情報処理学会論文誌　データベース　Vol.6　No.1　16–28 (Jan. 2013)

ウェブ検索クエリログと クリックスルーログを用いた同義語獲得

内海 慶[1,a)]　小町 守[2,b)]

受付日 2012年6月20日, 採録日 2012年8月1日

概要：近年のウェブ検索エンジンの多くはクエリ拡張機能やクエリ書き換えを備えている．これらの機能の実現にはシソーラスや同義語辞書を用いるが，人手での辞書作成はコストがかかる．そのため，ウェブ検索ログやクリックスルーログを用いた同義語獲得の研究が行われている．これまでに提案された手法では，生成モデルである Noisy Channel Model によって同義語獲得をモデル化しており，柔軟な素性設計が行えなかったため，クエリと同義語候補の表層の編集距離を素性として追加する等が難しかった．我々は，この問題に対処すべく，同義語獲得に識別モデルを用いた手法を提案する．クエリ書き換えのための同義語辞書では，1 つのクエリに対してより適切と考えられる 1 つの同義語を登録する．そのため同義語獲得手法には，同義語候補が複数ある場合には最適な候補を 1 位に提示することが求められる．そこで提案手法では，クエリと同義語候補の表層に基づく素性を利用した ListNet を用いて 1 位正解率を直接最大化する．また，従来の識別モデルでは，有効な組合せ素性の追加等，素性エンジニアリングを行う必要があったが，我々は ListNet に隠れ層を導入することで，素性エンジニアリングなしに有効な組合せ素性の生成と重み付けを可能とした．これにより，Noisy Channel Model を用いた従来の手法に比べ，より高い精度で同義語を獲得することができた．

キーワード：同義語獲得，ランキング学習，クエリ拡張，クエリ書き換え

Synonym Extraction Using Web Search Query and Click-through Logs

Kei Uchiumi[1,a)]　Mamoru Komachi[2,b)]

Received: June 20, 2012, Accepted: August 1, 2012

Abstract: Recent web search engines often employ query expansion and query reformulation techniques. These techniques use thesauri and synonym dictionaries, but manually making dictionary requires time and costs. Thus, automatic acquisition of synonymous expressions using web-search logs and click-through logs has been studied. One of the previous work formulates the synonym extraction problem as a generative process using the noisy channel model, but since generative models do not allow flexible feature design, it is difficult to use as features edit distance between the surface of a query and its synonym. To deal with this problem, we employed discriminative approaches for synonym extraction. When creating a synonym dictionary for query reformulation, only one synonym which better leads to appropriate search results is registered for each query. Therefore, it is required that the synonym acquisition method for query reformulation must pick an optimal entry if there are several synonym candidates. Hence we propose to maximize the 1-best accuracy using ListNet with features based on the surface of a query and its synonym to achieve the goal. Moreover, though most traditional discriminative methods require feature engineering to find efficient combinations of features, we automate this process by introducing hidden layers to the ranking function. Our proposed method outperformed previous method based on the noisy channel model in the task of synonym extraction.

Keywords: synonym acquisition, learning to rank, query expansion, query reformulation

1. はじめに

近年，ほとんどのウェブ検索エンジンはクエリ拡張機能を備えている．クエリ拡張では，シソーラスや同義語辞書を用いて入力クエリをより適切なクエリへ置き換えたり，入力クエリへ同義語の追加を行ったりして，クエリの表記を直接含まなくても，表記揺れや異表記を含んでいるウェブページを検索結果に加えることで，検索結果の再現率を向上させることができる．

初期のクエリ拡張では，同義語辞書は人手によって作成されていた．しかしながら，膨大な量の検索クエリをカバーするような大規模な辞書の構築にはコストがかかり，かつ高品質なリソースを作るためには作成者にドメイン知識が要求される．加えて，ウェブ検索では新しいクエリが継続的に現れる．そのため，辞書はつねに更新し続ける必要があり，すべてを人手で行うのは現実的ではない．

日本語の検索クエリの研究では，セッションログ中にある人手によるクエリ改善が行われたと考えられるクエリペアのうち，スペル訂正と同義語への置き換えは 13% を占めると Jones ら [14] は報告しており[*1]，ウェブ検索のクエリ書き換えへの利用として「もしかして」や「次の検索結果を表示しています」といった機能を考えた場合，同義語獲得は重要な課題である．

こうした問題に対処するために，検索クエリログを用いた辞書の自動構築が研究されている．検索クエリはユーザの意図を推定するには曖昧かつノイジーではあるものの，低コストに新規クエリを含む辞書を構築するには有効な資源である．検索クエリログはクエリ訂正への利用 [12] や意味カテゴリ獲得 [26] 等，近年の情報検索や自然言語処理の領域において広く使われるようになっている．

また最近では，検索クリックスルーログも語彙獲得の分野で注目されている．ウェブクリックスルーは，検索結果にある URL をユーザがクリックし，ウェブページを参照したことを表す．ウェブ検索において，ユーザがタイトルや ‘URL’，ページの要約をチェックしたうえでクリックするため，どのページをクリックしたかにはユーザの意図が直接反映されている．そのため，同じ ‘URL’ に到達する 2 つの異なるクエリは，同じ意図で検索された可能性が高く，関連性があると考えられる．クリックスルーログは，自然言語処理分野において，意味カテゴリ獲得 [17]，固有表現抽出へと適用 [13] されている．また，Komachi ら [17]，Jain ら [13] では，コーパスに検索クエリログを用いるよりも，クリックスルーログを用いた方が再現率，適合率が優れていることを報告している．

検索クエリログとクリックスルーログの両方を用いた同義語獲得の研究には，Uchiumi ら [30] の研究がある．彼らの手法では，Noisy Channel Model によって同義語獲得をモデル化している．クエリ訂正を目的とした同義語獲得では，獲得された語彙が入力されたクエリの同義語であるだけでなく，クエリとして適切であることが望まれる．Uchiumi らは，Channel Model にクリックスルーグラフ上のラベル伝搬を，Source Model に検索クエリログから構築した言語モデルを用いることでこの問題に対処している．しかし，生成モデルである Noisy Channel Model はモデルの導出に素性の独立性の仮定が入るため，任意の素性を柔軟に利用することができない．また，「もしかして」や「次の検索結果を表示しています」では，ユーザの入力したクエリに対して最も適切な候補の提示，置き換えを行う．そのため，実際に用いられる同義語辞書では，クエリに対して適切と判断された同義語が 1 件のみ登録されている．したがって，同義語獲得手法には，同義語候補が複数ある場合にはその中から最も適切な候補を 1 位に提示することが要求されるが，Noisy Channel Model では 1 位正解率を直接最大化することはできない．

これに対し，表記ゆれの獲得に識別モデルを用いた Suzuki ら [29] の研究がある．彼女らの研究では，識別モデルを用いてクエリの表記ゆれをモデル化することで，クエリと候補の間の編集距離や文字種等の表層に関わる情報やその組合せを素性として利用している．文字種の組合せ素性を入れることで，それらを使わない場合に比べエラー率で 0.4% ほどの改善が得られている．しかし，適切な組合せ素性の作成にはタスクや言語に対する知識を必要とする．

本稿では，上述の問題に対処するべく，同義語獲得に中間層を導入したランキング学習を用いた手法を提案する．提案手法では，Uchiumi らと同様，検索クエリログと検索クリックスルーログの両方を用いる．ただし，生成モデルを用いた手法と異なり，識別モデルによって入力クエリが与えられた際の同義語候補の条件付き確率を直接モデル化することにより，自由な素性の設計が行えるようになっている点が異なる．たとえば，ログデータ以外にもクエリと同義語候補の間の編集距離や文字種等の表層に関わる情報も素性として利用することができる．加えて，中間層によって素性の組合せと重み付けを行うことで，人手での素性の組合せを不要とする．

本研究の主要な貢献は以下の 2 点である．

(1) 先行研究では生成モデルである Noisy Channel Model を用いた同義語獲得手法が提案されていたが，本研究は識別モデルであるランキング学習を導入し，1 位正解率を最大化するように学習した．

