

ユーザフィードバックとクエリ学習を利用した P2P 型情報検索の複数コミュニティ上での評価

小林 寛 武[†] 峯 恒 憲^{††}

分散検索を行う上で、ユーザフィードバックとコミュニティ内でのクエリ学習の利用は、コミュニティ全体の検索精度の向上と通信負荷の削減に有効である。本稿では、複数コミュニティ環境におけるユーザフィードバックとクエリ学習の利用法について議論する。特に、クエリ学習において使用した履歴を用いることにより、複数コミュニティ環境における通信量の問題を解決する手法を提案し、検索精度の改善にもつながることを実験結果を通して論じる。

The ACP2P Method using User Feedbacks and Query Learning in Multiple Communities - An Evaluation

HIROTAKE KOBAYASHI[†] and TSUNENORI MINE^{††}

This paper proposes refinement methods of the ACP2P using user feedbacks and query-learning methods. The proposed methods are effective in increasing retrieval accuracy in communities and for decreasing communication loads required for documents retrieval. The paper especially focuses on user feedbacks and query learning in multiple communities. The experimental results illustrated the validity of our proposed methods.

1. はじめに

インターネットの普及は、ユーザに、多くの有益な情報とサービスをもたらしている。特に検索サービスは、有益な情報源を特定するために日常的に利用されている欠かすことのできないサービスの一つである。この検索サービスは、情報を集中的に管理する方式(集中型の検索方式)で実現されてきた。集中型の検索方式では、検索対象の情報量とユーザのアクセス頻度の増加に対応するため、サービス提供者はシステム自体を分散化し、スケーラビリティや耐故障性の問題に対処してきたが、分散化の代償として、システムの維持・管理に多大なコストの支払を余儀なくされている。

これに対し、Peer-to-Peer (P2P) ネットワークを用いることで、膨大な情報と維持・管理コストを分散化する研究が提案されている(例えば^{3),4)}。しかし、初期に提案された構造化されていない P2P ネットワーク(以下、非構造化 P2P ネットワーク)は、検索の自由度はあるものの、非効率的でネットワーク全体の通

信量が増加するといった問題があった。これを改善するため、構造化された P2P ネットワーク(以下、構造化 P2P ネットワーク)を利用する研究が提案されている(例えば^{8),10)}が、構造化 P2P ネットワークを利用した検索は、その構造的な特性から、ほとんどが完全一致型の単語や句単位の検索に限られている。そのため、ドキュメントの内容を踏まえ、検索要求(クエリ)に対する関連度順序を考慮した検索には、主に非構造化 P2P ネットワークが利用されている(例えば^{5),6),11)}。これらの研究では、ネットワーク上のノード(Peer)の持つドキュメントの内容に応じて Peer のクラスタリングや、どのクラスタ(もしくは Peer 単体)にクエリを投げるかを決定するクエリの配送経路制御及びその学習手法、検索対象情報の複製の作成法などが研究されている。しかしながら、これらの研究で採用されているモデルは、検索過程で他の Peer から得た情報の再配布を考慮しないモデルであったり、応答する Peer の意思に関係なく、検索で利用された経路上の Peer に複製を置くなど、ユーザが所属するコミュニティ内での身近な情報のやり取りを考慮したモデルではない。

一方、我々が提案してきた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)^{7),13)}

[†] 九州大学工学部電気情報工学科
Faculty of Engineering, Kyushu University

^{††} 九州大学大学院システム情報科学研究所
Faculty of ISEE, Kyushu University

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	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)
	reputation	コンテンツに対する評価
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 IR agent のアドレス
	attribute	クエリを受信した形式

は、ユーザ間での身近な利用を想定し、各ユーザに 1 つのエージェントを割り当て、各エージェントが互いにクエリとその関連コンテンツのやり取りを行うモデルを採用している。ACP2P 法でのエージェント間ネットワークは、非構造化 P2P ネットワークに対応すると考えられる。ACP2P 法では、各エージェントが持つデータとクエリの類似度により検索を行うほか、他のエージェントとの検索履歴やクエリ受信履歴を基に情報の所在を特定する。しかし、クエリとドキュメントとの類似度は、クエリ内の単語の含有率に応じて決められるものであり、クエリが意図する意味までは考慮していない。そのため類似度が高いからといって、必ずしもそのドキュメントがユーザの検索意図を満たすとは限らない。そこで、コミュニティ全体の検索精度の向上と通信負荷の軽減を行うために、検索結果のドキュメントに対してユーザから評価を受け、それをフィードバックとしてクエリとドキュメントの類似度計算に利用する手法や、マルチキャスト時にクエリとその送信元エージェント、送信先エージェントなどの情報を PA の履歴に蓄え利用するクエリ送信先学習機能を提案した¹²⁾。

