

モバイル協調検索のためのクエリ推薦・提示手法

大重 智志[†] 中村 聡史^{††} 田中 克己^{††}

[†] 京都大学工学部情報学科 〒606-8501 京都府京都市左京区吉田本町

^{††} 京都大学大学院情報学研究科社会情報学専攻 〒606-8501 京都府京都市左京区吉田本町

E-mail: [†]{ohshige,nakamura,tanaka}@dl.kuis.kyoto-u.ac.jp

あらまし 本稿では、モバイルを使用した複数人での協調検索を効率的に行うための手法を提案する。モバイル協調検索では結論を求めるために意思決定を行う必要があるが、スムーズに進まないことが多い。その理由として、複数ユーザが重複した検索行動を行うため網羅的に検索できていないという点と、ユーザが各自バラバラに検索を行うことでの的を絞れていないという点の2つの問題がある。この2つは、探索範囲が狭くなってしまっていることと、探索範囲が広がってしまっていることとで、トレードオフの関係にある。そこで本稿では、グループ全体として検索の話題がどれほど発散しているか、もしくはどれほど収束しているかということを推測し、状況に応じて話題の発散・収束を促すクエリを生成して推薦する手法を提案する。また、提案した手法を実装し、有効性を実験により明らかにする。キーワード 協調検索, モバイル協調検索, クエリ推薦

1. はじめに

爆発的な人気を博すスマートフォンによって、人々は時と場所を選ばずに Web 検索を行うことができるようになった。総務省によると 2011 年 9 月時点での携帯電話の人口普及率は 96.2%に達し [1], 携帯電話所有者のうち 8.9%がスマートフォン所有者, 11.5%が所有意向者である。特に, 20 代のスマートフォン所有者は 23.1%, 所有意向者は 16.2%にもなる [2]。

Morris [3] は, 複数のユーザが協調して Web 上で情報検索を行うという行為が多く行われていることを明らかにしている。例えば, グループ旅行の計画のために必要な情報を皆で検索したり, ネットショッピングで様々な候補の中から最善のものを選択し購入したりするなどの状況が挙げられる。本論文では, このような複数ユーザでの検索行動を協調検索と呼ぶ。スマートフォンの普及により, 時と場所を選ばずに Web 検索を行うことができるようになったため, PC を利用した協調検索だけではなく, 小型のモバイル端末を用いて状況に応じた協調検索 (モバイル協調検索) を行う機会も増えてきた。例えば, グループでの旅行中にその周辺の観光スポットを探すような状況, パーティーの後で急遽行くこととなった 2 次会の店を探す状況, 災害直後に被災したメンバーと帰路を探す状況などがある。一般的な PC を用いた協調検索とモバイル協調検索の大きな違いは, 後者は突発的に検索を開始することが多く, 前者に比べて情報検索に十分な時間を掛けることができない点である (理由については 4 章で述べる)。また, 後者はディスプレイサイズが小さいため表示および共有できる情報量にも制限がある。

モバイル協調検索では, グループとして満足する結論に達する必要がある, 意思決定が行われる。しかし, モバイル協調検索では意思決定までスムーズに進めることは容易ではない。この原因として, 我々は大きく分けて 2 つあると考えている。1 つ目は, グループの複数ユーザが重複およびとても類似した検索行動を行ってしまう点である。グループとしての検索行動の

方針が決まっていない検索開始初期などにおいて, 似たようなクエリで検索すると, グループ全体での探索範囲が狭まってしまう。そのために, 網羅的な探索を行うことができず, グループ全体として質の高い良いページに辿り着けない場合がある。結論として, 満足のいく結論に至らないこともある。2 つ目は, 各ユーザが個人の好みに合わせすぎて自由に探索を行ってしまうという点である。各ユーザの好みだけで行動すると, いつまで経っても探索範囲がバラバラでグループとしての的を絞りきれなくなってしまう。結果として, 最終的なゴールに辿り着けず目的を果たせなくなってしまううえ, 時間の浪費となってしまう。また, この両者は探索範囲が狭くなっていることと探索範囲が広がっていることとでトレードオフの関係にある。

そこで本論文では, 上記の問題を解決し, モバイル協調検索における円滑な意思決定を支援する手法を提案する。ここでは, 色々な候補を検討しつつ意思決定までスムーズに進めるため, グループ全体の話題の発散と収束をシステムによりコントロールすることを考える。ここでの「話題」とは, ある時点におけるグループ全体の検索目的に関するトピックやテーマのことである。本手法では, 話題を発散させることで探索範囲を広げてグループ内での重複した検索行動を防ぎ, 話題を収束させることでグループ内での探索範囲を狭めて的を絞った情報検索を行えるようにする。具体的には, 話題の発散を促すクエリと話題の収束を促すクエリを推薦することで, グループ全体としての話題の発散・収束をコントロールし, 意思決定における問題の解決を図る。ここでは, 発散を促すクエリを「発散クエリ」, 収束を促すクエリを「収束クエリ」と呼ぶ。また, システムが各ユーザの入力したクエリからグループ全体としてのおおまかな意図を推定し, その推定された意図に基づいて発散および収束クエリを生成・推薦する。なおここで, 発散クエリとは, ある入力されたクエリとは関連があるが別の観点によるもののことである。また, 収束クエリとは, ある入力されたクエリと関連があり, さらに詳細なものを指すものとする。

協調検索のある段階において「今は話題を発散させ、さらに検索範囲を広げるべきだ」、「今は話題を収束させ、的を絞って検索するべきだ」と一概に言えるとは考えにくい．そこで本論文では、現在のグループ全体での話題の発散または収束度合いを割合（話題の収束度）として表現し、その割合に基づいて推薦する発散クエリと収束クエリの数の割合を変更する．我々はこれらの手法をプロトタイプシステムとして iPod touch 上で実装し、提案手法の有用性を実験により明らかにした．

