Callan. The robustness of content-based search in hierarchical peer to peer networks. In *Proceedings of the thirteenth ACM conference on Information and knowledge management*, pages 562–570, 2004.
12) S. E. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford. Okapi/keenbow at trec-8. In *NIST Special Publication 500-246: The Eighth Text REtrieval Conference (TREC-8)*, pages 151–162, 1999.
13) Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *WWW10*, pages 285–295, 2001.
14) Kunwadee Sripanidkulchai, Bruce Maggs, and Hui Zhang. Efficient content location using interest-based locality in peer-to-peer systems. In *IEEE INFOCOM 2003*, 2003.
15) Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, and Hari Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *Proceedings of the 2001 conference on applications, technologies, architectures, and protocols for computer communications*, pages 149–160, 2001.
16) Chunqiang Tang, Zhichen Xu, and Sandhya Dwarkadas. Peer-to-peer information retrieval using self-organizing semantic overlay networks. In *SIGCOMM*, 2003.
17) Kazaa v3.0. http://www.kazaa.com/.
18) Gnutella Protocol Development v6.0. http://rfc-gnutella.sourceforge.net/.
19) Yahoo. http://www.yahoo.co.jp/, 2003.
20) Guoqiang Zhong, Satoshi Amamiya, Ken'ichi Takahashi, Tsunenori Mine, and Makoto Amamiya. The design and application of kodama system. *IEICE Transactions INF.& SYST.*, E85-D(04):637–646, 4 2002.
21) 峯 恒憲, 松野 大輔, and 雨宮 真人. エージェントコミュニティを利用した P2P 型情報検索. **人工知能学会論文誌** *J-STAGE http://tjsai.jstage.jst.go.jp/ja/*, 19(5):421–428, 2004.
22) 峯 恒憲, 古後 陽大, and 雨宮 真人. エージェントコミュニティを利用した P2P 型情報検索とその評価. **電子情報通信学会：ソフトウェアエージェントとその応用特集号**, J88-D-I(9):1418–1427, 9 2005.