キーワード型検索エンジンにおける修正キーワード候補提示アルゴリズム

平手　勇宇† 　竹中　孝真† 　森　正弥†

† 楽天株式会社 楽天技術研究所　〒 140–0002 東京都品川区東品川 4-12-3 品川シーサイド楽天タワー
E-mail: †{yu.hirate,takamasa.takenaka,masaya.mori}@mail.rakuten.co.jp

あらまし　今日幅広く用いられている転置インデックスに基づくキーワード検索エンジンは，ユーザに転置インデックスに存在するキーワードの入力を要求する．そのため，インデックス上に存在しないキーワードが入力された場合，転置インデックス上に存在する適切なキーワード候補を提示し，ユーザを支援することは重要である．このような背景のもと，本稿では，クエリログデータを基にした修正キーワード候補提示アルゴリズムを提案する．提案手法は，入力キーワードに対する修正キーワード候補のスコアリング手法，およびスコアリングされた修正キーワード候補集合から，ユーザに提示すべき適切なキーワードリストを生成する手法で構成されている．実験により，提案手法によりユーザが入力した転置インデックス上に存在しないキーワードに対する，適切なキーワード候補を提示することが可能であることを確認した．

キーワード　キーワード型検索エンジン，クエリキーワード修正，漢字誤変換，スペルミス，表記揺れ，Jaro 距離

Keyword Correction Algorithm in Keyword Based Search Engine

Yu HIRATE†, Takamasa TAKENAKA†, and Masaya MORI†

† Rakuten Institute of Technology, Rakuten, Inc.　Shinagawa Seaside Rakuten Tower, 4-12-3 Higashishinagawa, Shinagawa-ku, Tokyo, 140–0002, Japan
E-mail: †{yu.hirate,takamasa.takenaka,masaya.mori}@mail.rakuten.co.jp

Abstract　Keyword Based Search Engines based on inverted index require users to specify keywords which are indexed in their inverted indexes. Therefore, when users fail to specify indexed keywords, it is important to assist user by offering appropriate indexed keywords instead of user-input keywords. We propose a novel keyword correction algorithm using query log data for keyword based search engines. Our proposed algorithm consists of two phases: scoring candidate keywords for correction, and generating offering keyword list based on scored candidate keywords. Our evaluation shows the proposed algorithm is able to offer a list of appropriate indexed keyword to users.

Key words　Keyword Based Search Engine, Correcting Query Keyword, Kanji Conversion Error, Spelling Error, Spelling in Multiple Ways, Jaro Distance

1. はじめに

近年，Web ページに代表される多数のドキュメントから適切なドキュメントを検索するシステムとして，転置インデックスを利用した検索エンジンが幅広く用いられている．転置インデックスを利用した検索エンジンは，ユーザに転置インデックスに存在するキーワードの入力を要求する．しかし，ユーザの入力したキーワードが転置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止まってしまう．実際，一般的な Web ページ検索エンジンでは，約 10〜12%のクエリキーワードがスペルミスであると報告されており [1]，ユーザの入力したキーワードが転置インデックス上に存在しない割合が多いと考えられる．

ユーザが転置インデックス上に存在しないキーワードを入力した場合，転置インデックス上に存在するキーワード集合の中から，ユーザ入力キーワードと似ている適切なキーワードを提示するができれば，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を実現することは，大変重要である．

これまでに，Web 検索エンジン各社 [2] [3] [4] は，上記の課題に対応するキーワード修正アルゴリズムを構築し，当該アルゴリズムを検索エンジンに組み込むことでユーザビリティを向上させている．しかしながら，著者らの知る限りではそのアル

ゴリズムの詳細は公表されていない．

そこで，本稿では，EC サイト [5] [6] の商品検索のクエリログを用いて，修正キーワード候補提示アルゴリズムを提案する．提案手法は，ユーザ入力キーワードに対する修正キーワード候補のスコアリング手法，およびスコアリングされた修正キーワード集合からユーザに提示する提示キーワードリストを生成する手法の 2 つで構成されている．

本稿では次節以降，第 2 節にて関連研究を示し，第 3 節にて提案手法の概要について示す．第 4 節，第 5 節では，提案手法のうちのスコアリング手法，および提示キーワードリスト生成手法についてそれぞれ説明する．第 6 節では，実験結果と提案手法の精度向上について述べ，第 7 節でまとめを記述する．

2. 関 連 研 究

テキスト上のスペルミスを自動的に検出する手法は，古くから研究されている [7]．編集距離を用いたスペル修正アルゴリズム [8] [9] や，n-gram の確率的な手法を用いたアルゴリズム [10] などが挙げられる．これらの研究成果は，Web に限らず幅広いツールに適用されている．

一方，検索エンジンにおける修正キーワード候補提示手法に関しては，検索エンジン各社 [2] [3] [4] が，自社の検索サービスに組み込んでいるが，著者の知る限りではその手法は公開されてない．公開されている手法では，ルールベースで修正キーワード候補を提示するシステム [11] が存在する．[11] では，ユーザ入力キーワードと比較し，あらかじめ定義された制約を満たす候補キーワードを抽出し，抽出された候補キーワード集合から，ヒューリスティクスな手法を用いて提示単語を選択する手法をとっており，ポルトガル語を対象として検証を実施している．しかしながら，日本語のキーワード誤りには，表記ゆれ，スペルミスのほかに，漢字誤変換によるものが存在するため，[11] の手法をそのまま適用しても，よい精度は出ないと考えられる．

3. 修正キーワード候補提示アルゴリズム概要

本節では，提案手法における提示キーワードの条件，および提案アルゴリズムの概要を述べる．

3.1 提示キーワードの条件

提案手法である修正キーワード候補提示アルゴリズムは，図 1 で示すように，ユーザが転置インデックス上にないキーワードを入力した際，提示候補となる候補キーワード集合からユーザに提示する．修正キーワード候補を提示する際，以下の 3 つの条件を考慮する必要がある．