1 ヤフー株式会社
Yahoo Japan Corporation, Minato, Tokyo 107–6211, Japan
2 奈良先端科学技術大学院大学
Nara Institute of Science and Technology, Ikoma, Nara 630–0192, Japan
a) kuchiumi@yahoo-corp.jp
b) komachi@is.naist.jp

*1 彼女らはスペル訂正と同義語への置き換えは両方意味的には同じ置換であるとして，区別していない．

(2) 先行研究では最適な素性の組合せの発見に人手による素性エンジニアリングが必要であったが，入力素性ベクトルと出力ラベルの間に中間層を導入し，素性の組合せと非線形な重み付けを自動化した

以降，2 章ではクエリ拡張に関連する先行研究について説明を行い，3 章で同義語獲得問題の定式化を行う．4 章では，ランキング素性について説明を行う．5 章で我々の手法の評価を行い，その効果を示す．6 章では，総論を行い，今後の課題を示す．

2. 関連研究

ウェブ検索におけるクエリ拡張では，新語やウェブのスラングを扱う必要がある．そのため，正しい検索クエリのリストを継続的に維持し続けるには多大なコストがかかる．加えて，日本語におけるクエリ拡張は，単語分割や接尾辞処理，略語・頭字語の展開，単語の追加・削除，文脈を考慮した単語の修正，スペル訂正等複数のタスクを含む．これまでの研究では，各タスクは個別に焦点が当てられてきた [1], [2], [7], [19], [24], [25]．一方，近年はそれぞれのタスクを同時に処理する手法が提案されている．以下にそれぞれについて述べる．

Cucerzan ら [27] は，検索クエリにおけるスペル訂正の問題を明確化し，Noisy Channel Model で上述の各種問題に対処した．Gao ら [10] および Sun ら [28] は，Cucerzan らの手法で獲得した検索クエリのスペル訂正候補に対し，ニューラルネットによるリランキングを行うことでスペル訂正を行った．彼らのリランキング手法ではランキング素性の翻訳モデルの学習に，検索クリックスルーログを利用している．ここであげた手法のそれぞれは訂正候補の獲得に編集距離を用いているため，表記が異なる同義語への置き換えや略語・頭字語の展開等は扱えない．

Wei ら [31] はクエリ間で共通してクリックされる URL の分布の Jensen-Shannon ダイバージェンスに基づく同義語の抽出に取り組んだ．彼らのアプローチはクリックスルーグラフから同義語候補を獲得するという点で，我々の同義語候補の獲得手続きと類似している．しかし，彼らの手法は全クエリ間の co-click 分布から求めた JS ダイバージェンスに基づくクラスタリング，およびクエリ間の JS ダイバージェンスに基づいてクエリの共起語間の類似度を求めており，クエリ数に対してスケールしない．また，我々の手法では獲得した候補集合に対して識別モデルによるランキングを導入している点が異なる．

Guo ら [11] はクエリ拡張に識別モデルを用いた統一的手法を提案している．この手法では，CRFs [18] の素性関数にクエリ拡張の各手続きを表す ‘operation’ を入れ，3 つ組 (‘feature’, ‘label’, ‘operation’) に拡張したモデルを使用している．‘operation’ には，‘deletion’, ‘insertion’, ‘substitution’, ‘transposition’ 等がある．この手法では教師データを必要とするが，人手で入力クエリに対する訂正とその手続きのラベルを与えるのはコストが大きい．また，素性には語彙化したものも利用しているため，教師データに含まれないクエリが与えられた場合には予測に使用する素性が不足する．

検索ログを用いた別のタスクに，クエリサジェスチョンがある [5], [20]．クエリサジェスチョンは，ユーザが入力したクエリから意味的に異なるクエリが推薦されうる点で我々のタスクとは異なる．

また，日本語の略語獲得に取り組んでいる研究もある．Murayama ら [22] は日本語略語の生成過程を Noisy Channel Model で定式化した．しかし，彼らの手法では略語の展開は扱わない．Okazaki ら [23] は日本語略語獲得を 2 値分類問題として扱った．彼らはニュース記事からヒューリスティックを用いて単語のペアを獲得し，これらを略語と略語以外に分類している．しかし，彼らのヒューリスティックはウェブ検索クエリには適用できない．

3. ランキング学習を用いたウェブ検索ログからの同義語獲得

本章では，我々の提案する同義語獲得手法について説明する．**図 1** に，提案手法のフレームワークを示す．ウェブ検索ログやクリックスルーログを用いた同義語候補の獲得では，Uchiumi ら [30] および，Sun ら [28] による先行研究がある．どちらの手法も，ユーザ行動によって結び付けられたクエリのペアを取り出している．ウェブ検索に関するユーザ行動としては，クリックスルーや検索セッションが考えられるが，後者のセッションについては，セッションの基準となる時間を非常に狭く設定しても取り出せるデータがノイジーであったことが Sun らによって報告されている．これについては我々も同様にセッションログの集計を行い，クエリの変化を見ることで確認できた．したがって，本研究ではクエリのペアを取り出すユーザ行動として，ク

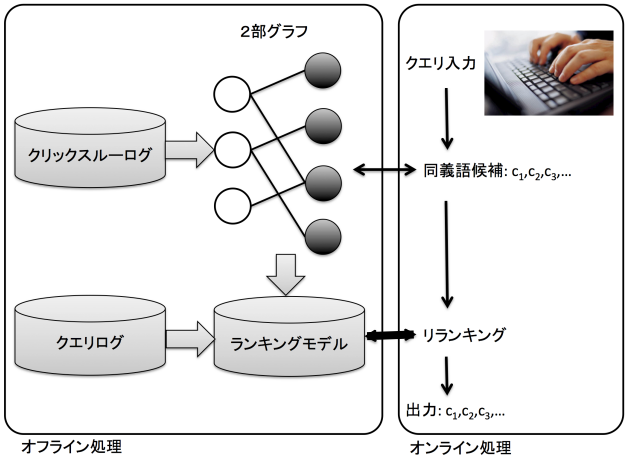

図 1　ウェブ検索ログを用いた同義語獲得のフレームワーク
Fig. 1　Synonym extraction framework using web search logs.

リックスルーを対象とした．

クリックスルーログの利用に関して，Sun らは，ブラウザの検索窓を通じて入力されたクエリに対し，各種ウェブ検索エンジンが訂正を返したクエリのペアのうち，ユーザが訂正クエリをクリックしたものについて収集し翻訳モデルの訓練データとしている．Sun らの手法はデータとしては非常に正確なものとなるが，それをそのまま同義語候補の獲得に使用した場合，既存のウェブ検索エンジンの訂正できるクエリ以外は同義語候補を獲得することができないので，表記の異なる同義語も対象とした我々のタスクには適さない．

そのため我々の手法では，同義語候補の獲得に Uchiumi ら [30] と同様にクリックでつながる 2 つのクエリを同義語候補とし，検索クリックスルーグラフ上でのラベル伝搬を用いた．まず，検索クリックスルーログを用いて，入力クエリとクリックされた URL の 2 部グラフを構築する．この 2 部グラフ上でのクエリ間の関連度を計算し，同義語候補のセット C を獲得する．次に，獲得した同義語のリランキングに，我々はランキング学習を用いる．ランキングには ListNet [6] を採用した．ListNet は本質的に 1 位正解率を最大化するように最適化する手法であり，ウェブ検索エンジンの「もしかして」や「次の検索結果を表示しています」のように，1 位に正解を当てなければならないタスクには適している．本タスクでは，提案手法の利用としてウェブ検索でのクエリ拡張を想定しており，1 位正解率を最大化することが求められる．しかし，従来の Noisy Channel Model を使った生成モデルでは，確率値が最大である候補が必ずしも 1 位正解率を最大化する保証はない．したがって我々は ListNet を用いて 1 位正解率を直接最大化した．