これまでの実験では単一のコミュニティに限った実験のみを行ってきた。しかし、もともと ACP2P 法はユーザ間での身近な利用を想定し、あるコミュニティに属するユーザが同じコミュニティに属する別のユーザと情報を容易に共有できるように考案された。実社会において、ユーザが属するコミュニティは無数にあり、またユーザが単一のコミュニティにのみ属することはほとんどない。そこで本稿では、複数コミュニティ環境を想定したシミュレーション実験を行う。その際、クエリ送信先学習機能を応用した通信量の削減手法を提案する。また、ユーザからのフィードバックの利用法として、不適合と評価された情報を活用する手法を複数コミュニティへ適応した結果について報告する。

以下、2 で ACP2P 法の概要を説明し、3 でユーザからのフィードバックを利用する手法について述べる。4 でマルチキャスト時のクエリの送信先を学習する手法および、複数コミュニティへの応用法の詳細について述べる。5 でそれぞれの手法の有効性を検証するためのシミュレーション実験とその結果について議論し、6 で本論文のまとめと今後の課題を述べる。

2. ACP2P 法の概要

ACP2P 法ではユーザ毎に検索エージェント (Information Retrieval (IR) agent) を割り当てる。IR agent は自分のユーザが所属するコミュニティ内の他の IR agent 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 IR agent との対話を通して探索する。ACP2P 法では、各コミュニティにそのコミュニティを代表するポータルエージェント (Portal Agent (PA)) の存在を仮定している。PA はコミュニティ内の全 IR agent のアドレスを管理する役目を担う IR agent であり、かつ上位コミュニティの IR agent でもある。PA の存在により、コミュニティを 1 IR agent として扱うことができ、これによりコミュニティの階層構造を実現することができる。

IR agent は自身のユーザからクエリを受け取ると、そのクエリの検索を依頼する他のユーザの IR agent (検索対象エージェント) を見つけるため、コンテンツと検索結果履歴 (Q/RDH)、クエリ受信履歴 (Q/SAH) を利用する。検索対象エージェントの決定方法については、2.2 で述べる。ここでコンテンツとは自身のユーザが作成したドキュメントと検索により獲得したドキュメントのことである。また Q/RDH は自身が送信したクエリとその検索結果についての情報を保持し、Q/SAH は自身が他の IR agent から受信したクエリについての情報を保持する。コンテンツと Q/RDH、

Q/SAH の形式を表 1 に示す。

ユーザが指定した検索要求数 (N_R) の検索対象エージェントは、自身の履歴から探すか、PA にコミュニティ内の IR agent へのマルチキャストを依頼することで探す。マルチキャスト依頼を受けた PA は、コミュニティ内の全 IR agent にクエリを送信する。PA からクエリを受け取った IR agent は、そのクエリに関する情報を持っている (YES) か否か (NO) と、持っている場合はクエリと最も関連度の高いコンテンツのスコア (類似度) および関連コンテンツ数を PA に返答する。IR agent からの返答を受けた PA は、YES と答えた N_R 個の IR agent のリストをマルチキャスト依頼してきた IR agent に送信する。このようにして見つけた検索対象エージェントに対して、クエリを送信し、それに対する検索結果を受け取る。他の IR agent からクエリを受信した IR agent は、ある閾値 δ よりも高い類似度のコンテンツのみを検索結果として返す。IR agent がクエリに関連する情報を持っているか否かは 2.1 の方法で調べる。複数のコミュニティが階層化されて存在している場合、必要に応じて**PA は一つ上位の PA にマルチキャスト依頼を行う。