距離条件　タイプミス，漢字誤変換，表記揺れに代表されるように，提示キーワード文字列は，ユーザ入力キーワードの文字列と似ていることが必須である．したがって，ユーザ入力文字列と修正キーワード候補文字列の距離を計算し，その距離が小さい修正キーワード候補を優先的に提示する必要がある．

検索頻度条件　高頻度で検索されているキーワードの提示優先度を上げ，低頻度で検索されているキーワードの提示優先度を下げる必要がある．

図 1　修正キーワード候補提示の例

検索可能性条件　修正キーワード候補提示アルゴリズムの本質は，検索結果 0 件となる検索を回避することである．したがって，提示したキーワードでどれだけの検索結果数が得られるかを考慮することは重要である．

3.2 修正キーワード候補集合の生成

提案手法は，まずクエリログに存在するキーワードから，修正キーワード候補集合 $W = \{w_1, w_2, \cdots, w_N\}$ を生成しておく．修正キーワード候補集合のエントリ $w_i (1 \leqq i \leqq N)$ は，クエリログ中に出現するキーワードのうち，1 件以上の検索結果を返すキーワードであり，個々のエントリは以下の情報にて構成されている．

（1） キーワード文字列
（2） キーワードの「読み」
（3） クエリログ中の出現回数
（4） 当該キーワードで検索した際の検索結果数

3.3 修正キーワード候補のスコアリング

ユーザから検索結果数 0 件のキーワード u を受け取ると，提案手法は，修正キーワード候補集合 W の要素のうち，当該要素の文字列長と u の文字列長が近い修正キーワード候補を選択し，ユーザ入力キーワード u に対するスコア $score(w_i|u)$ を計算する．具体的なスコア計算式は，第 4 節にて記述する．その後，スコア降順にスコア計算対象の修正キーワード候補を並び替え，提示キーワード候補リスト $C =< c_1, c_2, \cdots, c_n >\ (n \leqq N)$ とする．

3.4 提示キーワードリストの生成

ユーザに提示する提示キーワードリストを生成する際，単純に提示キーワード候補リスト C のスコア値上位 k 件を選択するアプローチをとると，入力キーワード u に対して全く関係のないキーワードをユーザに提示する場合がある．例えば，u の提示キーワードとして適切な修正キーワードが 1 件しかなかった場合，残りの $k-1$ 件の提示キーワードは u とは関係のないキーワードとなってしまう．

本手法ではこの問題点を解決するために，単純に提示キーワード候補リスト C のスコア値上位 k 件を提示するのではなく，スコア降順にランキングされた提示キーワード候補リストから，適切な要素数の提示キーワードリスト $R =< r_1, \cdots, r_m >\ (m \leqq n)$ を生成し，ユーザに提示する．ただし，提示キーワードが存在しない場合は，$R = \phi$ となる．具体的な手法は，第 5 節に示す．

4. 修正キーワード候補のスコアリング手法

本節では，ユーザ入力キーワード u と修正キーワード候補集合 W から，提示キーワード候補リスト C の生成するために必要となる修正キーワード候補のスコアリングについて言及する．3.1 に記述した 3 つの条件を基にして，ユーザ入力キーワード u に対する修正キーワード候補 w_i のスコア $Score(w_i|u)$ は，式 (1) にて定義される．

$$score(w_i|u) = \frac{Pr(w_i) + \alpha}{D(w_i, u) + \beta} \cdot A(w_i) \qquad (1)$$

ここで，$D(w_i, u) + \beta$，$Pr(w_i) + \alpha$，$A(w_i)$ はそれぞれ，距離 (=Distance) 条件，検索頻度 (=Probability) 条件，検索可能性 (=Availability) 条件を表現する項である．$score(w_i|u)$ が大きな値であればあるほど，入力キーワード u に対する提示キーワードとして確からしいキーワードであることを意味し，$score(w_i|u)$ が小さな値であればあるほど，確からしくないキーワードであることを意味する．本節では以降，距離条件，検索頻度条件，検索可能性条件の項について具体的に言及する．

4.1 距離条件

距離条件とは，ユーザ入力文字列 u と修正キーワード候補 w_i の類似性を判断する条件であり，$D(w_i, u) + \beta$(以下，距離条件項とする.）によって定量的に表現する．u と w_i が似ている場合には距離条件項は小さな値になり，$score(w_i|u)$ の値増加に寄与する．

ここで，日本語のキーワード検索において，検索結果 0 件となる原因の 1 つに漢字の誤変換が存在する．したがって，本研究におけるキーワードの距離は，式 (2) に示す様に u,w_i のキーワード文字列距離に加え，u の読みと w_i の読み(注1)の距離も考慮することにする．

$$\begin{aligned} D(w_i, u) \;=\; & a\,(1 - Jaro(w_i, u)) + \\ & b\,(1 - Jaro(w_i.yomi, u.yomi)) \qquad (2) \\ where \quad & a + b = 1 \end{aligned}$$

ここで，$Jaro(w_i, u)$ は，w_i と u の Jaro 距離 [13] を表し，$Jaro(w_i.yomi, u.yomi)$ は，w_i の読みと u の読みの Jaro 距離 [13] を表す．パラメータ a,b は，文字列の距離を測る際,「読み」をどれだけ強くするのかのパラメータである．文字列が同じ場合，同じ読みに変換されることを考え，b は比較的大きな値を設定する．

文字列 $str1$ と $str2$ の Jaro 距離は，式 (3) の様に表される．

$$Jaro(str1\ ,\ str2) = \frac{1}{3}\left(\frac{m}{|str1|} + \frac{m}{|str2|} + \frac{m-t}{m}\right) \qquad (3)$$

