クエリログの時系列情報に基づくキーワード修正リスト生成手法

平手 勇宇　　竹中 孝真
楽天株式会社　楽天技術研究所
{yu.hirate, takamasa.takenaka}@mail.rakuten.co.jp

1　はじめに

近年広く利用されている検索エンジンは，インデックスに存在するキーワードの入力をユーザに要求する．しかし，ユーザ入力キーワードがインデックス上に存在しない場合，検索結果数は 0 件となり，その時点で当該ユーザの検索行動は停止する．このとき，検索エンジン側から適切なキーワードを提示できれば，ユーザは検索行動を継続することができ，ユーザビリティの向上に寄与する．したがって，検索エンジンにおいて，修正キーワード候補を提示するシステムは大変重要である．

検索キーワードの自動修正アルゴリズムは，これまでに多くの研究事例 [1][2][3][4] が存在し，Web 検索エンジンのモジュールとしても実装されている．これらの既存アルゴリズムは，誤りキーワードの文字列と正しいキーワードの文字列表記が似ているという条件の下，誤りのある入力キーワードに対し，正しいキーワード集合から距離の小さいキーワードを選択するアプローチを取っている．したがって，例えば“桃らー”→“辛そうで辛くない少し辛いラー油”といったように，誤りのある入力キーワードと正しいキーワードの距離が大きいが，検索キーワード修正として有用なキーワードペアを抽出することが困難である．

この問題を解決するため，本稿では，クエリログに記録されているユーザのキーワード修正行動を検出することで，修正前キーワードと修正後キーワードの距離が大きいキーワード修正リストを抽出することを目的とする．ここで，キーワード修正行動とは，同一ユーザによる以下の行動を指す（図 1）．

1. あるキーワードで検索し，検索結果数 0 の結果を獲得
2. その後，キーワードを変更し，検索結果数 1 以上の結果を獲得

クエリログの時系列情報からユーザのキーワード変更行動を抜き出し知識獲得を行う研究として，文献 [5] があげられる．文献 [5] では，ある一定時間内に共起するキーワード集合を対象として，関連語を抽出することを目的としている．本稿で目的とするキーワード修正リスト集合を抽出する課題とは異なる．

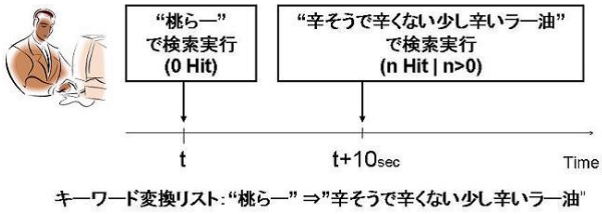

図 1　ユーザのキーワード修正行動の例

次節以降，第 2 節では提案手法であるクエリログの時系列情報に基づくキーワード修正リスト生成手法を説明する．第 3 節で評価実験を行い，第 4 節でまとめを記述する．

2　時系列情報に基づくキーワード修正リスト生成手法

提案手法は，第 1 節で述べたとおり，検索クエリログからユーザのキーワード修正行動を検知し，キーワード修正リストとして抽出する手法であり，(1) 検索クエリログからキーワード修正行動に着目したキーワードペアの抽出，及び (2) 抽出したキーワードペア集合からノイズペアの除去の 2 ステップで構成されている．本節では，以下，キーワードペアの具体的な抽出方法，及びノイズの除去方法について述べる．

2.1　キーワードペアの抽出方法

本手法では，同一 IP アドレスから時間間隔をあけずして検索されたキーワード集合は，同一ユーザ，同一検索意図の下で発行されていると考える．そして，ある IP アドレスから検索結果数 0 件を返すキーワードで検索された後，一定時間内に同一 IP アドレスから検索結果数 1 件以上を返すキーワードで再度検索された場合，前者のキーワードを誤りキーワード，後者のキーワードを正しいキーワードと考え，これら二つのキーワードを，キーワード修正リスト生成のためのキーワードペアとして抽出する．ただし，検索結果数 1 件以上のキーワードが，検索結果数 0 件のキーワードの部分文字列である場合，この行動は検索条件を緩める行動であると解釈できるので，抽出キーワードペアの対象としない．