2. 関連研究

2.1 PC を用いた協調検索

SearchTogether [4] は、遠隔地にいるユーザが PC を用いて同期的・非同期的に協調検索を行うためのシステムである．各ユーザがそれぞれの PC で入力したクエリが他の協調ユーザと共有されたり、気に入ったページを推薦したり、ページに対して評価やコメントなどを付加できるなどの機能がある．ユーザはこうした機能を利用することで、効率的に協調検索を行える．

CoSearch [5] では、複数のユーザが 1 台の PC と複数のマウス・モバイル端末を用いて協調検索を行う．各ユーザにはカーソルが 1 つずつ割り当てられ、ディスプレイ上に色で区別されたカーソルが表示されている．各ユーザは自身のカーソルをマウスで操作し、クエリや閲覧ページをキューに貯めて協調検索を行う．モバイル端末では、マウスの機能に加え、クエリ入力やページの閲覧、ページに対するメモなどが行える．これにより、1 台の PC 上で効率的に協調検索を行うことができる．

伊豆らの研究 [6], [7] では、ユーザの探索に同期してシステムが有用な Web ページをグループの共有データ内から選び提示することで、探索を支援する．検索結果に至るプロセスやその行為を含めて共有することで、ユーザの探索目的や探索範囲を把握する．ユーザの行動に沿ったページを提示するために、ユーザの検索行動を、いろいろなページを見ている発散フェーズ、類似ページを続けて見ている収束フェーズ、ユーザが考える範囲を調べ終えた補完フェーズに分け、ユーザがどのフェーズにいるかを判定し、それに応じて何を提示すべきかを判断している．それらのフェーズは、探索目的に沿ったページをどれ程多く閲覧したかを表す指標である達成度と、どの程度他の類似 Web ページと吟味されて推薦ページは選ばれたのかを表す指標である精選度の 2 つの尺度を用いて決定している．

これらの研究は 1 台以上の PC が使えることを想定している．本論文は、モバイル端末のみが使える PC は使えない状況を想定しているという点でこれらの研究とは異なる．また、伊豆らの研究は既に閲覧されたページを提示するのにに対し、本論文ではクエリを推薦するという点で異なる．さらに、本論文でも発散・収束という語を用いてはいるが、発散・収束がどの程度の強さであるかという考え方をを用いている我々とは異なる．

2.2 モバイルだけを用いた協調検索

奥らの研究 [8] では、ユーザの意見を反映したウェブコンテンツを共有し、それらをユーザの評価で整理および提示することで、ウェブコンテンツの比較・検討を支援するインタフェースを提案している．

表 1 「札幌旅行で食事する場所を探す」というタスクのクエリログ

経過時間	User A	User B	User C	User D
00:00			札幌 観光	
00:02				ろっかてい 北海道
00:40			札幌駅 観光	
01:57				海鮮丼 札幌
02:10		札幌 名所		
02:16			札幌駅 名産	
02:23	水族館 札幌			
03:50				札幌 牧場
04:08	北大			
04:44		札幌 観光		
05:51	石山緑地			
06:03				札幌 海鮮丼
06:05			札幌駅 海鮮丼	
06:17				札幌 海鮮丼
07:17				札幌 ろっかてい
08:15		真駒内 駅		
08:33	札幌 ケーキ			

表 2 「福岡旅行で食事する場所を探す」というタスクのクエリログ

経過時間	User A	User B	User C	User D
00:00		福岡		
00:05				博多 観光
00:08		福岡 観光		
00:10	博多 観光			
00:17			博多 観光	
00:19		福岡 観光 半日コース		
01:22				博多ぶらぶら
01:54				博多 明太子
02:37				博多 駅前
02:49				博多 名物
02:58			博多 寺社仏閣	
03:03		福岡 食べ物		
04:06				博多 もつ鍋
04:15		お土産 博多駅		
04:41				とおりもん
04:42		お土産 博多駅		
05:13		福岡 周辺 もつ鍋		
05:28				博多 アウトレット
05:54		福岡 周辺 もつ鍋 博多駅		
06:42				博多の塩
06:43			博多 買い物	
07:04				明太子 有名店
07:47				柳橋通 博多
07:56				柳橋 博多
08:13			博多 名所	
08:59	博多 土産			

小谷らの研究 [9] では、気に入ったウェブページを他のユーザの検索結果一覧に割り込ませることにより通知する機能や、各ユーザの探索空間を共有する機能により、ユーザ間の情報共有を支援している．

これらの研究は我々の研究同様モバイル端末を用いた協調検索を想定しており、情報の比較・共有に注目している．本論文では比較や共有という観点ではなく、話題の発散や収束を促すことで意思決定をスムーズに進めるよう支援するものである．

3. 分 析

我々は、モバイル協調検索を行う際のユーザの検索行動の特徴を検証するため、これまでに予備実験を行ってきた [9]．ここでは、被験者として 20 代の男性 16 人を 4 人ずつ 4 グループに分け、それぞれのグループに 2 つのタスクを 1 タスクあたり 10 分程度で実行してもらっている．各クエリログの内容に関して分析することで、クエリ推薦の重要性を示す．そのときのクエリログの一部は表 1～表 4 の通りである．

表 1 のタスクでは、初期クエリを入力するまで、User A も User B も 2 分以上という長い時間がかかっている．この理由としては、(1) 他のユーザと話している、(2) 入力に手間取っている、(3) クエリを思いつかない、このようなことが考えられる．このとき、クエリを推薦することで入力自体の煩わしさを解消でき、(2) を支援することができる．これは、特に入力か

表 3 「仙台旅行で食事する場所を探す」というタスクのクエリログ

経過時間	User A	User B	User C	User D
00:00	仙台駅前 牛タン			
00:04			牛タン	牛タン 焼肉
00:18		仙台 牛タン 焼き肉		
00:20			牛タン	
00:26				牛タン 焼肉 仙台
01:13			仙台	
01:24		仙台駅前 牛タン 焼き肉		仙台牛タン
01:28				
03:40	仙台駅前 牛タン			食べログ 牛タン
03:48				食べログ 牛タン 仙台
05:23				
05:58				