2.1 クエリとコンテンツとの類似度計算

IR agent が他の IR agent からクエリ Q を受けた際、 Q に関連するドキュメント D を求めるために BM25⁹⁾ に基づいた式 (1) によって Q と D の類似度計算を行う。

$$Sim_d(Q, D) = \sum_{T \in Q} w \frac{(k_1 + 1)tf}{K + tf} \quad (1)$$

ここで T は Q に含まれる単語であり、 tf は D に含まれる T の数である。 $K = k_1((1-b) + b \frac{dl}{avdl})$ であり、 K 中の dl , $avdl$ はそれぞれ D のドキュメント長 (D に含まれる単語の数)、 Q を受け取った IR agent がコンテンツとして保持する全ドキュメントに対する平均ドキュメント長である。 k_1, b は、利用するドキュメント集合によって適切な値を設定する定数であり、後述の実験では予備実験の結果から $k_1 = 2.5, b = 0.85$ としている。 w は以下の式で表される T の重みである。

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

ここで N は各 IR agent 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。 $Sim_d(Q, D)$ の値がある閾値を超えた値 δ 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 検索対象エージェントの決定

履歴から検索対象エージェントの決定には式 (3) に示す、クエリ Q に関する IR agent $agent_j$ ($j = 1 \dots n$) のスコア $Score(Q, agent_j)$ を使用する。ここで n は履歴中に登録されている IR agent 数である。

$$Score(Q, agent_j) = \sum_{i=1}^k \cos(Q, Q_{RDH_i}) + \sum_{i=1}^l (\cos(Q, Q_{SAH_i}) + \varphi(i)) + \max_{1 \leq i \leq m} Sim_d(Q, doc_i) \quad (3)$$

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

第 1 項は $agent_j$ に送信した k 個のクエリ Q_{RDH_i} ($i = 1, \dots, k$) と Q とのスコア値であり、第 2 項は $agent_j$ から送信された l 個のクエリ Q_{SAH_i} ($i = 1, \dots, l$) と Q とのスコア値である。また、 $\varphi(i)$ は Q_{SAH_i} が PA を経由せず他の IR agent から直接送られた場合の重みであり、これまでの実験と同様に $\delta = 0.1$ とする。第 3 項は、コンテンツの *original* フィールドが $agent_j$ である m 個のドキュメントと Q の類似度の最大値である。 $Sim_d(Q, doc_i)$ は Q とドキュメント doc_i との類似度であり、式 (1) を利用して計算される。その際、式 (1) 中の N の値として、クエリ送信 IR agent の持つドキュメント数を使う $\ast 4$ 。検索対象エージェントは、 $Score(Q, agent_j)$ の上位から N_R 個を取り出した $agent_j$ とする。

3. ユーザフィードバックの利用

3.1 ユーザからのフィードバックの取り込み

ユーザは IR agent が示した検索結果に対して、そのドキュメントが検索要求を満たしているか否かを判断し、評価を与える (図 1 参照)。IR agent はユーザの評価を Q/RDH に蓄える $\ast 5$ 。

\ast 後述の実験では、クエリ送信 IR agent が持つ関連コンテンツのうち上位から 10 番目の類似度を閾値とした。

$\ast \ast$ たとえばコミュニティ内で N_R 個の IR agent が見つからなかった場合や、他のコミュニティに関連情報が分散していることがわかっている場合。

$\ast \ast \ast$ 今回は閾値を 0 とした。

$\ast 4$ クエリ送信 IR agent は、クエリを送信する際に、自分の持つドキュメント数などを一緒に送る。

$\ast 5$ ユーザが陽に評価値を与える手間を嫌う傾向があることから、暗黙的な評価値の推定を行う研究がなされてきている。ここでは陽暗を問わず、正しく評価がなされた場合を仮定する。

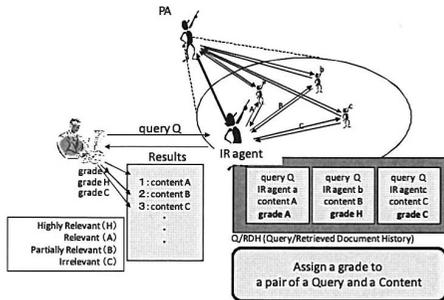


図1 検索結果へのユーザフィードバック
Fig. 1 Grade Assignment to a Pair of a Query and a Retrieved Content.