具体的には，検索リクエスト s の条件を表 1 のように

表 1　検索リクエスト s の条件

symbol	description
$s.query$	s のキーワード
$s.ip$	s のリクエスト発行元の IP アドレス
$s.timestamp$	s が発行された時間
$s.num$	s の検索結果数
$s.genre$	s のジャンル指定条件

定義したとき，検索リクエスト s_1, s_2 が，以下に示す条件を全て満たすとき，$s_1.query \rightarrow s_2.query$ を，キーワードペアとして抽出する．

$$s_1.ip = s_2.ip \quad (1)$$
$$s_2.timestamp - s_1.timestamp \leq \Delta t(sec) \quad (2)$$
$$s_1.num = 0 \wedge s_2.num \geq 1 \quad (3)$$
$$s_1.genre = s_2.genre = \phi \quad (4)$$
$s_2.query$ は，$s_1.query$ の部分文字列でない　(5)

ここで，Δt は隣接する検索リクエストの時間間隔上限値を定めるパラメータである[*1]．

2.2　相関ルールを用いたノイズ除去

前述した方法によって抽出したキーワードペアには，キーワード修正リストとして無効であるノイズペアが多数混入している．その理由は，Proxy サーバ，DHCP サーバ等，同一 IP アドレスを複数人で共有するケースや，短時間でユーザが検索意図を変更するケースが存在するためである．

そこで，提案手法は相関ルール [6] を用いてノイズを除去する．具体的には，最小サポート値制約，最小確信度制約の 2 つの制約条件を定義し，両者の制約条件を満たすキーワードペアのみを，キーワード修正リストの要素とする．

■最小サポート値制約　キーワードペア $(s_1.query \rightarrow s_2.query)$ のサポート値 $(= sup(s_1.query \rightarrow s_2.query))$ とは，$s_1.query \rightarrow s_2.query$ のキーワード変更が観測された IP アドレスの異なり数とする[*2]．

最小サポート値制約とは，あまり観測されないキーワードペアをキーワード修正リストに登録しないようにするための制約であり，式 (6) にて定義される．

$$sup(s_1.query \rightarrow s_2.query) \geq min\_sup \quad (6)$$

ここで，min_sup は最小サポート値であり，ユーザ定義のパラメータである．

■最小確信度制約　キーワードペア $(s_1.query \rightarrow s_2.query)$ の確信度 $(= conf(s_1.query \rightarrow s_2.query))$ とは，$s_1.query$ で検索した IP アドレスのうち，$s_2.query$ で再度検索した IP アドレスの割合を表し，式 (7) にて定義される．

$$conf(s_1.query \rightarrow s_2.query) = \frac{sup(s_1.query \rightarrow s_2.query)}{sup(s_1.query)} \quad (7)$$

最小確信度制約は，信頼が低いキーワードペアをキーワード修正リストに登録しないようにするための制約であり，式 (8) にて定義される．

$$conf(s_1.query \rightarrow s_2.query) \geq min\_conf \quad (8)$$

ここで min_conf は最小確信度であり，ユーザ定義のパラメータである．

3　評価実験

本節では，楽天市場 [7] の 2010 年 05 月のクエリログデータを対象として，提案手法を適用し，キーワード修正リストの生成を実施した．

3.1　距離が近いキーワードペアの除去

本手法は，キーワード間の文字列表記上の距離が大きく，既存アルゴリズムで抽出が困難であるキーワード修正リストを生成することである．よって，キーワード間の距離が小さいリストは，既存のアルゴリズムでも抽出可能と判断し，本実験の評価対象外とした．具体的には，[4] で示している下記のキーワード距離関数 $D(k_1, k_2)$ が 0.2 以下のキーワードペアを評価対象外とした[*3]

$$D(k_1, k_2) = 0.2(1 - Jaro(k_1, k_2)) + 0.8(1 - Jaro(k_1.kana, k_2.kana)) \quad (9)$$
$$Jaro(k_1, k_2) = \frac{1}{3}\left(\frac{m}{|k_1|} + \frac{m}{|k_2|} + \frac{m-t}{m}\right) \quad (10)$$