表 4 「金沢旅行で観光する場所を探す」というタスクのクエリログ

経過時間	User A	User B	User C	User D
00:00	金沢駅前 食べログ		金沢市 晩ご飯	
00:06		金沢駅 寿司		
00:12	食べログ			金沢駅 ご飯
00:14		金沢 名物	金沢市 ご飯	
00:17			金沢市 ディナー	
00:50		金沢駅 かに		金沢駅 ご飯 オススメ
01:00			金沢市 中華	
01:58		金沢駅 かにめし		
02:03		金沢駅 かに		
03:03				
04:38				
06:17				
07:28				
07:43				

容易ではないモバイル環境だからこそ必要なものである。また、グループの意図に沿ったクエリを推薦することでクエリ例を提示でき、(3) を支援することができる。

表 2 のタスクの場合、クエリ中に「もつ鍋」、「明太子」、「通りもん」などがあり、これらは全て福岡の名産である。しかし、福岡の名産としては他にも「水炊き」、「博多ラーメン」、「ひよこ」などがあり、これらに関しては調べられておらず、名産をカバーしきれていない。ここで、「水炊き」、「博多ラーメン」、「ひよこ」などをクエリとして推薦することで、グループ全体での網羅性をあげることができると期待される。

表 3 のタスクでは、クエリ中に「牛タン」という語が多く出てきており、「牛タン」という素材はほぼ決定しているということになる。さらに、「牛タン 焼き肉」というクエリがあることから、牛タンをどのように食べるかということを調べていることが分かる。つまり的が絞れている場合、より踏み込んだ詳細な検索を必要としているということがわかる。

表 4 のタスクでは、User B は 2 分 3 秒に「金沢駅 かに」、User C は 4 分 32 秒に「金沢市 中華」というクエリで検索している。この場合、「かに」と「中華」という好みで衝突が起き、答えが決まらない可能性があり、どちらかが妥協するか、別の候補を見つけるということが必要となる。そこで、クエリ推薦によって、歩み寄りを促したり新たな候補を提示したりすることで、衝突を緩和することが考えられる。例えば、「金沢駅 かに」というクエリを User C に推薦して歩み寄りを促したり、逆に「金沢市 中華」というクエリを User B に推薦して歩み寄りを促したりすることができると期待される。また、両者の間をとるような「金沢 かに玉」というクエリや、全く違う候補としての「金沢 寿司」というクエリを推薦し、衝突を緩和および回避する方法も考えられる。

以上のように、効果的なクエリ推薦によって入力自体の煩わしさを解消したり、グループ全体での網羅性をあげたり、より

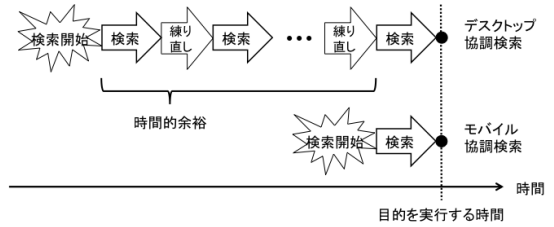


図 1 協調検索による時間的余裕の違い

詳細な踏み込んだ検索を支援できたり、ユーザ間の衝突を緩和したりすることができると期待される。

4. 提案手法

4.1 モバイル協調検索の問題点と解決案

モバイル協調検索はグループ単位で行う検索であるので最終的にグループ全体として満足する必要がある、全員が満足する意思決定のためのページを発見しなければならない。しかし、モバイル協調検索では意思決定までスムーズに進めることは容易ではない。1 章で述べた通り、我々はグループの重複した検索行動と目的を絞れない検索の仕方が原因であると考えている。

一般的な協調検索でも意思決定までスムーズに進まないという問題はあ。しかし、一般的な協調検索では意思決定まで時間を掛けることが可能であるが、モバイル協調検索は突発的に行うことが多く検索行動に掛けられる時間が限られているため、意思決定までスムーズに進めることが特に重要となる。図 1 は、デスクトップを用いた一般的な協調検索に掛けられる時間と、モバイル協調検索に掛けられる時間の差を模式的に表したものである。一般的な協調検索では、ある目的を実行する時間まで時間的な余裕があるのが普通であり、検索や繰り返しを繰り返し行うことで全員が満足する結論を得ることができる。しかし、モバイル協調検索では目的を実行する時間までに余裕がなく、意思決定までスムーズに進めることがより重要となる。

そこで我々は、話題の発散と収束という概念を導入し、話題の発散を促すクエリと話題の収束を促すクエリをユーザに推薦することで問題の解決を図る。また、協調検索のある段階において話題がどれほど発散・収束しているかという割合を推定する。ここでは、特に、どれほど話題が収束しているかに着目し、その割合を話題の収束度と呼ぶ。話題の収束度が高いほど話題は収束し、低いほど発散しているということになる。そして、話題の収束度から、話題をどれほど発散・収束すべきかという方針を考える。また、話題の収束度に基づいて、推薦する発散クエリと収束クエリの数の割合を変更する。すなわち、ある時点での話題の発散・収束具合を動的に推定し、それに応じて推薦するクエリの内容も動的に変更するというものである。

4.2 話題の収束度

グループの話題の収束度を求めるため、ユーザの閲覧したページの類似度と選択した推薦クエリの 2 つに注目する。

4.2.1 閲覧ページの類似度の利用

ある時点における、閲覧しているページと今までに閲覧した各ページとの類似度の重み付き平均 SimilarityScore を、話題

の収束度を決定するための一要素とする。

まず、ユーザが新規ページ p を閲覧するたびにそのページに含まれている語を抽出し、その語の頻度を利用した特徴ベクトル $v_p = (v_{p,t_1}, v_{p,t_2}, \dots, v_{p,t_n})$ を求める。 v_{p,t_i} はここで、

$$v_{p,t_i} = \frac{\text{freq}(p, t_i)}{\sum_{t \in T_p} \text{freq}(p, t)}$$

であり、 $\text{freq}(p, t)$ はページ p における単語 t の出現頻度、 T_p はページ p に出現する全単語集合である。

また、最近閲覧したページと協調検索開始直後に閲覧したページとでは重要度が異なり、最近閲覧したページの方がグループの話題の収束具合を表していると考えられる。そこで、あるページ p の重み w を、検索行動開始からページ p を閲覧し始めるまでに経過した時間 s_p を用いて、 $w(p) = \text{normalize}(s_p)$ とする。なお、 $\text{normalize}(s_p)$ は s_p を 0 から 1 までの値に正規化するための関数であり、0 が最も古いものを表し、1 が最も新しいものを表す。