以降で，我々の使用するランキング学習について述べる．

3.1　Top 1 ListNet

本節では，我々の利用する ListNet について説明する．ListNet では，クエリ $q^{(i)}$ が与えられた際に得られた $n^{(i)}$ 件の同義語候補から，上位 k 件の候補の順列が得られる確率を式 (1) のように定義する．

$$P_{z^{(i)}(f_w)} = \prod_{t=1}^{k} \frac{exp(f_w(x_{j_t}^{(i)}))}{\sum_{l=t}^{n^{(i)}} exp(f_w(x_{j_l}^{(i)}))} \tag{1}$$

f_w は同義語候補の素性ベクトル x_{j_t} に対するランキング関数で，$z^{(i)}(f_w)$ は同義語候補の順列に対する f_w のスコアリストを表す．k が大きい場合にはパラメータ更新の計算が困難となるため，通常は $k = 1$ の場合のみを扱う．$k = 1$ のときの ListNet を，ここでは Top 1 ListNet と呼ぶ．

モデルの学習では，学習データに対するクロスエントロピー式 (2) を最小化することで，ランキング関数のパラメータ w を推定する．

$$L(y^{(i)}, z^{(i)}(f_w)) = -\sum_{j=1}^{n^{(i)}} P_{y^{(i)}}(x_j^{(i)}) log(P_{z^{(i)}(f_w)}(x_j^{(i)})) \tag{2}$$

$y^{(i)}$ はクエリ $q^{(i)}$ に対する同義語候補 j の正解確率を表す．

パラメータの更新式を式 (3) に示す．

$$\triangle w = -\sum_{j=1}^{n^{(i)}} P_{y^{(i)}}(x_j^{(i)}) \frac{\partial f_w(x_j^{(i)})}{\partial w} + \frac{1}{\sum_{j=1}^{n^{(i)}} exp(f_w(x_j^{(i)}))} \sum_{j=1}^{n^{(i)}} exp(f_w(x_j^{(i)})) \frac{\partial f_w(x_j^{(i)})}{\partial w} \tag{3}$$

Cao ら [6] では，ランキング関数 f_w に単純な線形ニューラルネットモデルを採用している．

$$f_w(x_j^{(i)}) =< w, x_j^{(i)} >$$

Top 1 ListNet では，ランキング関数に線形ニューラルネットを採用していたため，出力 z と入力 x の間の線形な関係しか学習することはできない．このため，効果的でない組合せ素性を加えられた場合には，それがノイズとなって分類精度を低下させてしまう．

我々は予備実験として，後述するランキング素性のセットに対し，それらを組み合わせた素性を用いてランキングを行った．組合せ素性の生成には，素性テンプレートを利用した．素性テンプレートでは，素性をどのように組み合わせるかを，数式の形式で記述できる．

予備実験において，人間の直観に基づいたドメイン知識を用いた組合せ素性を追加することで精度は向上したものの，組合せ素性によっては精度の低下が見られた．有効な素性の組合せを見つけるには，人手による素性の組合せパターンの検討および，各素性を追加した際のランキング精度を評価する必要があるため，精度向上のために多くの組合せ素性を追加することは難しい．

3.2　NeuroListNet：中間層を用いた ListNet の拡張

組合せ素性の追加コストを下げるため，我々はランキング関数 f_w に非線形ニューラルネットを採用し，素性の組合せと，組合せ素性への適切な重み付けを行うよう ListNet を拡張した．ランキング関数に非線形ニューラルネットを用いることで，ノイズとなるような素性の組合せに対しては中間層で素性の重みを小さくすることができ，頑健に学習を行うことができる．これにより，素性設計も行いやすくなる．具体的には，出力と入力の間に中間層を導入し，中間層で組合せ素性を生成するとともにシグモイド関数による重みを与えることで，出力と入力の間の非線形な関係を学習できるように拡張している．これによって，ランキン

グ関数は式 (4) のように変更される．

$$f_w(x_j^{(i)}) =< w, \Phi(x_j^{(i)}) > \tag{4}$$

$\Phi(x_j^{(i)})$ は D 個の素性関数 ϕ_t からなるベクトルを表す．各素性関数 ϕ_t はシグモイド関数で表され，各素性関数ごとにパラメータ θ_t を持つ．

$$\phi_t(x_j^{(i)}) = \sigma(\theta_t, x_j^{(i)}) = \frac{1}{1 + exp^{-<\theta_t, x_j^{(i)}>}} \tag{5}$$

ランキング関数に変更を加えた Top 1 ListNet の損失関数を最小化するように，パラメータ w，θ_t を更新する．更新式を以下に示す．

$$\begin{aligned}\triangle w &= \frac{\partial L(y^{(i)}, z^{(i)}(f_w))}{\partial w} \\ &= -\sum_{j=1}^{n^{(i)}} P_{y^{(i)}(x_j^{(i)})} \frac{\partial f_w(x_j^{(i)})}{\partial w} \\ &\quad + \frac{1}{\sum_{j=1}^{n^{(i)}} exp(f_w(x_j^{(i)}))} \sum_{j=1}^{n^{(i)}} exp(f_w(x_j^{(i)})) \frac{\partial f_w(x_j^{(i)})}{\partial w}\end{aligned} \tag{6}$$

$$\begin{aligned}\triangle \theta_t &= \frac{\partial L(y^{(i)}, z^{(i)}(f_w))}{\partial \theta_t} \\ &= -\sum_{j=1}^{n^{(i)}} P_{y^{(i)}(x_j^{(i)})} \frac{\partial f_w(x_j^{(i)})}{\partial \theta_t} \\ &\quad + \frac{1}{\sum_{j=1}^{n^{(i)}} exp(f_w(x_j^{(i)}))} \sum_{j=1}^{n^{(i)}} exp(f_w(x_j^{(i)})) \frac{\partial f_w(x_j^{(i)})}{\partial \theta_t}\end{aligned} \tag{7}$$

$$\frac{\partial f_w(x_j^{(i)})}{\partial \theta_t} = w_t \sigma_t(x_j^{(i)})(1 - \sigma_t(x_j^{(i)}))x_j^{(i)} \tag{8}$$

$$\frac{\partial f_w(x_j^{(i)})}{\partial w} = \Phi(x_j^{(i)}) \tag{9}$$

素性関数 ϕ_t では，入力 $x_j^{(i)}$ に含まれる素性の組合せを生成し，それに非線形な重みを与えている．これによって，入力 $x_j^{(i)}$ を任意の D 次元に写像するとともに，入力と出力の間の非線形な学習を行う．

3.3 パラメータ推定

パラメータ推定の実装には，ListNet，NeuroListNet ともに L2 正則化を入れた FOBOS アルゴリズム [9] を用いた．FOBOS はパラメータ w に関する損失関数を，経験損失 $f(w)$ と正則化項 $r(w)$ の 2 つの関数に分け，2 ステップでパラメータ w の更新を行うアルゴリズムである．L2 正則化 $r(w) = \frac{\lambda}{2}||w||^2$ とした場合の式を以下に示す．

$$w_{k+\frac{1}{2}} = w_k - \eta_k \partial f,$$

$$w_{k+1} = \underset{w}{\operatorname{argmin}} \left\{ \frac{1}{2}||w - w_{k+\frac{1}{2}}||^2 + \frac{\lambda}{2}||w||^2 \right\}$$

2 ステップ目の左辺を 0 とおき，$\hat{\lambda} = \frac{\lambda}{2}$ とすると以下が得られる．

$$w_{k+1} = \frac{w_k - \eta_k \partial f}{1 + \hat{\lambda}} \tag{10}$$

η_k および λ はそれぞれ学習率および正則化の重みを表す．学習率のスケジューリングは，Collins ら [8] に従い，以下の式で行った．

$$\eta_k = \frac{\eta_0}{1 + k/N} \tag{11}$$

k はモデルの更新回数を，N は学習コーパスのサイズを表す．η_0，λ はあらかじめ与える定数である．θ_t についても w と同様にパラメータの更新を行う．

4. 検索クエリログとクリックスルーログを用いた同義語抽出素性

本章では，我々の同義語獲得で用いたランキング素性について説明する．**表 1** に，我々が使用したランキング素性の一覧を示す．

ランキング素性は，クリックスルーログからラベル伝搬で取り出した同義語候補を正解と不正解に分け，不正解とされたものに含まれる特徴をルールとして加える．$q == c$ は，クエリと同義語候補の一致を表し，同義語候補としてクエリ自身を提示していないかを判別する素性である．

Length(q)，Length(c) は，同義語として略語の展開，あ

表 1　検索クエリログとクリックスルーログを用いた同義語抽出素性
Table 1　Features for synonym extraction using web search and click-through logs.