$where$ $|str1|$: $str1$ の長さ　$|str2|$: $str2$ の長さ

m : $str1$ と $str2$ の共通文字数

t : $str1$ から $str2$ への転置数

ここで，$0 \leqq Jaro(str1, str2) \leqq 1$ であるため，$D(w_i, u)$ の定義域は $0 \leqq D(w_i, u) \leqq 1$ である．u と w_i が同一文字列であった場合 $D(w_i, u) = 0$，u と w_i が全く違った文字列であった場合，$D(w_i, u) = 1$ となる．

なお，β はパラメータであり，0 よりも大きい実数が代入される．パラメータ β 設定の目的は，以下の 2 つである．

（1） $D(w_i, u)$ の $score(w_i|u)$ に対する影響度を調節するため．

（2） 距離条件項の値が 0 になることを避けるため．

(注1)：任意の文字列の読みを生成する手段として，本研究では KAKASI [12] を用いた．

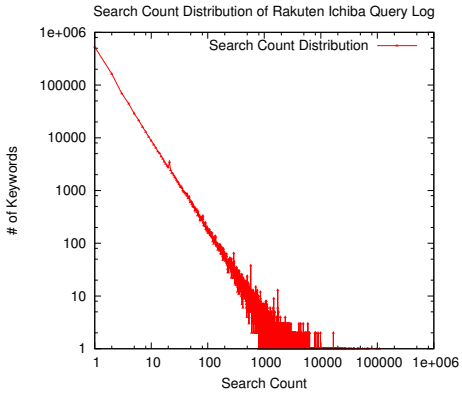

図 2　楽天市場 [5] におけるキーワード検索回数とキーワード数の分布

4.2 検索頻度条件

検索頻度条件とは，修正キーワード候補 w_i の検索頻度を判断する条件であり，$Pr(w_i) + \alpha$（以下，検索頻度条件項とし，α は定数項である.）によって定量的に表現する．w_i がよく検索される場合には検索頻度条件項は大きな値となり $score(w_i|u)$ の値増加に寄与する．

一般に検索エンジンにおいて，図 2 に示す様に，キーワード検索回数とキーワード数の関係は冪乗則に従う．そこで，$Pr(w_i)$ は，クエリログ中の w_i の出現回数 $SC(w_i)$ の対数をとることで定義する (式 (4))．

$$Pr(w_i) = log_{10} SC(w_i) \qquad (4)$$

なお，α は，ほとんど検索されない単語に対しても一定値以上のスコアを付与するためのパラメータである．

4.3 検索可能性条件

検索可能性条件とは，修正キーワード候補 w_i の検索可能性を判断する条件であり，$A(w_i)$ によって定量的に表現し，検索可能性を保証する．一般的な Web ページ検索エンジンでは，式 (5) で示すように，提示キーワードによる検索結果数が多ければ多いほど，$A(w)$ に大きな値を付与することで大きなスコアとすればよいと考えられる．また，検索可能性を保障するだけでよいのであるならば，式 (6) のように定めれば良い(注2)．

$$A(w_i) \;=\; log_{10}(RN(w_i) + 1) \qquad (5)$$

$$A(w_i) \;=\; \begin{cases} 1 (RN(w_i) \geqq 1) \\ 0 (RN(w_i) = 0) \end{cases} \qquad (6)$$

$where$ $RN(w_i)$: キーワード w_iによる検索結果数

(注2)：$Availability(w_i)$ の設計については，第 6 節にて言及する．

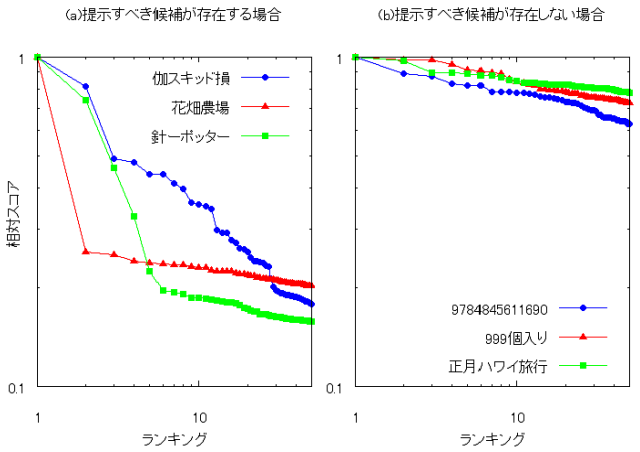

図 3　修正キーワード候補の相対スコア分布（両軸対数）

4.4　提示キーワード候補リストの生成

上述したスコア関数によって，ユーザ入力キーワード u に対する修正キーワード候補 $w_i|1 \leqq i \leqq N$ のスコア値 $score(w_i|u)$ が定義される．しかしながら，ユーザ入力キーワード u と修正キーワード候補 w_i の文字列長が大きく違う場合，距離条件項の値が大きくなり，$score(w_i|u)$ が小さな値となるのは自明である．したがって，提案手法ではスコア値を計算する修正キーワード候補を，$w_i \mid ||w_i| - |u| \leqq 4|$ に限定することで，高速化を図っている．ここで，$|w_i|, |u|$ は，文字列 w_i, u の長さである．

スコア計算対象の修正キーワード候補のスコア値の算出後，スコア計算対象の修正キーワード候補をスコア値降順に並び替えることで，提示キーワード候補リスト $C =< c_1, c_2, \cdots, c_n >$ を生成する．つまり，提示キーワード候補リスト C は，$score(c_i|u) \geqq score(c_j|u) \mid (1 \leqq i < j \leqq n)$ を満たす．