そして、あるユーザがページ p_{now} を見ているとき、その時点での SimilarityScore を、

$$\text{SimilarityScore}(p_{now}, P) = \frac{\sum_{p \in P} \text{Sim}(v_{p_{now}}, v_p) \cdot w(p)}{\sum_{p \in P} w(p) + 1} \quad (1)$$

として求める。ここで、 P はグループ全体の既閲覧ページ集合、 $\text{Sim}(v_{p_1}, v_{p_2})$ はページ p_1 と p_2 の特徴ベクトル v_{p_1} と v_{p_2} のコサイン類似度である。これにより、ある時点における、最近閲覧したページと今までに閲覧した各ページとの類似度の重み付き平均が求まる。この値が低い時、グループとして類似していないページを閲覧していることになり、話題は発散の段階にある。この値が高い時、グループとして類似したページを閲覧していることになり、話題は収束の段階にある。

伊豆らの研究 [6], [7] でも、先述の達成度と精選度を求めるために閲覧ページの特徴ベクトルの類似度を用いている。しかし、伊豆らは達成度を検索キーワードとクリックアンカー文字列による特徴ベクトルと各閲覧ページの特徴ベクトルの類似度の総和を取って求めており、本論文がページ同士の類似度を用いている点で異なる。また、伊豆らは精選度は推薦ページの特徴ベクトルと各閲覧ページの特徴ベクトルの類似度の総和を取って求めており、本論文がより最近閲覧したページが類似しているほどより収束していると考えている点で異なる。

4.2.2 選択した推薦クエリの利用

推薦クエリのうち発散クエリが多く選択されている場合、話題が発散の段階にあると考えられる。一方、収束クエリが多く選択されている場合、話題が収束の段階にあると考えられる。

そこである時点における、各ユーザの選択した直近の推薦クエリのうちで収束クエリの割合 ConQueryScore を、話題の収束度を決定するための一要素とする。

ある時点における直近の各ユーザの入力クエリ集合 $Q = (q_{u_1}, q_{u_2}, \dots, q_{u_n})$ のうち、発散クエリとして推薦されたものの集合を $Q_{div}(Q_{div} \subseteq Q)$ 、収束クエリとして推薦されたものの集合を $Q_{con}(Q_{con} \subseteq Q)$ として、以下で求める。

$$\text{ConQueryScore}(Q) = \frac{|Q_{con}|}{|Q_{div}| + |Q_{con}|} \quad (2)$$

4.2.3 話題の収束度の計算

以上が話題の収束度を求めるための要素を求める方法であり、そのページの類似度による式 (1) とクエリの種類による式 (2) を用いて、話題の収束度を求める。

ある時点における話題の収束度 ConvergenceScore は、

$$\begin{aligned} \text{ConvergenceScore} \\ = \alpha \cdot \text{SimilarityScore}(p_{latest}, P) + \beta \cdot \text{ConQueryScore}(Q) \end{aligned} \quad (3)$$

として求める。ここで、 p_{latest} は最も最近閲覧されたページ、 P はグループ全体の既閲覧ページ集合、 Q は直近の各ユーザの入力クエリ集合、 α と β は重みである。

4.3 推薦クエリの生成

各ユーザの入力したクエリからグループ全体としてのおおまかな検索の背景を推定し、その後、その背景に基づいて発散クエリと収束クエリを生成する。ここで背景とは、検索の文脈を表すもので、そのような背景を表した語を背景語と呼ぶ。

4.3.1 発散クエリ

発散クエリとしては、あるクエリをシードとし、そのクエリと関連はあるが別の観点であるものを生成する。以降、シードとなるクエリのことをシードクエリと呼ぶ。

例えば、“京都 観光”をシードクエリとして考える。このクエリには2つの背景が考えられ、それは「京都へ旅行するとき、観光地について検索する」という背景と「関西を観光したので、特に京都について検索する」という背景である。前者の場合、京都で観光だけでなくショッピングのことも考えられるため、“京都 ショッピング”が発散クエリとして考えられる。後者の場合、関西での観光地は他にも大阪があるため、“大阪 観光”が発散クエリとして考えられる。

ここで、各クエリに含まれる語に関して注目すると、前者の場合「京都」、後者の場合「観光」という語が共通して現れており、それぞれが背景に対応している。また、前者の場合「観光」と「ショッピング」という語が、後者の場合「京都」と「大阪」という語が、それぞれ同位概念のような関係になっている。以降、このように同位概念のような関係になる語を兄弟語と呼ぶ。

そこで、あるシードクエリ“ $t_1 t_2$ ”から背景を表す語 t_1 を選び、残りの語 t_2 の兄弟語 t_2' を求め、新しい発散クエリ“ $t_1 t_2'$ ”を求める。このようにして発散クエリを生成する。

4.3.2 収束クエリ

収束クエリとしては、あるクエリをシードとし、そのクエリと関連があり、さらにそのクエリを詳細化するものを生成する。

以下、“京都 観光”をシードクエリとして考える。“京都 観光”で「観光」に注目して詳細化する場合「京都での観光、特に紅葉が見れるところ」という意味で“京都 観光 紅葉”が収束クエリとして考えられる。また「京都」に注目すると「京都、特に嵐山での観光」という意味で“京都 嵐山 観光”などが収束クエリにあたると考えられる。

ここで、クエリに含まれる語に関して注目すると「紅葉」は

「京都」という背景で「観光」を詳細にするもので、「嵐山」は「観光」という背景で「京都」を詳細にするものである。

また、グループ全体としての絞るという意味で、シードクエリ自身もそのシードクエリを入力したユーザ以外のユーザにとっては収束クエリとなりうる。

そこで、あるシードクエリ“ $t_1 t_2$ ”から背景を表す語 t_1 を選び、残りの語 t_2 をより詳細にする語 t_3 (詳細語) を求め、収束クエリ“ $t_1 t_2 t_3$ ”を求める。また、シードクエリ“ $t_1 t_2$ ”自身も収束クエリとする。このようにして収束クエリを生成する。