ここで，$k.kana$:k の読み仮名[*4], $|k|$:k の長さ，m:k_1 と k_2 の共通文字数，t:k_1 から k_2 への転置数とする．

3.2　抽出数の実験

時間間隔パラメータ Δt の値を，$\Delta t = \{10, 20, 30, 40, 50, 60, 70, 80, 90\}$，最小サポート値パラメータを $3 \leq min\_sup \leq 10$，最小確信度パラメータを $0 \leq min\_conf \leq 1$ と設定し，キーワード修正リストの生成を行った．

図 2 に，$\Delta t = 90$ とした場合を例にとり，min_sup 及び min_conf の値によるキーワードペア抽出数の分布を示す．図 2 に示すとおり，min_conf の値を大きくとれば取るほど，min_sup の値を小さくすればするほど抽出キーワードペア数が増加していることが分かる．また，

[*1] 具体的な数字は，第 3 節にて述べる．
[*2] ロボットによる検索エンジンアクセスの影響を低減させるために，観測回数ではなく，観測 IP アドレス数の異なり数とした．
[*3] [4] の手法によって抽出されるキーワード修正リストの 95% 以上が，キーワード距離関数 $D(k_1, k_2)$ の値が 0.2 以下であったため，0.2 を閾値として設定した．
[*4] 読み仮名は，kakasi を用いて変換を行った [8]．

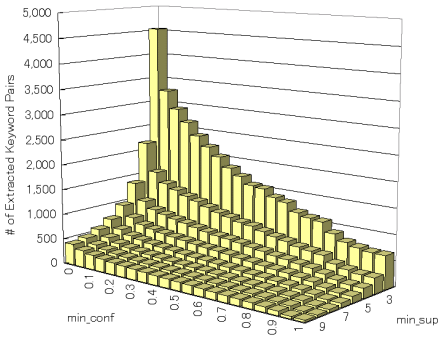

図 2　$\Delta t = 90$ とした時のキーワードペア抽出数の分布

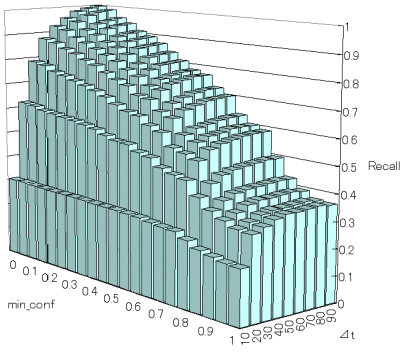

図 3　$min\_sup = 3$ とした時の再現率分布

Δt の値を大きくすればするほど，時間間隔の制約が緩くなるため，抽出キーワード数が増加していった．

パラメータを $(\Delta t, min\_sup, min\_conf) = (90, 3, 0)$ とした時，抽出キーワードペア数は最大となり，その値は 4,537 となった．

3.3　パラメータ設定実験

本実験では，パラメータ $(\Delta t, min\_sup, min\_conf)$ の最適な組み合わせを見出すことを目的とする．

■適合率，再現率の定義　$(\Delta t, min\_sup, min\_conf) = (90, 3, 0)$ とした時に抽出された 4,537 個のキーワードペア全てを，修正キーワードリストとして適切か否かを人手によりチェックを行った[*5]．その結果，4,537 個のキーワードペアのうち，1,629 個が適切なキーワードのペアであった．

本実験では，あるパラメータセットによって抽出されたキーワードペア数を α，その中で適切と判断されたキーワードペア数を β とした時，適合率は β/α，再現率は $\beta/1,629$ にて計算することとする．

■Δt の設定　パラメータ Δt は，時間間隔幅の上限値を定義するパラメータであり，主に計算コストに影響を及ぼすパラメータである．Δt が大きくなればなるほどメモリ使用量が増大する．よって，必要充分なキーワード修正リストが抽出できる範囲で，できる限り小さな Δt を選ぶことが重要である．