5.　適切な提示キーワードリストの生成手法

3.4 で示したとおり，第 4 節で示したスコア関数によって生成された提示キーワード候補リスト C の上位 k 件を提示キーワードとした場合，ユーザ入力キーワードとは全く関係のないキーワードを提示してしまう可能性が存在する．したがって，ユーザ入力キーワード u によって，提示キーワードリスト R を提示キーワード候補リスト C より自動的に選択しなくてはならないし，場合によっては候補キーワードを一切出さない判断もしなくてはならない．そこで，本手法では，提示キーワード候補リスト C のスコア値リストから，適切な要素数の提示キーワードリスト R の生成を行う．本節では，提示キーワードリスト R の生成方法について述べる．

5.1　修正キーワードのスコア値分布

提示キーワードリスト R を生成するにあたり，提案手法は提示キーワード候補リスト C のスコア値分布を考える．提案手法では，R を生成するにあたり，以下に示す仮定を行う．

- ユーザビリティを考慮すると R 要素数は高々5 個であり，C の要素数に比べ極めて小さい．
- R に属する要素は，高いスコア値を持つ．
- R に属しない多数の C の要素は，低いスコア値を持つ．

したがって，ユーザに提示すべきキーワードが存在する場合には，提示キーワード候補リスト C は，スコアの高いごく少数の要素とスコアの低い多数の要素で構成される．逆に，提示すべきキーワード候補が存在しない場合には，全ての提示キーワード候補リスト要素のスコアは低い値となる．

実際に，提示キーワードリストが存在すべきユーザ入力キーワードとして，「伽スキッド損」「花畑農場」「針ーポッター」(注3) を取り上げ，これら 3 つをユーザ入力キーワードとした際の C の上位 50 件の相対スコア値分布を図 3 の左側 (a) に示す．また，提示キーワードリストが存在すべきでないユーザ入力キーワードとして，「9784845611690(注4)」「999 個入り(注5)」「正月ハワイ(注6)」を取り上げ，同様に C の上位 50 件の相対スコア値分布を図 3 の右側 (b) に示す．なお，図 3 では両軸共に対数を取っており，相対スコア値とは，修正キーワード候補の中でのスコア値最大値を 1 としたときの当該スコア値の割合である(注7)

図 3 の (a),(b) の両グラフを比較すれば，提示キーワードリストが存在すべきか否かを区別することは容易である．具体的には，提示キーワードリストが存在すべき場合には，ランキング上位候補の高々数個のスコア値が，ランキング下位候補のスコア値と比較して非常に高いスコア値となっている．

5.2　提示キーワード集合生成手法

5.1 をふまえ，提案手法における提示キーワードリスト生成の基本的アプローチ方法は，以下の 2 ステップとなる．

（1） 提示キーワード候補リスト C のスコア値を用いて，累乗関数への近似式を算出する．(図 4 の (1))

（2） 近似式から計算される c_i の基準スコア値 $BaseLine(c_i)$ と実際のスコア値の乖離を計算する．Top 1 の修正キーワード候補から順に提示キーワードリストの要素としていくが，乖離が小さくなった時点で提示キーワード集合へのアペンドを停止する．(図 4 の (2))

以下では，それぞれのステップに関して具体的な手法を記述する．

5.2.1　近似式の生成

提示キーワード候補リスト $C = \{c_1, c_2, \cdots, c_n\}$ において Top s から Top t $(1 \leqq s < t \leqq n)$ までのスコア値 $Score(c_i|u) \mid s \leqq i \leqq t$ の累乗関数への近似式を生成することで，任意のランキング i に対する基準スコア $BaseLine(i)$ を導出できるようにする．例えば，図 4 においては，Top 5 から Top 50 までのスコア値を最小二乗法によって累乗関数に近

(注3)：それぞれ「キャス・キッドソン」もしくは「キャスキッドソン」,「花畑牧場」,「ハリーポッター」への変換を意図している．
(注4)：商品型番のキーワード場合，単純に文字列が似ているからといって，商品が似ているとは限らないので，似た文字列である別の商品型番をユーザに提示すべきではない．
(注5)：同様に数字部分が似ている別のキーワードに置き換えるべきではない．
(注6)：ユーザの検索意図に合致する商品は存在しない日本語文字列である．
(注7)：入力キーワード u によって計算される提示キーワード候補リスト要素のスコア値の絶対値が大きく変化する．絶対スコア値のでは複数のユーザ入力キーワード u に対する提示キーワード候補リストのスコア分布が同一グラフ上に描けないため，図 3 では相対スコア値によりグラフをプロットしている．

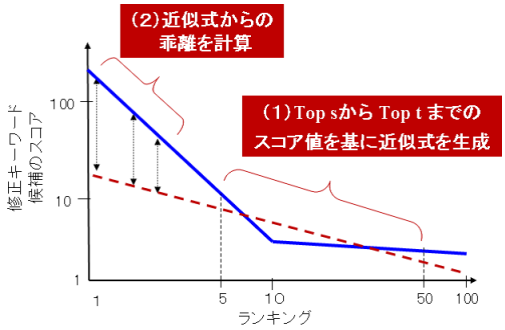

図 4　提示キーワード集合生成手法

似を行っている．

提示キーワードリストの選択条件を強くしたい場合は，s，t を小さい値にすることで解決できる．実際の (s,t) の値選択については，第 6 節にて言及を行う．

5.2.2　乖離に基づく提示キーワードリスト生成

提案手法では，提示キーワード候補リスト C の要素を順番に，実際のスコア値 $Score(c_i|u)$ と近似式から計算されるランキング i の基準スコア $BaseLine(i)$ の次式で示す乖離 $Diff(i)$ を計算していく．

$$Diff(i) = score(c_i|u) - BaseLine(i) \tag{7}$$

$Diff(i)$ が大きい値であれば，当該提示キーワード候補リスト要素 c_i を，提示キーワードリスト R にアペンドしていくが，一旦 $Diff(i)$ が小さい値であると判断された場合には，当該要素 c_i を R にアペンドせずに提示キーワードリストの生成を終了する．ここで，$Diff(i)$ の値の大小判定は，基準スコア $BaseLine(i)$ の定数倍 $(p \cdot BaseLine(i))$ とする．ここで，p は提示キーワードリスト生成にあたってのパラメータである．パラメータ p の選択については，第 6 節にて言及を行う．