4.3.3 背景語の発見

クエリに多く現れる語はそれだけ背景を意味していると考えられるので、背景語は全クエリに含まれる単語のうち頻出のものとする。頻出の語はひとつとは限らないので、背景語は集合として表す。求める背景語集合を $T_{context}$ とすると、

$$T_{context} = \operatorname{argmax}_{t \in T} \operatorname{freq}(t) \quad (4)$$

で求められる。ただし、 T は全クエリ中の単語集合で、 $\operatorname{freq}(t)$ は単語 t の出現頻度である。

4.3.4 シードクエリの発見

発散・収束クエリを生成するためのシードクエリは、クエリ履歴の中から、式 (4) で求めた背景語集合の要素が含まれるクエリを取得する。ただし、簡単化のために、シードクエリはある単語と背景語の2つだけからなるものに限る。このようにしてシードクエリを発見する。

ここで、取得したシードクエリ候補集合 Q_{base} の要素 q の重み $w_{base}(q)$ を求める。そのクエリ q に含まれる背景語ではない語を t 、単語 t の出現頻度を $\operatorname{freq}(t)$ 、検索行動を開始してからそのクエリを入力するまでに経過した時間を s_q とすると、

$$w_{base}(q) = \alpha \cdot \operatorname{freq}(t) + \beta \cdot \operatorname{normalize}(s_q) \quad (5)$$

で求める。ただし、 $\operatorname{normalize}(s_q)$ は式 (1) で使用したのと同じで、 α と β は重みである。この式は、グループとしてよく検索されている単語を含み、その語を含むクエリが最近検索されているほど重みが大きくなるというものである。検索されていなければグループとしての意図を反映できず、より最近のクエリほど最近の意図を表していると考えられるからである。

4.3.5 兄弟語と詳細語の発見

クエリ“ $t_1 t_2$ ”から背景を表す語 t_1 を選んだ後は、残りの語 t_2 の兄弟語もしくは詳細語を見つける必要がある。兄弟語や詳細語を求める研究は数多くあるが、本論文では大島ら [10]、野田ら [11] の研究に着目した。

大島らは本論文での「兄弟語」を「同位語」と呼んでいる。ある語に対する同位語は並列助詞「や」で接続されることを利用して、Web 検索エンジンに対するクエリを作成し、その検索結果から同位語を得るというものである。また、野田らは本論文での「詳細語」を「話題語」と呼んでいる。ある語に対する話題語は連体助詞「の」で接続されることを利用して、Web 検索エンジンに対するクエリを作成し、検索結果のウェブページ群から話題語を得るというものである。

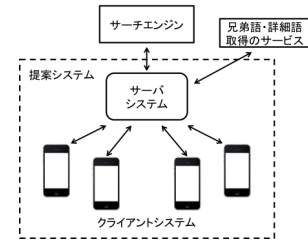


図2 システムの構成

これら同位語と話題語を取得する手法が実装された API^(注1) を利用し、本論文では兄弟語と詳細語を取得する。

4.4 クエリの推薦

ある状況において、発散クエリと収束クエリをそれぞれどのくらいの割合で推薦するかは、式 (3) で求めたスコアを使用する。推薦できるクエリの最大数を $n_{all} (> 0)$ としたとき、ある時点の収束クエリの数 n_{con} は、

$$n_{con} = \operatorname{round}(n_{all} \cdot \operatorname{ConvergenceScore})$$

で求める。ここで、 $\operatorname{round}(n)$ は n を四捨五入した値を求める関数である。また、発散クエリの数 n_{div} は $n_{div} = n_{all} - n_{con}$ で求める。ただし、発散クエリも収束クエリも最低1つは推薦できるように、 $n_{con} = 0$ のときは n_{con} を1とし、 $n_{con} = n_{all}$ のときは n_{con} を $n_{all} - 1$ とする。

モバイル協調検索では、ディスプレイサイズの制約があり一度に推薦できるクエリの数には制限があるため、4.3 節で生成したクエリのうちのクエリを推薦するかが重要になる。そこで、生成した各クエリに重み w を与え、その値の上位から先に求めた n_{div} もしくは n_{con} の分だけ推薦する。クエリ q の重み w は、式 (5) で求めた w_{base} と、そのクエリを構成する兄弟語もしくは詳細語の順位 r (前述の API から取得する際に付加されているメタデータ) を用いて、以下の式で求める。

$$w(q) = \begin{cases} w_{base}(q) & (\text{シードクエリ自身が収束クエリの場合}) \\ \alpha \cdot w_{base}(q) + \beta \cdot \frac{1}{r} & (\text{それ以外}) \end{cases}$$

5. システムの実装

我々は、4章の手法を基にサーバ/クライアントモデルを採用したプロトタイプシステムを設計した。サーバは Python を用いて CGI として動作し、クライアントは Objective-C を用いて iOS5.0 を搭載した端末上で動作する。システムの構成は図2のようになっており、サーバは検索エンジンや兄弟語・詳細語を取得するためのサーバとやりとりしている。

図3は検索を行うメイン画面である。上部にクエリを入力するサーチバーがあり、それを選択すると図4のようにキーボードが現れる。キーボードの検索ボタンを押すと、中央に検索結果がランキングで表示される。また、サーチバーと検索結果の間に、推薦クエリを表示するエリアがある。本システムでは、6つのクエリを推薦する。