Index	素性	概要
0	$q == c$	入力クエリと訂正候補の一致を表すブーリアン
1	Length(q)	入力クエリの文字数
2	Length(c)	同義語候補の文字数
3	Space(c)	同義語候補中の空白の正規化頻度
4	Alphabet(c)	同義語候補中のアルファベットの正規化頻度
5	Num(c)	同義語候補中の数字の正規化頻度
6	Hiragana(c)	同義語候補中のひらがなの正規化頻度
7	Katakana(c)	同義語候補中のカタカナの正規化頻度
8	Kanji(c)	同義語候補中の漢字の正規化頻度
9	Symbol(c)	同義語候補中の記号の正規化頻度
10	IsAcronym(q,c)	クエリと同義語候補が頭字語の関係[*2]
11	IsAcronym(c,q)	同義語候補とクエリが頭字語の関係
12	QMatches1stT(q,c)	同義語候補の第一トークンと入力クエリの一致
13	Tokens(c)	訂正候補に含まれるターム数
14	log $p(q\|c)$	ラベル伝搬のスコア
15	log $p_{lm}(c)$	クエリ言語モデルで求めた同義語候補の尤度
16	log $p_{tr}(c)$	TextRank で求めた同義語候補の popularity

[*2] 本タスクは日本語を対象としているため c，k，q や，r，l は同じ文字として扱った（e.g.，apple と appre は同じ文字列として扱う）．

るいは，同義語としてクエリの略語を提示する際に，入力と出力で文字列の長さが変化することを想定し，追加している．

Space(c) から Symbol(c) は文字種に関わる素性である．これらは属性語が同義語候補に含まれるかどうかを判断するために追加している．というのも，クリックスルーグラフから獲得した同義語候補には，属性語が付属するものが含まれるが，属性語[*3]を含むものは同義語への訂正候補としては不正解として扱われるからである．

通常属性語を含む場合，同義語候補には空白文字と属性語が含まれる．属性語は漢字の割合が多いが，日本語表記における固有名詞では空白文字を含まない．また，同義語が英語表記である場合，同義語候補に漢字とアルファベットと空白が含まれることになるが，通常こうした文字種を含むような固有名詞はまれである．

IsAcronym(q, c)，IsAcronym(c, q) は，同義語候補とクエリが頭文語の関係であることを想定した素性である．文字種の特徴は属性語が含まれるかどうか判別する大きな手がかりとなるため，文字種の違いを考慮し kakasi [15] を用いて入力クエリと同義語候補両方の読みを推定したうえで，読みについて第 1 引数の文字列のそれぞれが，第 2 引数に同じ順番で含まれているか否かを表す素性として追加した．また，QMatches1stT(q, c)，Tokens(c) も同義語候補に属性語を含んでいる場合を想定した素性である．属性語は，第 2 トークン以降に入力される傾向がある．そのため，複数トークンからなる同義語候補は属性語を含む可能性が高い．加えて，第 1 トークンとクエリが一致した場合には，その同義語候補は属性語を含む場合が多い．Tokens(c) は前者を，QMatches1stT(q, c) は後者を考慮するための素性である．

$\log\ p(q|c)$，$\log\ p_{lm}(c)$，$\log\ p_{tr}(c)$ はそれぞれ，クエリから同義語候補への訂正確率，同義語候補のクエリらしさ，同義語候補の固有名詞らしさを表す尺度である．これらについては，以降で詳しく説明する．

4.1　クリックスルー素性：$\log\ p(q|c)$

本研究ではクリックスルーログを用いた同義語候補の獲得に，Uchiumi ら [30] と同様ラプラシアンラベル伝搬を用いる．ラプラシアンラベル伝搬は，リスタート付きのランダムウォークと等価であり，クエリから同義語候補へ伝搬したラベルの確率は，クエリが与えられたときの同義語候補の条件付き確率と見なすことができる．したがって，この確率は同義語候補らしさを直接的に表すスコアであり，我々はこれをクリックスルー素性として用いる．ラプラシアンラベル伝搬の式を式 (12) に示す．

$$\mathcal{F}(t+1) = \alpha(-\mathcal{L})\mathcal{F}(t) + (1-\alpha)\mathcal{F}(0) \tag{12}$$

$\mathcal{L}$ はグラフラプラシアンで，グラフから計算されたノード間の距離行列を表す．我々はグラフラプラシアンとして正規化ラプラシアンを用いる．グラフラプラシアンの式を式 (13) に示す．

$$\mathcal{L} = I - D(A)^{-1/2} A D(A)^{-1/2} \tag{13}$$

A は隣接行列で，$A = W^{\mathrm{T}}W$ で表される．W はインスタンス・パターン行列で，ノード間の接続関係を表す．$D(A)$ は次数対角行列で，$D_{ii} = \sum_j A_{ij}$ となるような行列である．グラフラプラシアンは，行方向，列方向で総和が 0 となっており，自己ループを入れることで収束するよう調整されている．$\mathcal{F}(0)$ はシードのラベル，$\mathcal{F}(t)$ はグラフ上で伝搬させた各ノードのラベルを表す．α はシードベクトルとグラフのどちらをどの程度重視するかの調整パラメータであり，ラプラシアンラベル伝搬をリスタート付きのランダムウォークと考えた場合では $1-\alpha$ はリスタート確率と見なすことができる．本研究では，クエリをインスタンス，URL をパターンとして使用する．Uchiumi ら [30] に従い，$\alpha = 0.0001$ とし，W の要素には NPMI（Normalized Pointwise Mutual Information）[3] を使用した 1 ステップ近似を用いる．

4.2　クエリ言語モデル素性：$\log\ p_{lm}(c)$

獲得した同義語候補が，クエリとして適切かどうかを表す素性として，web 検索クエリログから構築したクエリ言語モデルを使用する．言語モデルには，式 (14) に示す文字 n-gram 言語モデルを採用した．

$$\begin{aligned} p(c) &= \prod_{i=0}^{N-1} p(x_i | x_{i-N+1}, \ldots, x_{i-1}) \\ &= \prod_{i=0}^{N-1} \frac{freq(x_{i-N+1}, \ldots, x_i)}{freq(x_{i-N+1}, \ldots, x_{i-1})} \end{aligned} \tag{14}$$

web 検索では，つねに新語が生成されているため，それらを辞書的に網羅することは困難であり，単語に基づくモデルでは分割誤りが発生する．したがって，本研究では単語 n-gram ではなく，文字 n-gram を用いて同義語候補のクエリらしさを求める．

4.3　TextRank 素性：$\log\ p_{tr}(c)$

Uchiumi ら [30] の報告では，クリックスルーグラフを用いた略語の展開の誤りとして，正解表記の部分文字列や属性語を含む場合を示し，これらへの対策として取り出された訂正候補のポピュラリティの使用をあげている．そこで，我々は検索クエリログに含まれるクエリのポピュラリティを計算し，素性の 1 つとして使用することにした．

TextRank [21] は文書や単語等をグラフのノードとし，

[*3] e.g.，クエリが商品の場合には，‘価格’ や ‘評判’ 等が属性語として用いられる．

ノード間に何らかの関連度（類似度や単語の共起等）で重み付けたエッジを張ることで，グラフベースの手法による重要度計算を行う手法である．Mihalcea ら [21] の研究では，HITS [16] や PageRank [4] を重要度計算の方法としてあげている．本研究では，ノードとして検索クエリログに含まれるクエリを用いた．また，重要度計算には PageRank を用いた．クエリ間の関連度は式 (15) で定義した．

$$H_{i,j} = \frac{1}{N_{q_j}}(\Sigma_{w \in q_i \cap q_j} \frac{1}{DF(w)}) \quad (15)$$

w は単語，N_{q_j} はクエリ中の単語数を表し，$DF(w)$ は検索クエリログに含まれるユニークなクエリ集合における Document Frequency を表す．つまり，少ない単語数かつ，希少性の高い単語で構成されているクエリほどスコアが高くなるように設計されている[*4]．