6.　性能評価および精度向上実験

本節では，提案手法の性能を評価するとともに，提案手法の精度向上について言及を行う．

まず，修正キーワード提示の正解・不正解リストを取得するために，楽天ブックス [6] のクエリログを基にして生成した修正キーワード提示リストを人手により判定を行った．修正キーワード提示の正解・不正解リストの取得について，6.1 に示す．

6.1 をテストデータとし，第 4 節にて記述したスコア関数 $score(w_i|u)$ の調整について 6.2 に示す．さらに，第 5 節にて記述した候補キーワードリストの生成手法の調整について 6.3 に示す．

最後に，6.2，および 6.3 の結果を踏まえたうえで，楽天市場 [5] のクエリログを対象にした提案手法の精度に関して，6.4 にて示す．

6.1　正解・不正解リストの取得

修正キーワード提示の正解・不正解リストを取得するために，楽天ブックス [6] の 2009 年 6 月 1 日〜30 日の期間のクエリログを対象として，提案手法を適用し，修正キーワード提示リストの生成を行った．ここで，提案手法を適用させるにあたり，

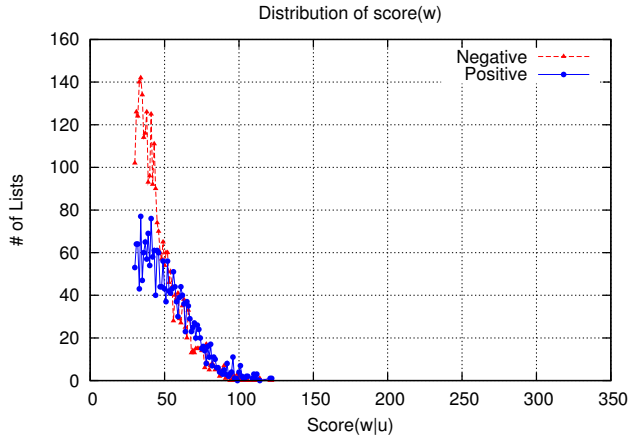

図 5　正解・不正解リスト要素のスコア分布

スコア関数を以下のように定めた．

$$score(w_i|u) = \frac{Pr(w_i)+2}{D(w_i,u)+0.05}A(w_i) \tag{8}$$

$$P(w_i) = log_{10}SC(w_i) \tag{9}$$

$$D(w_i,u) = 0.2(1-Jaro(w_i,u)) + 0.8(1-Jaro(w_i.yomi, u.yomi)) \tag{10}$$

$$A(w_i) = \begin{cases} 1(RN(w_i) \geqq 1) \\ 0(RN(w_i) = 0) \end{cases} \tag{11}$$

また，提示キーワード選択時においては，近似式導出のためのスコア区間を $s=11, t=50$ とし，一次関数への近似を実施した(注8)．さらに基準スコアからの乖離の閾値パラメータ $p=1.5$ と設定した．その結果 44,217 件の修正キーワード提示リストが生成された．

抽出された約 4 万件以上の修正キーワード提示リストのうち，スコア値 30 以上のリストから 7,281 件を選択し，人手でチェックを行った．その結果，正解リスト 2,875 件 (39.49%)，不正解リスト 4,406 件 (60.51%) を取得した．正解リスト・不正解リストの例を表 1 に示す．また，正解・不正解リスト要素のスコア分布を図 5 に示す．図 5 では，スコア値と当該スコア値を持つリスト数をプロットしており，凡例の Positive は正解リスト，Negative は不正解リストを示す．

この正解・不正解リストを基にして，6.2，6.3 にて提案手法の最適化を試みる．

6.2　スコア関数の調整

スコア関数の調整の目的は，正解リスト要素のスコア値を高くし，不正解リスト要素のスコア値を低くすることである．これにより，次のステップである候補キーワードリスト生成において，提示キーワード候補リストに含まれる不正解要素の提示を抑制することが期待できる．

第 4 節にて示したスコア関数の調整を図るために，スコア関数の 3 つの条件（距離条件，検索頻度条件，検索可能性条件）について，正解・不正解リスト要素の分布を調査した．距離条

(注8)：提案手法では，累乗関数への近似を実施するが，ここでは，正解・不正解リストを幅広く取得するために，累乗関数ではなく一次関数への近似を実施した．

表 1　正解・不正解リスト例

判定結果	ユーザ入力キーワード	提示キーワード	$Pr(w_i)$	$D(w_i,u)$	$A(w_i)$	$score(w_i,u)$
正解	久保田カヨコ	久保田カヨ子	4.32	0.022	1	121.015
	週間プロレス	週刊プロレス	3.283	0.022	1	101.157
	つれがうつになりまして	ツレがうつになりまして。	1.813	0.047	1	49.607
	横峰吉文	横峯吉文	3.356	0.033	1	84.563
	ザトレーシーメソッド	ザ・トレーシー・メソッド	2.749	0.038	1	70.067
不正解	トレーシーメソッド	ザ・トレーシー・メソッド	2.749	0.07	1	47.49
	久保田カヨコ	久保田カヨ	1.771	0.056	1	44.075
	トレイシーメソッド	トレーシー・メソッド	2.057	0.095	1	32.484
	つれがうつになりまして	ツレがうつになりまして	2.839	0.032	1	78.499
	ロッキンオンジャパン	ロッキングオンジャパン	1.892	0.061	1	42.929