(注1): <http://www.dl.kuis.kyoto-u.ac.jp/~ohshima/wiki/index.php?YayalaAPI>

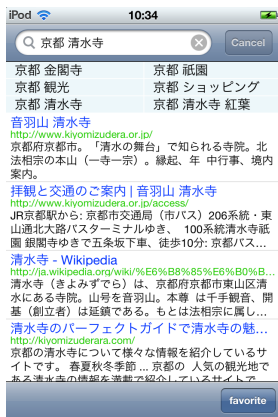


図 3 メイン画面

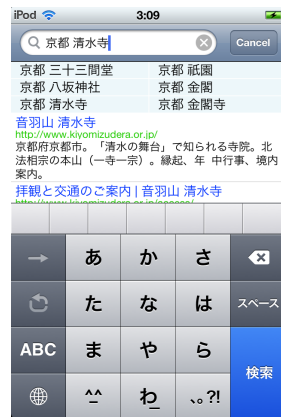


図 4 メイン画面のキーボード

6. 評価実験

提案手法の有用性を示すため、協調検索によって被験者にあるタスクを達成してもらうという実験を実施した。実験によって、発散クエリによってグループとして網羅的に検索し様々な候補を検討できているか、収束クエリによってグループとしての絞った検索ができているか、それらによって意思決定がスムーズに進んでいるか、これら 3 つを明らかにする。

6.1 比較システム

我々は、提案手法との比較のために、5 章のプロトタイプとほとんど同じではあるが、一部機能が異なるシステムを作成した。

提案手法によるプロトタイプシステム（以下、提案システム）と比較のためのシステム（以下、比較システム）とでは、推薦されるクエリの生成・推薦の仕方のみが異なっている。提案システムでは、推薦されるクエリを 4 章の手法により生成・推薦しているが、比較システムでは、Yahoo! の関連検索ワードの API^(注2)を用いている。ユーザの入力したクエリから API によって関連検索ワードを取得し、それらを推薦する。比較システムは、一般的な関連キーワードによるクエリ推薦を模したものであり、システムを通して協調することは想定していない。

6.2 実験内容

我々は、大きく分けて 2 つの実験を行った。1 つ目は協調検索による実験あり、2 つ目は全ての協調検索の実験が終わった後の客観的な評価である。

6.2.1 協調検索による実験

我々はまず、システムを利用しモバイル協調検索のシチュエーションを模した実験を行った。この実験では先述の通り、網羅的に検索できているか、的を絞った検索ができているか、意思決定がスムーズに進んでいるか、これら 3 つを明らかにする。

まず、実際にグループ単位でタスクを与え、協調検索を行ってもらう。その後、どのような方針で協調検索を行ったかをディスカッションしたうえで、最終的にアンケートに回答する。この流れを 1 グループにつきそれぞれの手法で 1 回ずつ行う。

なお、協調検索を開始する前には、システムに依存しない事

表 5 掛かった時間と発話数の平均

	結論に達するまでの秒数	発話回数
提案システム	589.1 ($\sigma = 54.5$)	25.2 ($\sigma = 12.9$)
比較システム	604.3 ($\sigma = 37.4$)	29.8 ($\sigma = 14.7$)

前アンケートに回答してもらう。内容は、日常的に協調検索を行うか、スマートフォンの入力に慣れているかなどである。

日常的に会話を行うメンバー 4 人をひとつのグループとして、合計 8 グループ、全部で 32 名の男女を被験者とした。年齢は 18 ~ 33 歳で、平均年齢は 22.3 歳である。

我々が与えたタスクを、提案システムと比較システムとでそれぞれ 1 タスク 10 分程度で行ってもらった。実験は、20m² 程度の個室で立った状態で、経過時間が分かるようにタイマーを被験者に見えるようにしたうえで、以下の手順で行った。

- (1) 実験の内容と目的を説明する。
- (2) システムの使い方を説明し、操作に慣れってもらう。
- (3) 事前アンケートに回答してもらう。
- (4) 1 つ目のタスクを与え、協調検索を行ってもらう。
- (5) 協調検索の方針をディスカッションしてもらう。
- (6) 5 段階のリッカート尺度で問うアンケートを行う。
- (7) 2 つ目のタスクに対して、(4) ~ (6) を繰り返す。

実験で使用したタスクは、タスク 0：京都で飲みに行く場所を探す、タスク 1：博多で観光する場所を探す、タスク 2：札幌で観光する場所を探す、タスク 3：金沢で食事する場所を探す、タスク 4：仙台で食事する場所を探す、というもので、タスク 0 はシステムに慣れるための練習用タスクであり、タスク 1 とタスク 2、タスク 3 とタスク 4 でペアとなっている。

6.2.2 協調検索後の客観評価

我々は、全ての実験を終えた後、推薦された全てのクエリの精度を評価した。各クエリに対して、そのクエリが推薦されるクエリとして適切であるかという尺度と、そのクエリがより広範囲を検索するのに役立つかまたは的を絞って狭い範囲を検索するのに役立つかという尺度で、それぞれ 5 段階で評価する。

被験者にはグループの検索意図（タスク）を見せ、実際に各時点でユーザが入力したクエリを提示した。それらを考慮し、先述の 2 つの尺度で、推薦されたクエリを評価してもらった。

6.3 実験結果

どのタスクについてもほぼ 10 分程度で結論に達した。実験の様子を撮影した映像と操作ログ、アンケート、クエリの精度の評価について分析を行った。

グループ毎の結論に達するまでに掛かった時間と発話数の平均を表 5 に示す。結論に達するまでに掛かった時間は提案システムの方が比較システムよりも短くなっており、10 分を切っている。つまり、提案システムの方が短時間でスムーズに意思決定に至ったことがわかる。また、グループ毎の平均発話数は、提案システムの方が比較システム比べて少なくなっている。

操作ログからのユーザの入力したクエリ数などを表 6 に示す。提案システムでは各ユーザに同じ時点では同じクエリを推薦するので、クエリ数もユニーククエリ数も比較システムの方がわずかに多くなっている。しかし、入力クエリのうち推薦クエリ

(注2): <http://developer.yahoo.co.jp/webapi/search/assistsearch/v1/webunitsearch.html>

表 6 クエリ数など

	クエリ数	ユニーククエリ数	入力クエリのうち推薦クエリの割合
提案システム	18.6 ($\sigma = 6.0$)	14.0 ($\sigma = 3.2$)	31.5%
比較システム	19.6 ($\sigma = 4.2$)	16.0 ($\sigma = 3.5$)	13.4%