5. ウェブ検索ログからの同義語抽出の評価実験

5.1 使用データ

5.1.1 クリックスルーグラフの構築

評価には，2010 年 1 月 1 日から 2011 年 6 月 30 日までのウェブ検索クリックスルーログを用いた．ウェブ検索クリックスルーログの集計は，ブラウザ cookie ごとに 1 日単位で集計し，同じブラウザ cookie からの（クエリ，URL）のペアが複数回存在する場合には 1 回としてカウントした．1 日ごとの集計結果を出した後，上述の期間について総和をとり，評価データとした．ただし実際にすべてのデータを使用したところ，計算機にグラフラプラシアンを保持できなくなったため，複数トークンで構築されるクエリを含むものについては，アルファベットで構成されるもの以外は除外した[*5]．また，クエリと URL のペアの頻度 5 以下のものについても除外した．最終的に，71,750,207 件の（クエリ，URL）ペアが得られた．

インスタンス・パターン行列 W の要素は $NPMI$ を計算した後，値が 0.1 以下となるものについては 0 とした．

5.1.2 クエリ言語モデルの構築

クエリ言語モデルの構築には，クリックスルーログ同様に 2010 年 1 月 1 日から 2011 年 6 月 30 日までのウェブ検索クエリログを用いた．こちらについても同様に，ブラウザ cookie ごとに 1 日単位で集計を行い，同日の同じブラウザ cookie から複数回クエリが入力された場合には 1 回とカウントした．1 日ごとの集計結果を出した後，全期間で総和をとり，頻度 5 以下のクエリについては除外した．最終的に，92,566,110 件のクエリを用意した．

今回の実験では，$N = 5$ として文字 n-gram 言語モデルを構築し，素性として用いた．

5.2 TextRank の計算

TextRank の構築には，言語モデルの構築に用いた，頻度 5 以下を除外した検索クエリログから頻度を排除したユニークなクエリ集合を取り出し，それに含まれる各クエリをノードとして PageRank の計算を行った．

5.2.1 学習データ

学習データの作成には，実際にサービスで利用されている同義語辞書を用いた．辞書には全部で 5,871 件のクエリと，人手で付けたその同義語のペアが登録されている[*6]．登録されている同義語には，以下の種類がある．

(1) 略語の展開
- かな漢字の略語とその展開
  （e.g.,「俺つば」から「俺たちに翼はない」への展開）
- アルファベットの略語とその展開
  （e.g.,「bsb」から「backstreet boys」への展開）
- アルファベットの略語とその日本語での展開
  （e.g.,「ptsd」から「心的外傷後ストレス障害」への展開）

(2) 略語化
- 正式表記とその略称
  （e.g.,「駅前探検倶楽部」を「駅探」へ省略）

(3) 異表記
- アルファベット表記からカタカナ表記への変換
  （e.g.,「karen walker」から「カレンウォーカー」へ変換）
- かな漢字表記の変更
  （e.g.,「護摩山スカイタワー」から「ごまさんスカイタワー」へ変更）
- カタカナからアルファベット表記への変更
  （e.g.,「エクストレイル」から「x-trail」へ変更）

(4) 別名

同じ物を指す別名への変更
（e.g.,「岐阜県世界淡水魚園水族館」から「アクア・トトぎふ」への変更）

この辞書から，クエリとその同義語を取り出し，クエリをシードとしてクリックスルーグラフから同義語候補を関連度順に 20 件取り出した．20 件の中に，同義語が含まれていた場合には，その同義語には正解のラベルを与えた．また，辞書に含まれていない正解が含まれている場合も考慮し，Wikipedia のリダイレクトに含まれるクエリと同義

[*4] エッジに重みを付けずに PageRank を計算した場合では，属性語等の様々なタームと共起するようなクエリが上位にきてしまい，誤りや属性語を含まない固有表現のみで構成されるクエリを効率的に取り出すことができなかった．

[*5] 開発機には，Xeon L5520 2.27 GHz，メモリ 12 GB のサーバを利用した．ヒューリスティックによるフィルタをかけない場合，クエリ・ID，URL・ID を保持する辞書のサイズがそれぞれ 2.8 GB，6.6 GB，グラフラプラシアンのサイズが 3.4 GB となった．フィルタ後のサイズはそれぞれ，1.2 GB，4.1 GB，1.5 GB となった．

[*6] 英数字はすべて半角小文字に正規化している．

語候補のペアも正解ラベルを与えた．1 つのクエリに対して複数正解がある場合には，正解ラベルの値は 1/正解の数とした．

最終的に，4,836 件のランキング学習データを用意した．

5.3 評価

評価尺度には，precision@k を用いた．ベースラインには，Uchiumi ら [30] の Noisy Channel Model（NCM）を用いた．Noisy Channel Model については，ラベル伝搬のスコアを Channel Model として使用するため，獲得された候補をすべて用いた場合は必ず 1 位に入力クエリがきてしまう．そのため，Noisy Channel Model の評価では獲得された候補から入力クエリは除外した．ListNet（LN），および NeuroListNet（NLN）の 2 つの手法について，5 分割交差検定で評価した．ListNet は，Noisy Channel Model で使用する言語モデルとラベル伝搬のスコアの 2 つのみを素性として使用した場合と，素性テンプレートによって表 1 であげた素性を組み合わせた 52 の素性を用いた場合の 2 パターンについて評価した．使用した素性テンプレートを**表 2** に示す．

NeuroListNet は，中間層のゲート関数の数 g をそれぞれ 100，500，1,000，2,000，3,000 と変化させて評価した．各ゲート関数では，各同義語候補の素性からランダムに w 個を取り出して組み合わせた．w の値はそれぞれ 3，5，7 と変化させて評価した．

学習率およびペナルティの重みは予備実験より，それぞれ $\eta_0 = 1$，$\lambda = 0.00001$ とした．

5.3.1 実験結果

実験結果を**表 3** に示す．Noisy Channel Model から ListNet へ変更することで，精度が向上されていることが分かる．また，素性テンプレートによる素性の組合せを入れることで，精度が改善されている．比較した手法の中では，$k = 1$，3，4 では $w = 5$，$g = 3{,}000$ とした NeuroListNet が，$k = 2$，5 では，$w = 7$，$g = 3{,}000$ とした NeuroListNet が最も良い精度となった．

実験結果では，中間層のゲート関数の数を増やすことで精度が向上している．これより，提案手法によって適切な素性の組合せを獲得できていることが分かる．

5.4 考察

生成モデルから識別モデルへ変更した効果，素性テンプレートを導入した効果，非線形拡張による効果の特徴をそれぞれ調べるため，**表 4** にそれぞれの手法で 1 位にランキングできるようになった同義語の例，**表 5** に Noisy Channel Model から ListNet への変更で 1 位にランキングできなくなった同義語の例，**表 6** に学習器を変えても 1 位

*7　t は同義語リストの相対位置を表す．

表 2　ListNet による同義語抽出に用いた素性テンプレート
Table 2　Feature templates for ListNet based synonym extraction.