3.2 ユーザフィードバックの検索への反映

Q/RDH に蓄えられたユーザからのフィードバックは、他の IR agent から同じクエリを受信したとき式 (4) に示すように類似度を計算する際に利用する。これによりユーザからの評価の高いドキュメントが検索結果のランキング上位に出現しやすくなる。

$$Sim_f(Q, D) = Sim_d(Q, D) * r(Q, D, E) \quad (4)$$

関数 $r(Q, D, E)$ は、 D が Q による検索で得られたドキュメントでその適合判定が E の場合の値を返す。実験では、 E =高適合, 適合, 部分適合, 不適合, 未評価の場合、それぞれ $r(Q, D, E) = 4, 3, 0.5, 0.5, 1$ とする。 E のデフォルトは、未評価である。また、 $Sim_d(Q, D)$ は式 (1) である。これを用いて、式 (3) 中の Sim_d を式 (4) の Sim_f とする。

また、Q/RDH 中に登録されているクエリを受信した際、検索結果とともに Q/RDH に記録されているの評価を送信することにより、異なる IR agent 間でドキュメントに対する評価を共有することが可能となり、コミュニティ全体で検索精度の向上が期待できる。

3.3 不適合判断情報の活用

3 で述べたように、ユーザからのフィードバックは評価に対応した評価値を類似度に乗算することにより利用される。このため、高適合、適合の評価を受けたドキュメントは関連度が高くなり、他のユーザへ検索結果として送信されやすくなる。一方、不適合と評価されたドキュメントは関連度が低くなり、他のユーザへ送信されにくくなる。しかし、不適合と評価されたドキュメントの情報（不適合情報）が全くやり取りされない場合、一度あるユーザにより不適合と評価されたドキュメントが再び検索結果として提示されるため、不適合情報すべてを保存する集中型と比べ検索精度が低下する。そこで、他の IR agent からクエリを受信し検索結果を返す際に、不適合情報も一緒に送信する

表2 マルチキャスト依頼受信履歴の構造とその例
Table 2 Structure and an example of Q/MRH.

マルチキャスト依頼受信履歴 (Q/MRH)			
query	from	res-list	tgt-list
依頼されたクエリ	マルチキャストを依頼してきた IR agent のアドレス	マルチキャストで YES と応答した IR agent のアドレスのリスト	マルチキャスト依頼した IR agent へ返された IR agent のアドレスのリスト
Q_A	A	B, C, D, E, F, ..	B, C, D

表3 初回のマルチキャストで作成した IR agent リストと PA が作成する IR agent リストの例 ($N_R = 3$)

Table 3 An example of a list of IR agents as a result of first query multicasting and an example of a list of IR agents made by PA ($N_R = 3$).

query	from	res-list	from	tgt-list
Q	A	B, C, D, E, F, G, H, I, J	A	B, C, D
			K	A, E, F
			L	K, G, H
			M	L, I, J
			N	M, B, C
		

ことにより検索精度の改善が期待できる。実験では、不適合情報を送信しない場合と公平に比較するため、検索結果のうちユーザから評価されていないドキュメントを選び、そのドキュメントの代わりに不適合情報を送信する。

4. PA のクエリ学習によるマルチキャスト手法の改善

4.1 マルチキャスト依頼受信履歴

マルチキャスト時の関連ドキュメントの発見率向上や IR agent 間のメッセージ数削減を目的とした PA のクエリ送信先学習のため、PA はコミュニティ内でマルチキャストしたクエリの管理する。クエリ管理は表 2 に示すマルチキャスト依頼受信履歴 (Q/MRH) を利用して行う。Q/MRH は、マルチキャスト依頼されたクエリ (query) と、依頼した IR agent のアドレス (from)、マルチキャストに対して YES と応答した IR agent のリスト (res-list) およびマルチキャストを依頼した IR agent に返した IR agent のリスト (tgt-list) からなる。

4.2 クエリ送信先学習手法

クエリ送信先学習機能としてマルチキャストで獲得した関連ドキュメントを持つ IR agent のリスト (res-list) を繰り返し利用して、マルチキャストの回数を制限する手法について述べる。この手法では、通信量削減効果とコミュニティ全体からの網羅的な関連ド

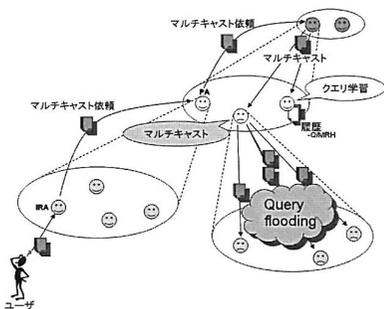


図 2 複数コミュニティ環境での ACP2P 法
Fig. 2 The ACP2P Method in Multiple Communities.