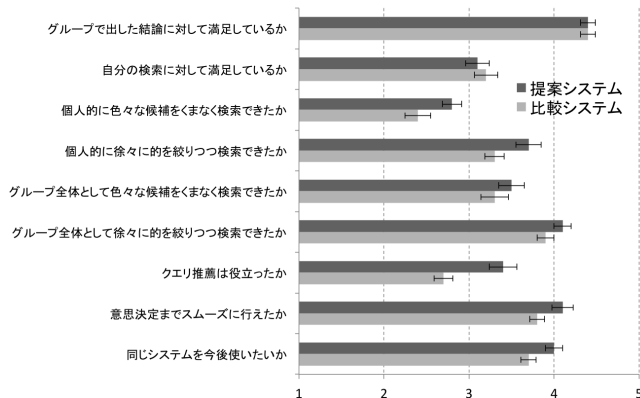


図 5 アンケート結果

から選択したものの占める割合は、提案システムで 31.5%、比較システムでは 13.4%と、提案システムの方が圧倒的に多く、より興味を示すクエリが提示されたということがわかる。

アンケートの結果を図 5 に示す。「グループで出した結論に対して満足しているか」、「意思決定までスムーズに行えたか」といった質問に対しては、両システムに有意差はなかった。しかし、「クエリ推薦は役立ったか」という質問に対しては、両システムで有意な差が見られ ($p < 0.05$)、提案システムの方が良い結果となった。また、「個人的に色々な候補をくまなく検索できたか」という質問に対しても有意差があり、提案システムの方が良い結果となった。

クエリの精度を 2 つの尺度で評価した結果を表 7 に示す。1 つ目の尺度は、そのクエリが推薦されるクエリとして適切であるかという尺度で、2 つ目の尺度は、そのクエリがより広範囲を検索するのに役立つかまたは的を絞って狭い範囲を検索するのに役立つかという尺度である。表中の「適合度」とは 1 つ目の尺度のことであり、不適切:1 ~ 適切:5 の 5 段階で評価してもらったものの平均である。表中の「収束度」とは 2 つ目の尺度のことであり、より広い範囲を検索するのに役立つ場合は 1、より的を絞って検索するのに役立つ場合は 5 の 5 段階で評価してもらったものの平均である。すなわち、値が低いほどより発散クエリらしい、値が高いほどより収束クエリらしいということになる。全推薦クエリについての平均、発散クエリとして推薦したクエリについての平均、収束クエリとして推薦したクエリについての平均を載せている。結果としては、全体の適合度はやや適合の傾向にあるが、収束クエリについてはわずかに高くなっている。収束度については、発散クエリの収束度は低く、収束クエリの収束度は高くなっており、つまり発散クエリは発散クエリとして適切で、収束クエリは収束クエリとして適切であったということがわかる。

また、協調検索を行った際にシステムが推定した各グループの

表 7 クエリの精度

	全クエリ	発散クエリ	収束クエリ
適合度	3.33	3.19	3.58
収束度	-	2.51	3.18

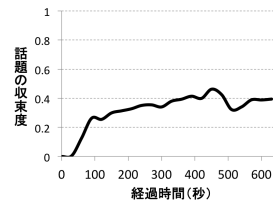


図 6 各グループの収束度の平均

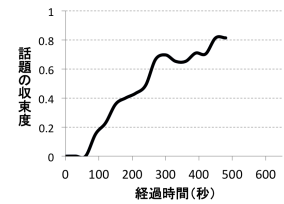


図 7 グループ 8 の収束度

収束度の平均の推移 (図 6) と、特に変化が顕著であったグループ 8 の収束度の推移 (図 7) の 2 つを示す。どちらのグラフも右肩上がりであり、網羅的に検索し、徐々に的を絞っていったということがわかる。

6.4 結論に達するまでの検索行動と意思決定に対する評価

グループで出した結論や自分の検索に対する満足度は、提案システムと比較システムでそれほど違いはない。しかし、個人の判断とグループ全体としての判断によらず、色々な候補をくまなく検索できたか、徐々に的を絞りつつ検索できたかという質問に対しては、提案システムの方が高く、比較システムを下回することはなかった。また、表 5 からわかるように、意思決定に至るまでの時間は提案システムの方が短く、10 分を切っている。つまり、提案システムの方が、ある程度の満足度を保ったまま、短時間で、色々な候補を検索した上で徐々に的を絞って検索できたということがわかる。

両者の満足度に差が出なかったため、今後はタスクを難しくしたり、時間をさらに制限することで、満足度に差が出るような実験を行おうと考えている。

6.5 クエリ推薦の有用性の評価

表 6 のようにクエリ数、ユニーククエリ数ともに提案システムの方が少なくなっている。これは、他のユーザが入力したクエリを元に推薦クエリを生成しているため、当然のことである。しかし、入力クエリのうちユーザが推薦クエリから選択したものの割合は提案システムの方が圧倒的に多くなっている。これは、推薦されたクエリに有用なものが多かったということが理由として考えられる。また、クエリ推薦は役に立ったかどうかという質問に対して、提案システムの方が良い結果が得られたことから、推薦クエリは有用であったと考えられる。

実際にクエリ推薦が有用にはたっているかどうかを撮影映像と操作ログから確認した。

事前アンケートでスマートフォンの入力に全く慣れていないと答えたユーザは、協調検索開始直後に入力に手間取っていた。そのとき、提案システムではクエリが推薦され、そのクエリを選択して検索を開始していた。3. の、入力の手間自体を解消するということが実現されているということである。また、このユーザは、グループ内で最もクエリ推薦を利用しており、アンケートの全ての項目において、提案システムの方が高いか比較システムと変わらないかであった。

あるグループのユーザは、札幌で観光する場所として、“札幌 時計台”を検索していた。その後“札幌 テレビ塔”というクエリが発散クエリとして推薦され、そこで「テレビ塔もあるのか」という発話がなされ、候補を広げることができた。さらに“札幌 大通公園”が発散クエリとして推薦され、「大通公園があるのか」という発話がなされ、さらに候補を広げることができていた。最終的に「時計台」も「テレビ塔」も「大通公園」から近いということが判明し、そのグループは「大通公園」を結論とした。このように、推薦クエリによって、ユーザの思いがつかない候補を提示できていた。