ID	テンプレート	ID	テンプレート
0	C	26	$f_{t,3} \times f_{t,7} \times f_{t,10} \times f_{t,12}$
1	$f_{t,0}$[*7]	27	$f_{t,3} \times f_{t,8} \times f_{t,10}$
2	$f_{t,2} - f_{t,1}$	28	$f_{t,3} \times f_{t,8} \times f_{t,10} \times f_{t,12}$
3	$f_{t,3}$	29	$f_{t,3} \times f_{t,4}$
4	$f_{t,4}$	30	$f_{t,3} \times f_{t,5}$
5	$f_{t,5}$	31	$f_{t,3} \times f_{t,6}$
6	$f_{t,6}$	32	$f_{t,3} \times f_{t,7}$
7	$f_{t,7}$	33	$f_{t,3} \times f_{t,8}$
8	$f_{t,8}$	34	$f_{t,3} \times f_{t,9}$
9	$f_{t,9}$	35	$f_{t,3} \times f_{t,10}$
10	$f_{t,10}$	36	$f_{t,3} \times f_{t,11}$
11	$f_{t,11}$	37	$f_{t,4} \times f_{t,5}$
12	$f_{t,12}$	38	$f_{t,4} \times f_{t,6}$
13	$f_{t,13}$	39	$f_{t,4} \times f_{t,7}$
14	$f_{t,14}/\Sigma_i f_{i,14}$	40	$f_{t,4} \times f_{t,8}$
15	$f_{t,15}/\Sigma_i f_{i,15}$	41	$f_{t,4} \times f_{t,9}$
16	$f_{t,16}/\Sigma_i f_{i,16}$	42	$f_{t,5} \times f_{t,6}$
17	$f_{t,14} + f_{t,15}$	43	$f_{t,5} \times f_{t,7}$
18	$f_{t,14} + f_{t,16}$	44	$f_{t,5} \times f_{t,8}$
19	$f_{t,15} + f_{t,16}$	45	$f_{t,5} \times f_{t,9}$
20	$f_{t,14} + f_{t,15} + f_{t,16}$	46	$f_{t,6} \times f_{t,7}$
21	$f_{t,3} \times f_{t,4} \times f_{t,10}$	47	$f_{t,6} \times f_{t,8}$
22	$f_{t,3} \times f_{t,4} \times f_{t,10} \times f_{t,12}$	48	$f_{t,6} \times f_{t,9}$
23	$f_{t,3} \times f_{t,6} \times f_{t,10}$	49	$f_{t,7} \times f_{t,8}$
24	$f_{t,3} \times f_{t,6} \times f_{t,10} \times f_{t,12}$	50	$f_{t,7} \times f_{t,9}$
25	$f_{t,3} \times f_{t,7} \times f_{t,10}$	51	$f_{t,8} \times f_{t,9}$

表 3　同義語抽出における各学習器ごとの precision@k
Table 3　Precision@k of each method on synonym extraction.

k	1	2	3	4	5
NCM (baseline)	0.557	0.345	0.252	0.200	0.166
LN with NCM features	0.584	0.380	0.285	0.229	0.191
LN with feature templates	0.655	0.416	0.305	0.242	0.200
NLN ($g = 100, w = 3$)	0.443	0.313	0.240	0.197	0.167
NLN ($g = 100, w = 5$)	0.540	0.352	0.260	0.210	0.177
NLN ($g = 100, w = 7$)	0.470	0.307	0.234	0.192	0.164
NLN ($g = 500, w = 3$)	0.526	0.347	0.263	0.212	0.179
NLN ($g = 500, w = 5$)	0.657	0.409	0.299	0.236	0.196
NLN ($g = 500, w = 7$)	0.648	0.407	0.297	0.236	0.195
NLN ($g = 1{,}000, w = 3$)	0.615	0.390	0.290	0.231	0.192
NLN ($g = 1{,}000, w = 5$)	0.722	0.438	0.317	0.246	0.203
NLN ($g = 1{,}000, w = 7$)	0.723	0.438	0.316	0.247	0.203
NLN ($g = 2{,}000, w = 3$)	0.647	0.404	0.296	0.234	0.195
NLN ($g = 2{,}000, w = 5$)	0.724	0.444	0.319	0.249	0.204
NLN ($g = 2{,}000, w = 7$)	0.727	0.443	0.319	0.249	0.204
NLN ($g = 3{,}000, w = 3$)	0.624	0.394	0.292	0.232	0.193
NLN ($g = 3{,}000, w = 5$)	**0.735**	0.448	**0.321**	**0.251**	0.205
NLN ($g = 3{,}000, w = 7$)	0.728	**0.449**	0.319	0.250	**0.206**

表 4　1 位にランクできるようになった同義語の例
Table 4　Examples of synonymous which a discriminative model failed to rank at the top.

	synonyms (query) ranked top
NCM vs LN with NCM features	cosmic wonder (コズミックワンダー), ジャレコ (jaleco), zazen boys (ザゼンボーイズ), maccheronian (マカロニアン), dvd shrink (dvd シュリンク), heritage stone (ピエールアルディ), solatina (ソラチナ), sotte bosse (ソットボッセ), aquagirl (アクアガール), burberry (バーバリー), logovista (ロゴヴィスタ), halb (ハルブ), corona (コロナ), cornelius (コーネリアス), anya hindmarch (アニヤハインドマーチ), dior homme (ディオールオム), julepe (ジュレップス), muchacha (ムチャチャ), peter jensen (ピーターイェンセン), シュトレン (シュトーレン), new york hat (ニューヨークハット), banal chic bizarre (バナルシックビザール), shinhwa (シンファ), ferragamo (フェラガモ)
LN with NCM feature vs LN with feature templates	cookpad(クックパッド), count down tv (カウントダウン tv), canon (キャノン), バットレッド (vatled), metal gear solid (メタルギアソリッド), 駒澤大学 (駒大), タディ (tady), christian dior (クリスチャンディオール), ホワイトコミック (white comic), キングダムハーツ (kingdom hears), rhythm footwear (リズムフットウェア), プロケッズ (pro-keds), クレヨンしんちゃん (クレしん), 福島工業高等専門学校 (福島高専), 農業共同組合 (農協), 議員連盟 (議連), noble (ノーブル), ディスティニーズチャイルド (デスチャ), brown sugar (ブラウンシュガー), マキシマムザホルモン (maximum the hormone), santastic! (サンタスティック), ドレスキャンプ (dresscamp), ノーブル (noble), rat simons (ラフシモンズ), pe'z moku (ペズモク)
LN with feature templates vs NLN	日本平パーキングエリア (日本平 pa), バンブーブレード (bamboo blade) dig design (ディグデザイン), ポストペット (ポスペ), マジョリカマジョルカ (マジョマジョ), じんべい (甚平), 山陽自動車道 (山陽道), イマジナリーファンデーション (the imaginary foundation), chage and aska (チャゲアス), loto (ロト), ヴェネツィア国際映画祭 (ベネツィア映画祭), kingdom hearts birth by sleep (khbbs), mixi (ミクシィ), 近畿日本ツーリスト (近ツリ), christian lacroix (クリスチャンラクロワ), 大阪厚生信用金庫 (大阪厚生信金), pierre hardy (ピエールアルディ), 東海環状自動車道 (mag ロード), sierra designs (シェラデザイン), 身体障害者福祉法 (身障者福祉法), 漫画喫茶 (漫喫), charcoal filter (チャコールフィルター), 早稲田大学 (早大), ゲルマニウム (ゲルマ), 生駒山上遊園地 (スカイランドいこま)

にランキングできなかった同義語の例を示す[*8].

- **生成モデルから識別モデルへ変更した効果**
  表 4，表 5 より，Noisy Channel Model から ListNet へ変更したことにより，カタカナからアルファベットへの変換等，異表記の同義語の抽出が改善されていることが分かる．ListNet の素性には Noisy Channel Model 同様，クリックスルー素性とクエリ言語モデル素性のみを利用しており，抽出精度の改善は 2 つの間の重み付けを適切に行えたことを意味している．一方，改善できなかった事例には略語とその展開した文字列のペア，およびアルファベットからカタカナへの変換等が含まれており，カタカナからアルファベットへの変換を重視したモデルとなっていることが分かる．

- **素性テンプレートによる組合せの効果**
  表 1 の素性を追加し，素性テンプレートを用いてその組合せをランキング学習の素性として利用した場合では，クリックスルー素性とクエリ言語モデル素性のみを用いた場合と比べて，略語からの展開文字列の獲得で改善が見られている．また，異表記の同義語の獲

[*8] 括弧内は入力クエリを表す．

表 5　識別モデルへの変更で 1 位にランクできなくなった同義語の例
Table 5　Examples of synonyms which the latter model was able to rank at the top.

	synonyms (query) not ranked top
NCM vs LN with NCM features	イマジナリーファンデーション (the imaginary foundation), ホワイトコミック (white comic), キングダムハーツ (kingdom hearts), プロケッズ (pro-keds), kingdom hearts birth by sleep (khbbs), 近畿日本ツーリスト (近ツリ), クレヨンしんちゃん (クレしん), 大阪厚生信用金庫 (大阪厚生信金), 農業協同組合 (農協), デスティニーズチャイルド (デスチャ), マキシマムザホルモン (maximum the hormone), 東海環状自動車道 (mag ロード), ドレスキャンプ (dress camp), ノーブル (noble), 福島工業高等専門学校 (福島高専), 大牟田柳川信用金庫 (大牟田柳川信金), キングスフィールド (king's field), sg ワナビー (sg wannabe), 横浜国立大学 (横国大), 家庭裁判所 (家裁), イタリアントマト (イタトマ), 有給休暇 (有休), パーフェクトリポート (perfect report), 名神高速 (名神高速道路)