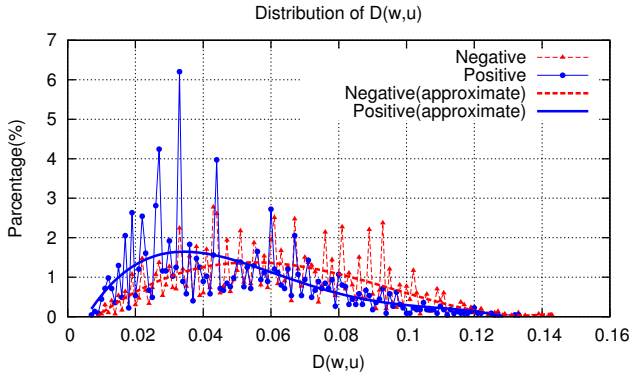

図 6　正解・不正解リストに含まれる提示キーワードの Distance 分布

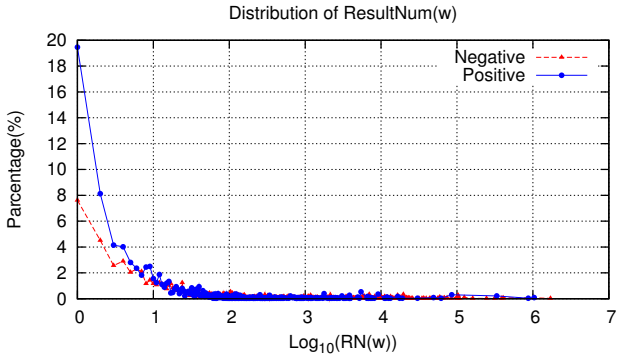

図 8　正解・不正解リストに含まれる提示キーワードの検索結果数分布

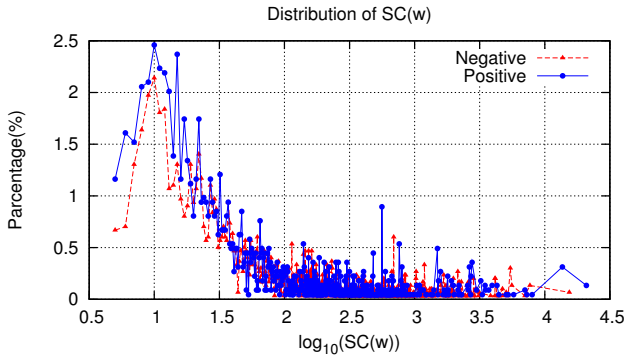

図 7　正解・不正解リストに含まれる提示キーワードの検索回数分布

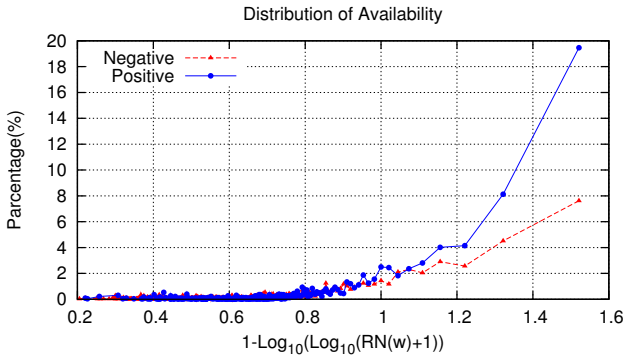

図 9　正解・不正解リストに含まれる提示キーワードの Availability 分布

件の変数にあたる $D(w_i,u)$ の分布を図 6 に，検索頻度条件の変数にあたる $SC(w_i)$ の分布を図 7 にそれぞれ示す．また，検索可能性条件の変数にあたる $A(w_i)$ を設計するために，提示キーワードによる検索結果数である $RN(w_i)$ の分布を図 8 に示す．図 6，7，8 は，値と当該値を持つ要素数のリスト全体に対する割合をプロットしており，凡例の Positive は正解リスト，Negative は不正解リストを表す．

6.2.1　距離条件項

図 6 に示すとおり，正解リスト要素は $D(w,u)$ の値が小さい範囲に多く存在する．逆に，不正解リスト要素の場合，$D(w,u) \simeq 0.06$ を頂点とした正規分布に近い分布となっている．したがって，$D(w,u)$ の値の大小が，スコア関数の値の大小に大きく影響させるようにすれば良い．具体的には，距離条件項のパラメータである β の値を，0.05 から 0.01 に低くすることで，$D(w,u)$ の $score(w|u)$ に対する影響度を大きくした．

6.2.2　検索頻度条件項

図 7 に示すとおり，$SC(w)$ の値には正解・不正解リストの際に大きな差はないと考えられる．したがって，検索頻度条件項は変更しない事とした．

6.2.3　検索可能性条件項

図 8 に示すとおり，検索結果数 $RN(w_i)$ が小さい値の範囲において，正解リストと不正解リストの割合に明確な差が存在する．これは，提示キーワードによって再検索した際に，ユーザの求める商品が明確に定まる事を意図して，修正キーワード提示リストを判定したことによるものだと考えられる．した