キュメントの収集を狙い、あるクエリに対するマルチキャストを最初の 1 回だけに限定する。あるクエリについて初回のマルチキャストに対して返答をしてきた IR agent のリスト (res-list) を繰り返し使用して IR agent からのマルチキャスト依頼に返答を行う。例えば、 Q というクエリの最初のマルチキャストに対して表 3 に示すような res-list が作成された場合、これ以降同じクエリ Q のマルチキャスト依頼を受けると、PA は前回の依頼 IR agent に加えて res-list の先頭から順番に $N_R - 1$ 個の IR agent を選択し、 N_R 個の IR agent リスト (tgt-list) を作成する。また、リストの終りに達すると再びリストの先頭から選択する。この手順で表 3 のような tgt-list が作成される。

この手法は通信量の削減効果が大きく、またクエリを送信する IR agent の重複が少なくなるため、受信する検索結果ドキュメントの重複も少なくなり、より多くの関連ドキュメントの獲得が期待できる^{*6}。

4.3 複数コミュニティへの応用

複数コミュニティ環境での ACP2P 法を考えると、ACP2P 法の典型的な利用では、コミュニティ内で検索要求数 N_R 個の検索対象エージェントが見つからない場合、PA は 1 つ上位の PA にマルチキャスト依頼を行う。マルチキャスト依頼を受信した上位 PA はコミュニティ内の全 PA にクエリを送信する。このようにして、別のコミュニティ内の IRA も検索対象とすることが出来る。しかし、上位 PA からクエリを受信するたびに自分のコミュニティ内の全 IR agent にクエリをマルチキャストすると、コミュニティ内の IR

agent 分だけ通信が起こりネットワークがクエリで溢れてしまうことが懸念される (図 2 参照)。

そこで、クエリ送信先学習機能で用いた Q/MRH を利用した通信量の削減手法を提案する。上位 PA から受信した際、そのクエリが自身の持つ Q/MRH 中に登録されている場合、from フィールド中の IR agent のみを上位 PA に返す。登録されていない場合は何も返さない。上位 PA は、それぞれの PA から受け取った IR agent の情報を、マルチキャストを依頼した PA に返し、IR agent の情報を受け取った PA は、その中から $N_R - 1$ 個分の IR agent を選択する。 $N_R - 1$ 個に満たない場合は、自分のマルチキャスト依頼受信履歴のリスト中から足りない分を選択し N_R 個の検索対象エージェントのリストを作成する。そして、そのリストをマルチキャストを依頼してきた IR agent に返す。Q/MRH の情報を使用することにより、上位 PA からクエリを受信するたびにマルチキャストを行う必要がなく通信量の削減が期待できる。

5. 実 験

5.1 準 備

今回の実験では、ある PA のコミュニティに 5 個の IR agent が存在し、その各 IR agent が PA となるコミュニティには 100 個の IR agent が参加している状況を想定する。すなわち、PA の役割でない IR agent の数は 500 である。各 IR agent が保持するドキュメントとして、検索性能評価用のテストコレクションである NTCIR3 WEB¹⁾、NTCIR4 WEB²⁾ を使用する。それぞれの検索対象文書データは共通であり、約 1100 万の Web 文書からなる。検索課題として NTCIR3 から 47 個、NTCIR4 から 80 個の計 127 個を用いる。検索課題に対して適合判定が行われている文書数は約 15 万であり、それぞれ高適合、適合、部分適合、不適合の 4 段階の適合判定が行われている。検索課題に対して適合評価が行われている文書だけを 500 個のエージェントに割り当てると一つのエージェントの持つ文書数が少なくなるため、検索課題に対して適合評価が行われていない文書を加え、合計 150 万個の文書を 500 個の IR agent にそれぞれ 3000 個ずつランダムに割り当てる。適合判定が行われていない文書はすべて不適合とする。127 個の検索課題については、その中からランダムに 20 個ずつを IR agent に持たせる。各 IR agent は順番に検索課題から生成されるクエリを送信する。クエリとして使用するのは、検索課題中の < CONCEPT > に記述された 4~7 語のキーワードであり、検索要求に関連した概念、およびその類義語、