さらに、別のグループでは、金沢で食事する場所として“金沢 寿司”というクエリであるユーザが検索していた。そのとき、“金沢 海鮮丼”というクエリが推薦され、グループの数人が「海鮮丼」という候補で検索を始め、最終的に「海鮮丼」に的を絞り、「海鮮丼が美味しそう」という理由である店に決定した。

以上のように、提案システムでクエリ推薦が有効にはたらくことが多々あった。

6.6 クエリの精度

クエリの精度は表 7 のようになったが、それほど良い結果ではない。例えば“博多 観光”というクエリをユーザが入力したとき、システムは“博多 ビジネス”というクエリを推薦している。「博多に行く目的として観光だけでなくビジネスもありうる」という考えに基づくものであるが、元々の意図としては観光であるので、“博多 ビジネス”は不適切なクエリ推薦となる。現在の手法では、「博多」というおおまかな意図を推定することは可能であるが、粒度の小さい「旅行」、「観光」といった意図は考慮していない。そのような理由で、適合度は低くなってしまったと考えられる。また、適合度に関しては収束クエリの場合でわずかに高くなっているが、これは収束クエリがユーザの入力クエリ自身からそれに詳細語を加えるという生成の仕方であるために、グループの意図から大きく逸脱したようなクエリは推薦されなかったからだと考えられる。また、発散クエリと収束クエリの収束度の差が小さいので、今後はその差を広げていきたいと考えている。

7. ま と め

本論文では、複数ユーザがモバイル端末を用いて協調検索を行う際に、満足度を高めつつ意思決定をスムーズに進めるための手法を提案した。具体的には、意思決定がスムーズに進まない原因として、各ユーザが重複した検索行動をとることでグループとして網羅的に検索できないということと、各ユーザが自由に検索することでグループとしての的を絞った検索ができないということの 2 つがあると考えた。そのうえで、前者の問題を解決するために話題の発散を促すようなクエリを推薦し、後者の問題を解決するために話題の収束を促すようなクエリを推薦するという手法を提案した。

また、提案手法をプロトタイプシステムとして実装し、評価実験によって、色々な候補をくまなく検索できたか、クエリ推薦は役に立ったかというような尺度では提案手法の効果は示せ

た。特に、提案手法によるクエリ推薦については、実際の撮影映像や操作ログからもアンケートからも有用性があるということが判明した。

今後の課題としては、クエリ推薦の精度がそれほど良くなかったため、生成の仕方を改善する必要がある。具体的には、グループの意図をより粒度を細かく推定できるようにしたり、兄弟語・詳細語だけでない発散・収束クエリについて検討したりする必要がある。また、本論文では、グループ内の全てのユーザには同じクエリが推薦されていたが、ユーザ毎に検索を分担していたり、先行知識が違ったりと、全ユーザが平等であるとは限らないので、ユーザ毎に推薦するクエリを変更するというものについても考えている。

謝辞 本研究の一部は、文部科学省科学研究費補助金 挑戦的萌芽研究「モバイル協調検索に関する研究」(研究代表者: 中村 聡史, #22650018)、若手研究(A)「インタラクティブな再ランキング・再サーチを可能とする次世代検索に関する研究」(研究代表者: 中村 聡史, #23680006)、グローバル COE 拠点形成プログラム「知識循環社会のための情報学教育研究拠点」(拠点リーダー: 田中克己)によるものです。ここに記して謝意を表すものとします。

文 献

- [1] 総務省. 情報通信統計データベース 携帯電話・PHS の加入契約数の推移. <http://www.soumu.go.jp/johotsusintokei/field/data/gt01020101.xls>.
- [2] 社団法人中央調査社. 2011 年パーソナル先端商品の利用状況 (第 26 回). <http://www.crs.or.jp/data/pdf/ptg2011.pdf>.
- [3] Meredith Ringel Morris. A survey of collaborative web search practices. In *Proceedings of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, CHI '08, pages 1657–1660. ACM, 2008.
- [4] Meredith Ringel Morris and Eric Horvitz. Searchtogether: an interface for collaborative web search. In *Proceedings of the 20th annual ACM symposium on User interface software and technology*, UIST '07, pages 3–12. ACM, 2007.
- [5] Saleema Amershi and Meredith Ringel Morris. Cosearch: a system for co-located collaborative web search. In *Proceedings of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, CHI '08, pages 1647–1656. ACM, 2008.
- [6] 伊豆 陸, 中島 伸介, and 田中 克己. グループ支援型 web 閲覧における閲覧履歴の視覚化と共有. 日本データベース学会 *letters*, 3(1):121–124, 6 2004.
- [7] 伊豆 陸, 中島 伸介, and 田中 克己. グループ支援型 web 探索におけるナビゲーションのための既閲覧ページ群の同期化提示. 日本データベース学会 *letters*, 3(2):13–16, 9 2004.
- [8] 奥 梓, 小牧 大治郎, 荒瀬 由紀, 原 隆浩, 上向 俊晃, 服部 元, and 西尾 章治郎. 携帯端末を用いた協調 web 検索におけるコンテンツ比較支援インタフェース. In *DEIM Forum 2010 D9-4*, 2010.
- [9] 小谷 大祐, 中村 聡史, and 田中 克己. モバイル協調検索におけるユーザ間の情報共有支援. In *DEIM Forum 2011 A6-6*, 2011.
- [10] 大島 裕明, 小山 聡, and 田中 克己. Web 検索エンジンのインデックスを用いた同位語とそのコンテキストの発見. 情報処理学会論文誌. データベース, 47(19):98–112, 12 2006.
- [11] 野田 武史, 大島 裕明, 小山 聡, 田島 敬史, and 田中 克己. 主題語からの話題語自動抽出とこれに基づく web 情報検索 (情報抽出). 電子情報通信学会技術研究報告. DE, データ工学, 106(149):239–244, 7 2006.