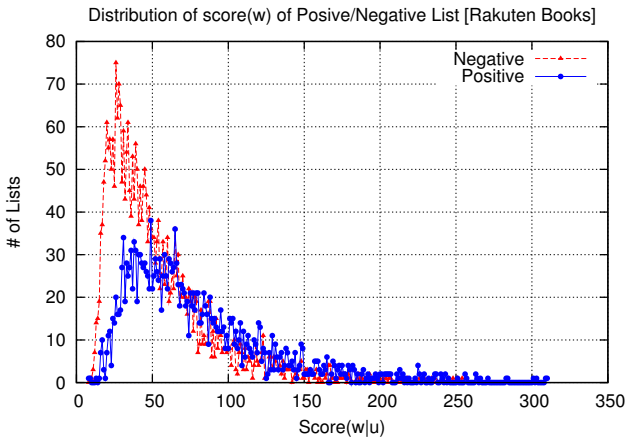

図 10　スコア調整後の正解・不正解リスト要素のスコア分布

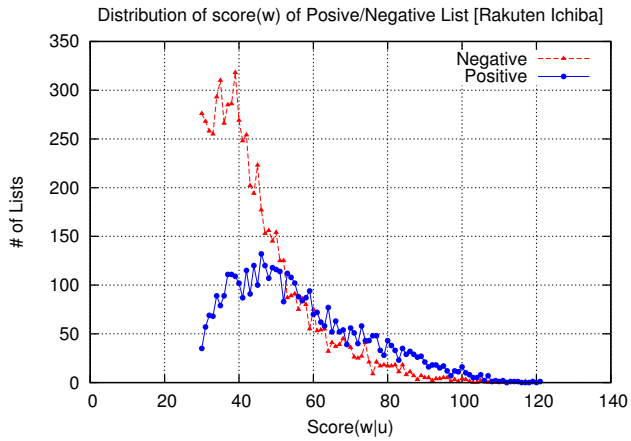

図 12　提案手法調整前のスコア分布（楽天市場）

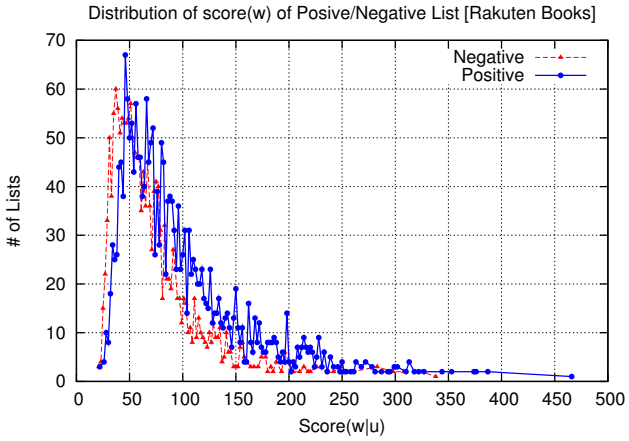

図 11　提案手法調整後のスコア分布

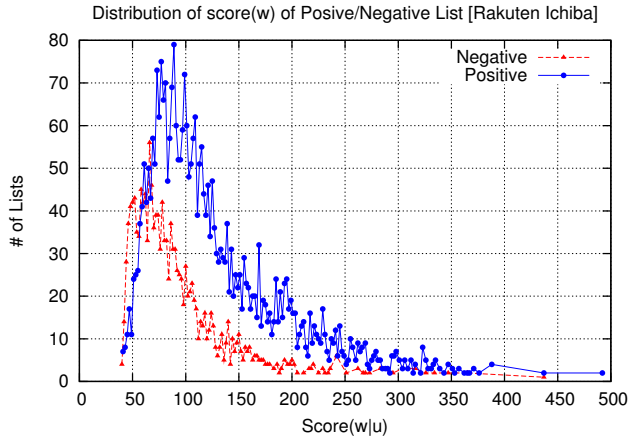

図 13　提案手法調整後のスコア分布（楽天市場）

がって，$A(w_i)$ の設計においては，$RN(w_i)$ が小さい場合に，$A(w_i)$ が大きくなる設計を行う方針をとった．

具体的には，式 (12) のように $A(w_i)$ を定め，式 (12) を基にした正解・不正解リストの $A(w)$ の分布は，図 9 のとおりとなる．

$$A(w_i) = 1 - \{log_{10}(log_{10}(RN(w_i) + 1))\} \quad (12)$$

6.2.4　調整後のスコア関数

6.2.1〜6.2.3 の議論を踏まえ，調整後のスコア関数は，以下のとおりとなる．

$$score(w_i|u) = \frac{Pr(w_i) + 2}{D(w_i, u) + 0.01} A(w_i) \quad (13)$$

$$Probability(w_i) = log_{10} SC(w_i) \quad (14)$$

$$Distance(w_i, u) = 0.2(1 - Jaro(w_i, u)) + 0.8(1 - Jaro(w_i.yomi, u.yomi)) \quad (15)$$

$$A(w_i) = 1 - \{log_{10}(log_{10}(RN(w_i) + 1))\} \quad (16)$$

調整後のスコア関数に基づいて，正解リスト要素・不正解リスト要素全てをスコア計算しなおすと，図 10 となる．図 5 と図 10 を比較すると，スコア関数の調整により，正解リストの要素のスコア値が相対的に大きくなったことが確認できる．

6.3　候補キーワードリスト生成手法の調整

6.2 にて示した修正キーワード候補のスコア関数の調整を反映させた上で，提示キーワードリスト生成手法の最適化を図る．本実験では，近似式を計算する為の対象ランキング区間 (s, t)，乖離判定のための閾値を決定する上でのパラメータ p を変化させた上で，6.1 にて示した同様のクエリログを対象にし，提案手法を適用することで修正キーワード候補リストの生成を行った．その上で，人手で判定を行った正解・不正解リストと照らし合わせ，精度，再現率，F 値を計算することで，適切な対象ランキング区間，パラメータ p の設定を導き出す．ここで，ランキング区間は $(s, t) = (1, 5), (1, 10), (1, 20), (5, 10), (5, 20), (10, 20)$ の 9 つの区間を候補とし，パラメータ p は，1.1，1.2，1.3 の 3 つの値を候補とした．