表 6　1 位にランクできなかった同義語の例（○は 1 位にランクできたことを表す）
Table 6　Examples of synonyms which not all methods were able to rank at the top (○ indicates that the method ranked the synonym at the top).

query	NCM	LN with NCM features	LN with feature templates	NLN
pig nose (ヒックノース)	×	×	×	×
dig design (ティクテサイン)	×	×	×	○
count down tv (カウントタウン tv)	×	×	○	×
イマシナリーファンテーション (the imaginary foundation)	×	×	×	○
ハットレット (vatled)	×	×	○	○
全米オーフンテニス (全米テニス)	×	×	×	×
タティ (tady)	×	×	○	○
モンクレール (モンクレ)	×	×	×	×
マショリカマショルカ (マショマショ)	×	×	×	○
疾走、ヤンキー魂。(ヤン魂)	×	×	×	×
ヴェネツィア国際映画祭 (ベネツィア映画祭)	×	×	×	○
アントレア・タミコ (andrea d' amico)	×	×	×	×
ameba なう (アメーバなう)	×	×	×	×
トラコンクエストモンスタースショーカー 2 (dqmj2)	×	×	×	×
zoo keeper (スーキーハー)	×	×	×	×
mixi (ミクシィ)	×	×	×	○
flanklin&marshall (フランクリンマーシャル)	×	×	×	×
狛江第一中学校 (狛江第一中学)	×	×	×	×
スターリースカイ (starry sky)	×	×	×	×
emporia armani (エンホリオアルマーニ)	×	×	×	×
le feter (ルフェテ)	×	×	×	×
bryan adams (フライアンアタムス)	×	×	×	×
gameboy advance (gba)	×	×	×	×
山陽自動車道 (山陽道)	×	×	×	○
chage and aska (チャケアス)	×	×	×	○
しんへい (甚平)	○	○	×	○
西名阪自動車道 (西名阪道)	×	×	×	×
futura laboratories (フューチュララホラトリース)	×	×	×	×
近畿日本ツーリスト (近ツリ)	○	×	×	○
iwc (インターナショナルウォッチカンハニー)	×	×	×	×

得についても，表 1 の素性を加えない場合ではカタカナからアルファベットへの置き換えが主であったのに対して，アルファベットからカタカナへ置き換えるパターンが改善されている．これは素性の抽出の際に文字種の違いを吸収するため，漢字やカタカナ，ひらがなについては読み仮名を推定し，ローマ字へ置き換えたことによる効果と考えられる．略語の展開については，クエリと同義語候補の間の頭字語の関係を見る IsAcronym 素性が効いていると予想できる．表 5 で見られた 1 位にランキングできなくなった事例につい

表 7　NLN で 1 位に当てられなかったクエリに対する同義語候補のランキング出力
Table 7　Outputs for queries which were not ranked at the top by NLN.

query	ranking of a correct candidate	ranking output （下線は正式表記，二重下線は異表記，波線は表記揺れ，破線は略語の関係を表す）
ピッグノーズ	2	ピックノーズ, pig nose, 長山時盛，ピグノーズ, pignose, ピッグノーズ
狛江第一中学	3	狛江市立狛江第一中学校, 狛江市立第一中学校, 狛江第一中学校, 狛江第一中, 狛江市第一中学校, 狛江一中
starry sky	5	starry ☆ sky, starry ☆ sky wiki, starry sky avex, スタスカ, スターリースカイ, starry ☆ sky youtube, starrysky
エンポリオアルマーニ	2	エンポリオ・アルマーニ, emporia armani, アルマーニ，エンポーリオアルマーニ, shop at emporioarmani.com
全米テニス	3	オープンテニス, テニス全米オープン, 全米オープンテニス, テニス，フラッシングメドウ, ニューステニス
モンクレ	2	モンクレー，モンクレール, デュベティカとモンクレール，monclerjapan.com, moncler duvetica
ルフェテ	2	フランシスコ・ルフェテ, le feter, le.feter, ルフェテ
ブライアンアダムス	2	ブライアン・アダムス, bryan adams, bryanadams, brayan adams, brian adams, bryan adams recless
gba	3	ゲームボーイアドバンス, game boy advance, gameboy advance, ゲームボーイアドバンスソフト，ゲームボーイアドバンスレビュー
西名阪道	2	西名阪道路, 西名阪自動車道, 名阪自動車道，西名阪国道, 西名阪自動車，西名阪, 松原本線，西名阪工事情報
ヤン魂	6	ヤンキー, ヤンキー魂, ヤンキーゲーム，疾走ヤンキー，ヤンキーオンラインゲーム, 疾走、ヤンキー魂。，魔球魔球
フューチュララボラトリーズ	2	フューチュラ，futura laboratories, futura, フューチュララボラトリーズ
インターナショナルウォッチカンパニー	7	インターナショナル腕時計，international watch co, international watch, インターナショナルウォッチ，international watch company, tanaka iwc, iwc
andrea d'amico	2	アンドレアダミコ, アンドレア・ダミコ, アンドレア・ダミーコ, アンドレアダミーコ, andrea damico
アメーバなう	2	アメーバナウ, ameba なう, アメブロなう，amebanow, アメーバ nau, スタッフなう，ameba now
ズーキーパー	5	zookeeper, zoo keeper ds, zookeeper ds, 対戦ズーキーパー，zoo keeper, ds zookeeper, 対戦 zookeeper
dqmj2	7	ジョーカー，ドラゴンクエストモンスターズジョーカー 2 プロフェッショナルチャット, ドラゴンクエストジョーカー 2, ドラゴンクエストジョーカー 2 攻略, ドラクエジョーカー 2, ドラクエモンスターズジョーカー 2, ドラゴンクエストモンスターズジョーカー 2
フランクリンマーシャル	3	フランクリン＆マーシャル, franklin marshall, franklin&marshall, フランクリン&マーシャル, franklin & marshall

ても，ランキング素性の追加とその組合せを用いることで改善されていることが分かる．

- **非線形拡張による効果**
NeuroListNet でも，改善の傾向は素性テンプレートをいれた ListNet に近く，略語の展開での改善が目立っている．素性テンプレートの追加では見られなかった改善として，「東海環状自動車道（mag ロード）」や「生駒山上遊園地（スカイランドいこま）」のような別名の改善がある．これは，今回追加したクエリと同義語候補の間の表層文字列に基づく素性だけでは単純には取り出せない．NeuroListNet では，中間層で素性の組合せを生成しており，別名の獲得はクリックスルー素性やクエリ言語モデル素性と表層文字列の素性を適切に組み合わることができたことによる効果と考えられる．

- **1 位にランキングできなかった事例に対する考察**
表 7 に，NeuroListNet で 1 位にランキングできなかった事例のランキング出力と，正解の同義語の位置を示す．正解の多くは 2 位から 3 位に取り出せている．また，1 位にランキングされている同義語候補を見ると，「狛江第一中学」に対する「狛江市立狛江第一中学校」は正解として与えられているものよりも適した正式名称となっている．「エンポリオアルマーニ」に対しては，中黒を含む「エンポリオ・アルマーニ」を 1 位に出

力している．これは表記ゆれである．「ズーキーパー」に対する「zookeeper」は異表記への置き換えとして正しい．ここであげた例は，ラベル伝搬で獲得した候補のうち，人手で作成した同義語辞書に含まれる候補のみを正解としたことで，辞書に含まれていない，本来正解となるべき候補が誤りとしてラベル付けられていることに起因している．

「ピッグノーズ」に対する「ピックノーズ」は入力誤りの例である．「モンクレ」や「フューチュララボラトリーズ」に対する「モンクレー」，「フューチュラ」は，正解表記や入力クエリの部分文字列となっている．「インターナショナルウォッチカンパニー」に対する「インターナショナル腕時計」は意味が変化している．「dqmj2」に対する「ジョーカー」は略語となっているが，一般名詞でもある．そのため，これらはクエリ拡張としては適さない．これらを適切にランキングで下位に落とすには，クエリと同義語候補が部分文字列の関係になっていないかや，2 つの間の編集距離を素性に入れることが考えられる．