図 3 に，$min\_sup = 3$ とした際の $\Delta t, min\_conf$ の値による再現率の分布を示す．図 3 では，Δt の増加に伴い再現率が高くなることが確認できる．$min\_sup = 3$ において，$\Delta t \geq 60$ を満たす時，再現率が 95% 以上となることより，ユーザのキーワード修正行動のほとんどが 60 秒以内に完了すると考えられる．したがって，本実験では Δt の適正値は 60 であるとする．

■min_sup, min_conf の設定　Δt の設定の議論を受け，$\Delta t = 60$ した時の最適な min_sup, min_conf の組み合わせを抽出する．ここで，min_sup は，キーワードペアの最低出現 IP アドレス数を定義するパラメータであるため，主に抽出結果の再現性に影響を及ぼす．また，min_conf は確信度であるため，主に抽出結果の適合性に影響を及ぼす．提案手法は，一定以上の適合率を担保した上で，正確なキーワード修正リストをできる限り多く抽出したいため，適合率が 80% 以上を示す (min_sup, min_conf) のセットから，最も高い F 値を示す (min_sup, min_conf) セットを抽出する．

図 4 の (a) に，$\Delta t = 60$ 時の適合率の分布を示す．凡例に示すとおり，図 4 では，精度 80% 未満のパラメータセットは黒い網掛のグラフにて表示している．図 4 の (a) に示すとおり，$min\_conf <= 0.15$ の範囲では，任意の min_sup に対し適合率が 80% 未満であり，逆に，$min\_sup >= 0.45$ の範囲では，任意の min_sup に対し適合率が 80% 以上である．

適合率が 80% 以上を示す任意の (min_sup, min_conf) の組み合わせに対し，再現率，F 値を計算したところ，それぞれ図 4 の (b),(c) となった[*6]．図 4 の (c) において，F 値が最大となった組み合わせは，$min\_sup = 3, min\_conf = 0.45$ の時であり，適合率・再現率・F 値がそれぞれ，82.7%，75.9%，0.791 であった．

3.4　抽出例

最後に，本手法によって抽出されたキーワード修正リストの例を表 2 に示す．表 2 に示すように，本手法では“原田ラスク”→“ガトーフェスタ・原田”や“モモラー”→“辛そうで辛くない少し辛いラー油”のように，修正前と修正後キーワード間の距離が大きいキーワード修正リストの抽出に成功していることが確認できる．

4　おわりに

本稿では，検索クエリログに記録されているユーザのキーワード修正行動を検出することで，キーワード間の距離に基づくキーワード修正アルゴリズムでは抽出が困難であるキーワード修正リストを抽出する手法を提案した．具体的には，同一 IP から短時間に複数回実行され