^{*6} 一方、毎回のマルチキャストを行わないことから、IR agent リストが更新されず無関係なドキュメントを集めるようになる危険性も考えられ、コミュニティ内でのドキュメントの更新が頻繁に行われる状況では、IR agent リストの更新を適度に行う必要性も考えられる。この検証については今後の課題とする。

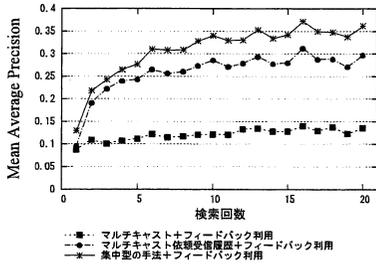


図 3 複数コミュニティ環境での検索精度比較
Fig. 3 Comparison of the MAP Values of Query-Multicasting Methods Applied to Multiple Communities

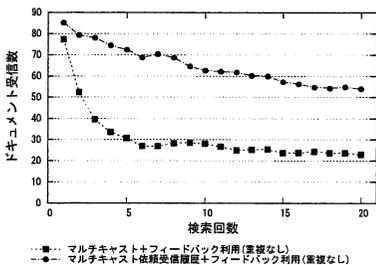


図 4 複数コミュニティ環境でのドキュメント受信数比較
Fig. 4 Comparison of the Number of Documents Used by Query-Multicasting Methods with/without Query-Learning in Multiple Communities

上位語, 下位語からなる. なお検索要求数 $N_R = 10$ とする.

5.2 評価手法

本実験では, 検索課題に対する正解判定を利用して求めた平均精度 (Average Precision) を評価指標とする. 平均精度 $Average\ Precision = \frac{\sum_{r=1}^L I(r)P(r)}{R}$ で求められる. ここで R は全正解文書数であり, L はシステムが出力したランクつき検索結果数であり, 本実験では $L = 10$ とする. $I(r)$ は検索結果の第 r 位における文書が正解であるとき 1, そうでないとき 0 となるフラグである. 本実験では, 検索課題に対して適合および高適合の文書のみを正解と見なす.

$P(r)$ は $P(r) = \frac{count(r)}{r}$ で表され, $count(r)$ は第 r 位までに含まれる正解文書の個数である.

本稿では, 各 IR agent の 1 回の検索毎に平均精度について全 IR agent の平均 (Mean Average Precision (MAP)) を求め, 各手法の比較を行った.

5.3 複数コミュニティにおけるクエリ送信先学習機能の応用

まず, 4.3 で述べた手法について実験を行う. 実験では IR agent は履歴を利用せず毎回 PA にマルチキャスト依頼を行う. 上位 PA からクエリを受信するたびにコミュニティ内にマルチキャストを行う場合と Q/MRH を利用する場合と比較した結果を図 3 に示す. 集中型の手法とは, 全ドキュメントとクエリの類似度計算を行い検索を行う手法である. なおグラフの横軸に検索回数, 縦軸に MAP 値を取る.

図 3 の ACP2P 法の 2 手法を比較すると検索精度に大きな差が出ていることが分る. これは, 以下の 2 つの要因からなると考えられる.

- 過去に同じクエリを検索し, ユーザからの評価を受け取った IR agent は検索対象エージェントになりやすい
- 上位 PA からクエリを受信するたびにマルチキャストを行う場合, 全体に対してマルチキャストを行う場合と同等である

つまり, 過去に同じクエリを検索した IR agent が検索対象エージェントに選ばれるため, 検索対象エージェントが固定化してしまい, 受信する検索結果も重複が多くなる. このため毎回マルチキャストを行う場合, 検索精度が伸びないと考えられる. 図 4 に Q/MRH を利用した場合と利用しない場合で IR agent が受信した検索結果のうち重複を除いたドキュメント数の平均値を示す. この図から, 毎回マルチキャストを行う場合受信する検索結果は Q/MRH を利用する場合と比べ重複が多く, 先の推測が正しいことが裏付けられる.

次に通信量について考察する. ここでは通信量を 1 回の検索に要するエージェント間のメッセージ数と考える. また, 検索要求エージェント数を N_R , コミュニティ数を N_{Com} , 1 つのコミュニティ内のエージェント数を N_{Agent} とする.