「インターナショナルウォッチカンパニー」に対する「iwc」は略語化であるが，学習データには略語への置き換えが略語の展開に比べると少ない．そのため，略語化を適切に行えるようにするためには，学習データの拡充が必要である．

6. まとめ

我々はランキング学習を用いた同義語獲得手法の提案を行った．提案手法では，これまでの Noisy Channel Model に基づく手法から識別モデルに基づく手法へ変更することで，クエリと同義語候補の表層に基づく素性の追加を可能とした．また，ランキング関数に中間層を導入することで，素性の組合せと重み付けを適切に行えるよう拡張した．実験によって，実際に素性の組合せが有効に働くことを示した．

本研究では，ランキング学習に Top 1 ListNet およびその非線形拡張を用いた．ListNet では，その後の研究で損失関数に対数尤度を用いることで，Top k ListNet の学習を効率的に行う方法が示されている．また，ランキング学習の研究では，集団学習を用いることで精度を高めることに成功している．Top k ListNet への拡張および集団学習の導入による高精度化は我々の今後の課題である，

参考文献

[1] Ahmad, F. and Kondrak, G.: Learning a spelling error model from search query logs, *Proc. conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pp.955–962 (2005).

[2] Bergsma, S. and Wang, Q.I.: Learning noun phrase query segmentation, *Proc. 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp.819–826 (2007).

[3] Bouma, G.: Normalized (pointwise) mutual information in collocation extraction, *Proc. Biennial GSCL Conference*, pp.31–40 (2009).

[4] Brin, S. and Page, L.: The anatomy of a large-scale hypertextual web search engine, *Computer networks and ISDN systems*, Vol.30, No.1-7, pp.107–117 (1998).

[5] Cao, H., Jiang, D., Pei, J., He, Q., Liao, Z., Chen, E. and Li, H.: Context-aware query suggestion by mining click-through and session data, *Proc. 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp.875–883 (2008).

[6] Cao, Z., Qin, T., Liu, T.-Y., Tsai, M.-F. and Li, H.: Learning to rank: from pairwise approach to listwise approach, *Proc. 24th international conference on Machine learning*, pp.129–136 (2007).

[7] Chen, Q., Li, M. and Zhou, M.: Improving query spelling correction using web search results, *Proc. 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp.181–189 (2007).

[8] Collins, M., Globerson, A., Koo, T., Carreras, X. and Bartlett, P.: Exponentiated gradient algorithms for conditional random fields and max-margin markov networks, *The Journal of Machine Learning Research*, Vol.9, pp.1775–1822 (2008).

[9] Duchi, J. and Singer, Y.: Efficient online and batch learning using forward backward splitting, *Journal of Machine Learning Research*, Vol.10, pp.2899–2934 (2009).

[10] Gao, J., Li, X., Micol, D., Quirk, C. and Sun, X.: A large scale ranker-based system for search query spelling correction, *Proc. 23rd International Conference on Computational Linguistics*, pp.358–366 (2010).

[11] Guo, J., Xu, G., Li, H. and Cheng, X.: A unified and discriminative model for query refinement, *Proc. 31st annual international ACM SIGIR conference on Research and Development in Information Retrieval*, pp.379–386 (2008).

[12] Hagiwara, M. and Suzuki, H.: Japanese query alteration based on semantic similarity, *Proc. Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp.191–199 (2009).

[13] Jain, A. and Pennacchiotti, M.: Open entity extraction from web search query logs, *Proc. 23rd International Conference on Computational Linguistics*, pp.510–518 (2010).

[14] Jones, R., Bartz, K., Subasic, P. and Rey, B.: Automatically generating related queries in japanese, *Language resources and evaluation*, Vol.40, No.3, pp.219–232 (2006).

[15] KAKASI, available from ⟨http://kakasi.namazu.org/index.html.ja⟩.

[16] Kleinberg, J.: Authoritative sources in a hyperlinked environment, *J. ACM*, Vol.46, No.5, pp.604–632 (1999).

[17] Komachi, M., Makimoto, S., Uchiumi, K. and Sassano, M.: Learning semantic categories from click through logs, *Proc. 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing for*

the Asian Federation of Natural Language Procesing: Short Papers, pp.189–192 (2009).
[18] Lafferty, J., McCallum, A. and Pereira, F.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data, *Proc. 18th International Conference on Machine Learning*, pp.282–289 (2001).
[19] Li, M., Zhang, Y., Zhu, M. and Zhou, M.: Exploring distributional similarity based models for query spelling correction, *Proc. 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pp.1025–1032 (2006).
[20] Mei, Q., Zhou, D. and Church, K.: Query Suggestion using hitting time, *Proc. 17th ACM conference on Information and Knowledge Management*, pp.469–478 (2008).
[21] Mihalcea, R. and Terau, P.: Textrank: Bringing order into text, *Proc. 2004 Conference on Empirical Methods in Natural Language Processing*, pp.404–411 (2004).
[22] Murayama, N. and Okumura, M.: Statistical model for Japanese abbreviations, *Proc. 10th Pacific Rim International Conference on Artificial Intelligence: Trends in Artificial Intelligence*, pp.260–272 (2008).
[23] Okazaki, N., Ishizuka, M. and Tsujii, J.: A discriminative approach to Japanese abbreviation extraction, *Proc. 3rd International Joint Conference on Natural Language Processing* (*IJCNLP-08*), pp.889–894 (2008).
[24] Peng, F., Ahmed, N., Li, X. and Lu, Y.: Context sensitive stemming for web search, *Proc. 30th annual international ACM SIGIR conference on Research and Development in Information Retrieval*, pp.639–646 (2003).
[25] Risvik, K.M., Mikolajewski, T. and Boros, P.: Query segmentation for web search, *Poster Session in The 12th International World Wide Web Conference* (2003).
[26] Sekine, S. and Suzuki, H.: Acquring ontological knowledge from query logs, *Proc. 16th international conference on World Wide Web*, pp.1223–1224 (2007).
[27] Cucerzan S. and Brill, E.: Spelling correction as an iterative process that exploits the collective knowledge of web users, *Proc. Conference on Empirical Methods in Natural Language Processing*, pp.293–300 (2004).
[28] Sun, X., Gao, J., Micol, D. and Quirk, C.: Learning phrase-based spelling error models from click through data, *Proc. 48th Annual Meeting of the Association for Computational Linguistics*, pp.266–274 (2010).
[29] Suzuki, H., Li, X. and Gao, J.: Discovery of term variation in japanese web search queries, *Proc. 2009 Conference on Empirical Methods in Natural Language Processing*, Vol.3, pp.1484–1492 (2009).
[30] Uchiumi, K., Komachi, M., Machinaga, K., Maezawa, T., Satou, T. and Kobayashi, Y.: Japanese abbreviation expansion with query and clickthrough logs, *Proc. 5th International Joint Conference on Natural Language Processing*, pp.410–419 (2011).
[31] Wei, X., Peng, F., Tseng, H., Lu, Y. and Dumoulin, B.: Context sensitive synonym discovery for web search queries, *Proc. 18th ACM conference on Information and Knowledge Management*, pp.1585–1588 (2009).

内海 慶

2004 年図書館情報大学図書館情報学部図書館情報学科卒業．2006 年筑波大学大学院図書館情報メディア研究科博士前期課程修了．修士（情報学）．同年 4 月ヤフー株式会社入社．2012 年現在，同社在職中．自然言語処理の研究開発に従事．言語処理学会会員．

小町 守

2005 年東京大学教養学部基礎科学科科学史・科学哲学分科卒業．2007 年奈良先端科学技術大学院大学情報科学研究科博士前期課程修了．2008 年より日本学術振興会特別研究員を経て，2010 年同大学博士後期課程修了．博士（工学）．現在，同研究科助教．専門は自然言語処理．大規模なコーパスを用いた意味解析および統計的自然言語処理に関心がある．人工知能学会，言語処理学会，ACL 各会員．

（担当編集委員　豊田 正史）