[*5] 任意のパラメータセットによって抽出されるキーワードペア集合は，4,537 個のキーワードペア集合のサブセットである．

[*6] 図 4 の (b) と (c) では，便宜上適合率が 80% 未満のパラメータの組み合わせに対しても，再現率，F 値を示している．

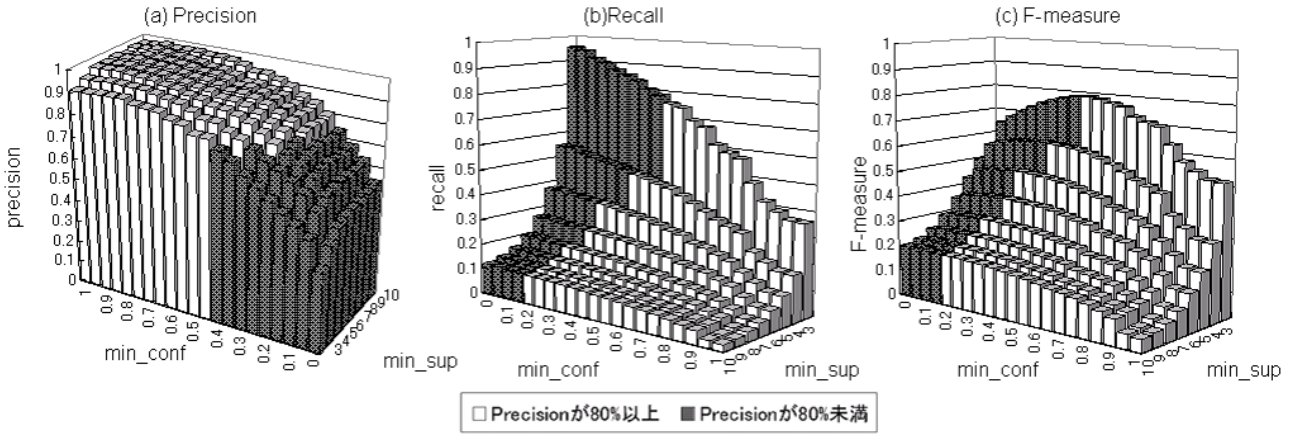

図 4　$\Delta t = 60$ とした時の適合率 (a), 再現率 (b) および F 値 (c)

表 2　抽出された修正キーワードリストの例

訂正前キーワード	訂正後キーワード	サポート値	キーワード間距離 [4]	確信度
原田ラスク	ガトーフェスタ・ハラダ	114	0.4721	0.613
ハラダラスク	ガトーフェスタ・ハラダ	92	0.4754	0.601
モモラー	辛そうで辛くない少し辛いラー油	75	0.7274	0.6
桃らー	辛そうで辛くない少し辛いラー油	60	0.7274	0.577
こうげんどう	江原道	47	0.7174	0.887
スナッフルス	チーズオムレット	39	0.653	0.52
桃屋のラー油	辛そうで辛くない少し辛いラー油	33	0.6339	0.623
こっかえん	国華園	30	0.6861	1
年輪屋	ねんりん家	25	0.4167	0.926
ハニーラボ	山田養蜂場	24	0.663	0.453
マテリアルフォース	ミクロマン	20	0.4578	0.645
くるくる本舗	まつげパーマ	19	0.6919	0.463

ている検索集合から，修正前と修正後キーワードペアを生成する．さらに，提案手法では，最小サポート値制約，最小確信度制約を用いて，ノイズとなるキーワードペアを除去する．

実験の結果，ユーザによるキーワード変更行動は，60 秒以内に殆ど終了することが分かった．さらに，$\Delta t = 60$, 適合率 80% 以上の条件下では，$min\_sup = 3$, $min\_conf = 0.45$ とした時，最も F 値が高い結果となり，1,236 個のキーワード変換リストを抽出することができた．

参考文献

[1] M. Li, Y. Zhang, M. Zhu, and M. Zhou, “Exploring distributional similarity based models for query spelling correction,” in Proc of ACL 2006, pp. 1025–1032, 2006.

[2] B. Martins and M. J. Silva, “Spelling Correction for Search Engine Queries”, In Proc. of 4th Int’l Conf on EsTAL 2004, pp. 372–383, 2004.

[3] F. Ahmad and G. Kondrak, “Learning a spelling error model from search query logs,” In Proc. of HLT/EMNLP’05, pp.955–962, 2005.

[4] 平手 勇宇, 竹中 孝真, 森 正弥, “キーワード型検索エンジンにおける修正キーワード提示アルゴリズム”, 日本データベース学会論文誌 Vol.9, No.1, pp.23-28, 2010.

[5] B. M. Fonseca, P. B. Golgher, E. S. de Moura and N. Ziviani, “Using association rules to discover search engines related queries,” In Proc. of LA-WEB 2003, pp. 66–71, 2003.

[6] R. Agrawal and R. Srikant, “Fast algorithms for mining association rules in large databases,” In proc. of VLDB’94, pp.487–499, 1994.

[7] 楽天市場，`http://www.rakuten.co.jp`

[8] KAKASI - 漢字→かな (ローマ字) 変換プログラム，`http://kakasi.namazu.org/`