5.3.1 毎回マルチキャストを行う場合

IR agent が 1 回の検索を行うときには,

- PA へのマルチキャスト依頼 (1)
- PA によるマルチキャストとそれに対する返答 ($2(N_{Agent} - 1)$)
- 上位 PA へのマルチキャスト依頼 (1)
- 上位 PA によるマルチキャスト ($N_{Com} - 1$)
- 各コミュニティにおけるマルチキャスト ($(N_{Com} - 1) \times 2N_{Agent}$)
- 上位 PA へのマルチキャストに対する返答 ($N_{Com} - 1$)
- 上位 PA からの IR agent リストの通知 (1)

- (H) PA からの検索対象エージェントの通知 (1)
- (I) 検索対象エージェントへのクエリ送信とそれに対する検索結果送信 ($2N_R$)

のメッセージ数が必要になるので、合計 $2(N_{Com} \cdot N_{Agent} + N_{Com} + N_R)$ 個のメッセージ数となる。

5.3.2 マルチキャスト信頼受信履歴を使用する場合

コミュニティ内で初めて検索されるクエリについては、

- (A) PA へのマルチキャスト依頼 (1)
- (B) PA によるマルチキャストとそれに対する返答 ($2(N_{Agent} - 1)$)
- (C) 上位 PA へのマルチキャスト依頼 (1)
- (D) 上位 PA によるマルチキャストとそれに対する返答 ($2(N_{Com} - 1)$)
- (E) 上位 PA からの IRA リストの通知 (1)
- (F) PA からの検索対象エージェントの通知 (1)
- (G) 検索対象エージェントへのクエリ送信とそれに対する検索結果送信 ($2N_R$)

となるので、合計 $2(N_{Agent} + N_{Com} + N_R)$ 個のメッセージ数となる。

さらに、2回目以降のクエリについては、

- (A) PA へのマルチキャスト依頼 (1)
- (B) 上位 PA へのマルチキャスト依頼 (1)
- (C) 上位 PA によるマルチキャストとそれに対する返答 ($2(N_{Com} - 1)$)
- (D) 上位 PA からの IR agent リストの通知 (1)
- (E) PA からの検索対象エージェントの通知 (1)
- (F) 検索対象エージェントへのクエリ送信とそれに対する検索結果送信 ($2N_R$)

となるので、合計 $2(1 + N_{Com} + N_R)$ 個のメッセージ数となる。

毎回マルチキャストを行う場合と Q/MRH を利用する場合を比較すると、2回目以降の検索で $2(N_{Com} \cdot N_{Agent} - 1)$ 個のメッセージ数を削減できる。

以上の結果から、複数コミュニティ環境において上位 PA からクエリを受信した際、Q/MRH を利用することにより検索精度、通信量ともに大幅に改善される。

5.4 複数コミュニティにおける不適合情報の利用

複数コミュニティ環境において、Q/MRH の利用の有無、ならびに不適合情報の利用の有無における検索精度を比較したグラフを図5に示す。Q/MRH と不適合情報の両方を利用した場合が最も検索精度が高く、集中型の手法に近づいている。図6に Q/MRH を利用した場合に IR agent が受信した不適合情報の平均値を比較した結果を示す。

不適合情報を強制的に送信することにより、不適合

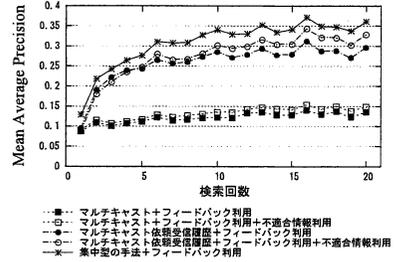


図5 複数コミュニティ環境での不適合情報利用時の検索精度比較
Fig. 5 Comparison of the MAP Values of Query-Multicasting Methods Applied to Multiple Communities

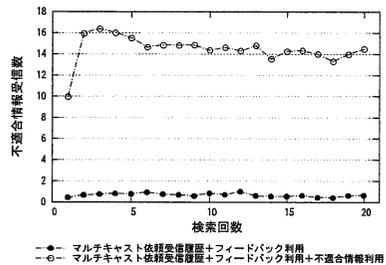


図6 複数コミュニティ環境での Q/MRH 使用時の不適合情報の利用数比較
Fig. 6 Comparison of the Number of Irrelevant Information Used by Query-Multicasting Methods with Query-Learning in Multiple Communities