抽出実験結果を表 2 に示す．表 2 において，TP，FP，FN はそれぞれ以下を意味する．

TP　抽出したリストのうち，正解リストに含まれる数
FP　抽出したリストのうち，不正解リストに含まれる数
FN　正解リストの中で，抽出したリストに含まれなかった数

表 2 より，F 値が最も高い値を示したのは，ランキング区間 $(s, t) = (1, 10)$，パラメータ $p = 1.1$ の時である．正解・不正解リスト取得時の提案手法の精度は 39.49%であったのに対し，

表 2　提示キーワードリスト生成手法の精度・再現率・F 値評価

p	s	t	生成リスト数	TP	FP	FN	精度 (%)	再現率 (%)	F 値
1.1	1	5	19,931	1,783	1,110	1,092	61.63%	62.02%	0.618
	1	10	33,729	2,265	1,757	610	56.32%	78.78%	0.657
	1	20	48,247	2,441	2,303	434	51.45%	84.90%	0.641
	5	10	110,756	2,560	2,670	315	48.95%	89.04%	0.632
	5	20	114,627	2,614	2,846	261	47.88%	90.92%	0.627
	10	20	152,767	2,637	2,976	238	46.98%	91.72%	0.621
1.2	1	5	11,791	1,392	693	1,483	66.76%	48.42%	0.561
	1	10	22,409	1,990	1,364	885	59.33%	69.22%	0.639
	1	20	32,531	2,334	1,970	541	54.23%	81.18%	0.650
	5	10	66,972	2,524	2,530	351	49.94%	87.79%	0.637
	5	20	70,385	2,582	2,704	293	48.85%	89.81%	0.633
	10	20	85,430	2,609	2,856	266	47.74%	90.75%	0.626
1.3	1	5	7,175	1,026	394	1,849	72.25%	35.69%	0.478
	1	10	16,834	1,778	1,056	1,097	62.74%	61.84%	0.623
	1	20	25,681	2,185	1,667	690	56.72%	76.00%	0.650
	5	10	50,304	2,478	2,415	397	50.64%	86.19%	0.638
	5	20	53,352	2,537	2,585	338	49.53%	88.24%	0.634
	10	20	62,275	2,568	2,747	307	48.32%	89.32%	0.627

最適化後の精度は 56.32%，再現率は 78.78%となった．この条件にて，抽出した修正キーワード提示リストのうち，正解・不正解リストに含まれていた要素のスコア分布を図 11 に示す．図 5 と比較することで，精度の向上を確認することができる．

6.4　楽天市場のクエリログへの適用

楽天市場 [5] の 2009 年 6 月 1 日〜7 月 9 日の検索クエリログをベースとして，6.1 と同じ条件にて抽出したスコア値 30 以上の修正キーワード提示リストから 9,268 件を選択し，人手でチェックを行った．その結果，正解 4,862 件 (40,05%) および不正解 4,406 件 (59.95%) のリストを取得した．この両者のリストのスコア分布について，図 12 に示す．

6.2 および 6.3 で示した提案手法の精度向上結果をそのまま適用し，再度同様のクエリログに対し修正キーワード提示リストを抽出したところ，3,333 件の要素が正解リストに，1,522 件が不正解リストに含まれ，精度 68.65%，再現率 74.43%，F 値 0.714 という結果となった．調整後の提案手法で抽出したリストのうち，正解・不正解リストに含まれていた要素のスコア分布を図 13 に示す．

7. おわりに

本稿では，キーワード型検索エンジンにおける修正キーワード候補提示手法を提案した．提案手法は，(1) 入力キーワードに対する修正キーワード候補のスコアリング，および (2) スコアリングされた修正キーワード候補集合から，適切な数の提示キーワードリストの生成の 2 つのステップにて構成されている．

修正キーワード候補のスコアリングにおいては，修正キーワード候補各々に対し，入力キーワードと修正キーワード候補の距離条件，修正キーワード候補の検索頻度条件，検索可能性条件の 3 つの条件を考慮してスコア値を計算する．また，スコア値降順による修正キーワード候補リストに対し，ランキング上位候補のスコア値突出値を検出することで，提示キーワードリストを生成する．

提案手法を，楽天ブックス，および楽天市場のクエリログを対象として実験を行ったところ，楽天ブックスでは 56.32%，楽天市場では 68.65%の精度で，修正キーワード候補を提示することに成功した．

今後の課題は，提案手法の高速化を実現することで，大規模な検索エンジンシステムにおいても，適用可能にすることである．

文　献

[1] H. Dalianis, “Evaluating a spelling support in a search engine”, Proc. of NLDB2002, pp.183–190, 2002.

[2] Google, http://www.google.co.jp

[3] Yahoo! JAPAN, http://www.yahoo.co.jp

[4] Bing, http://www.bing.com

[5] 【楽天市場】Shopping is Entertainment!：インターネット最大級の通信販売，通販オンラインショッピングコミュニティ，http://www.rakuten.co.jp

[6] 楽天ブックス｜本・DVD・CD・ゲームの通販 オンライン書店，http://books.rakuten.co.jp

[7] K. Kukich, “Techniques for automatically correcting words in text”, ACM Computing Surveys Vol.24 No.4, pp.377–440, 1992.

[8] F. J. Damerau, “A technique for computer detection and correction of spelling errors”, Communications of the ACM, Vol.7, No.3, pp.171–176, 1964.

[9] V. I. Levenshtein, “Binary codes capable of correcting deletions, insertions and reversals”, Soviet Physics-Doklady, Vol.10, No.8, pp.707-710, 1966.

[10] R. L. Kashyap and J. Oommen, “Spelling correction using probabilistic methods”, Pattern Recognition Letters, 1985.

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

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

[13] M. A., Jaro, “Advances in record-linkage methodology as applied to matching the 1985 census of Tampa, Florida”, Journal of the American Statistical Association Vol.84, pp.414–420, 1989.