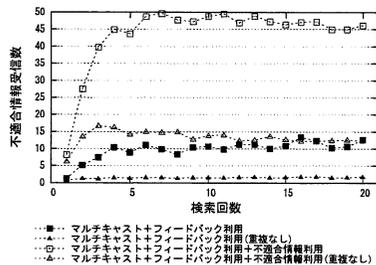


図7 複数コミュニティ環境での Q/MRH 未使用時の不適合情報の利用数比較
Fig. 7 Comparison of the Number of Irrelevant Information Used by Query-Multicasting Methods without Query-Learning in Multiple Communities

情報を送信しない場合と比べ不適合情報がエージェント間で共有されていることが分る。しかし、Q/MRH を利用した場合、コミュニティに網羅的にクエリが送信されるため、不適合情報の収集量は多くならない。そこで、クエリ送信先学習機能を使用する場合でも効

果的に不適合情報を集めることによりさらなる検索精度の改善が期待できる。

また、毎回マルチキャストを行う場合の不適合情報の利用の有無を比較すると、検索精度が多少は向上しているものの、その差はほとんど見られない。図 7 に毎回マルチキャストを行う場合で、IR agent が受信した不適合情報数の平均値と受信した情報のうち重複を除いた情報数の平均値を示す。不適合情報を強制的に送信することにより、大量の不適合情報が IR agent 間でやり取りされている。しかし、4.3 でも述べたように、Q/MRH を利用しない場合、検索対象エージェントが固定化され検索結果の重複が多くなるため、検索精度が向上しないと考えられる。

6. まとめ

本稿では、複数コミュニティ環境における通信量の削減手法と不適合情報の有効性について議論した。実験の結果から、複数コミュニティ環境において Q/MRH を利用することにより、検索精度と通信量のどちらも大幅に改善されることが示された。また、不適合情報を利用することにより検索精度が向上することが示された。とくに Q/MRH と不適合情報の両方を利用することにより検索精度が大きく改善し、クエリに対する適合情報の集約に大きな効果があることが示されたが、フィードバック情報を利用する集中型には若干及ばなかった。これは、クエリ送信先学習機能を使用した場合、クエリがコミュニティ内に網羅的に送信されるため不適合情報が集まりにくいためである。しかし、クエリ送信先学習機能をさらに活用することで不適合情報の効果をさらに引き出すことも可能である。これについては別稿で議論する。

今後の課題として、同一ドキュメントに対する各ユーザからの評価にばらつきを与えた環境での比較実験、複数コミュニティに IR agent を配置したより複雑かつ大規模な環境でのシミュレーションの実施などが挙げられる。

謝辞 NTCIR コレクションは国立情報学研究所の許可を得て使用させて頂きました。ここに深く感謝いたします。

参考文献

- 1) The 3rd NTCIR Workshop Data. <http://research.nii.ac.jp/ntcir/ntcir-ws3/work-en.html#data>, 2002.
- 2) The 4th NTCIR Workshop Data. <http://research.nii.ac.jp/ntcir-ws4/data-en.html>, 2003.
- 3) 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.
- 4) Gnutella. Gnutella protocol development v6.0, <http://rfc-gnutella.sourceforge.net/>, 2003.
- 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 (CIKM03)*, pp. 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, 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)*, pp. 1484–1485, 7 2004. poster.
- 8) Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard Karp, and Scott Shenker. A scalable content-addressable network. In *SIGCOMM*, pp. 161–172, 2001.
- 9) S. E. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford. Okapi at trec-3. In *NIST Special Publication 500-226: The Third Text REtrieval Conference (TREC-3)*, 1994.
- 10) 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*, pp. 149–160, 2001.
- 11) Haiheng Zhang and Victor Lesser. A reinforcement learning based distributed search algorithm for hierarchical peer-to-peer information retrieval systems. In *AAMAS2007*, pp. 231–238, 5 2007.
- 12) 古後陽大, 峯恒憲, 両宮聡史, 両宮真人. Acp2p 法におけるユーザフィードバックの利用とクエリ送信先決定法の提案. 合同エージェントワークショップ&シンポジウム 2008(JAWS2008), pp. CD-ROM, 10 2008.
- 13) 峯恒憲, 松野大輔, 両宮真人. 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/>, Vol. 19, No. 5, pp. 421–428, 2004.