意味的類似度を利用した日本語クエリ書き換え

萩原 正人*
名古屋大学
hagiwara@kl.i.is.nagoya-u.ac.jp

鈴木 久美
Microsoft Research
hisamis@microsoft.com

1　はじめに

Web 検索エンジンにおいて，クエリ訂正はロバストな検索を実現する上で重要な技術である．検索エンジンに入力されるクエリの 10%以上が誤りを含むと言われており [3]，これに対処するため，これまで英語のクエリ訂正手法が幅広く研究されてきた．しかし，英語以外の言語についてはほとんど着手されていない．

本稿では，日本語を対象とし，クエリ訂正よりさらに広い概念であるクエリ書き換えのタスクを扱う．日本語においては，図 1 に示すように，多数の文字種およびそれらに伴う異表記の問題が顕著であり，クエリの訂正は特に難しい課題である．これらの幅広い異表記を統一的に扱うことは，情報検索のみならず自然言語処理一般にとって重要であると考えられる．

日本語おける特に深刻な問題として，カタカナによる翻字の表記揺れが挙げられる．例えば「fedex」→「フェデックス」のような翻字よびその逆変換は機械翻訳に必須の処理であり，カタカナの表記揺れの獲得に関する多数の研究がなされている (e.g. [7])．一方，異なる文字種間での表記揺れ全般を統一的に取り扱った手法はこれまでに存在しない．

本稿では，日本語のクエリ訂正・書き換えための汎用的アプローチを提案する．本アプローチの目的は，あるクエリに対して書き換え候補を提示することであり，これにはスペルミスの訂正だけではなく，例えば「スパゲティ」と「スパゲッティ」などの異表記 (図 1 中 *Sp*) や,「MS」と「マイクロソフト」のような略語 (図 1 中 *Abbr*),「座席」と「シート」のような同義語 (図 1 中 *Syn*) も対象としている[1]．本アプローチは，綴りおよび意味の上での類似度を扱う点において，従来の英語クエリ訂正手法に基づいているが，入力クエリと綴りの全く類似していない書き換え候補も扱うことができる点においてより汎用的である．意味的な類似度を計算する際には，意味カーネル法 [6] を用いて，クエリ書き換えの精度の向上を図る．本稿ではまた，クエリと書き換え候補のペアの正解セットを，検索セッションログから効率的に作成する手法を新たに提案する．

2　関連研究

Web クエリにおいては未知語および新語の問題が顕著であるため，汎用の辞書に頼った訂正ができないという点で，従来のスペル訂正手法とは大きく異なっている．クエリ訂正に関する先行研究 [3] は，辞書ベースのスペル訂正は，クエリの訂正には適さないことを示している．またこの研究では，クエリ訂正を雑音のある通信路モデルを用いて定式化し，言語モデルとして検索ログから学習した単語バイグラムモデルを，誤りモデルとして重み付き編集距離を用いている．また，Brill and Moore [1] は，部分文字列の書き換えを考慮した汎用的な誤りモデルを提案しており，これにより高精度なスペル訂正が可能であることを示した．

Li et al. [8] は，Cucerzan and Brill [3] の手法を発展させ，クエリと訂正候補間の意味的類似度も考慮できるモデルを提案している．彼らは，クエリ「adventura」は,「adventure」ではなく「aventura」のスペルミスである場合が多いが，これを正しく訂正するためには，語の意味的な類似度に関する知識が必要であることを指摘した．また，スペル訂正に対する識別モデルを提案し，従来の生成モデルと比べ訂正精度を改善できることを示した．この結果を受け，本稿においても最大エントロピー法に基づく識別モデルを用いる．

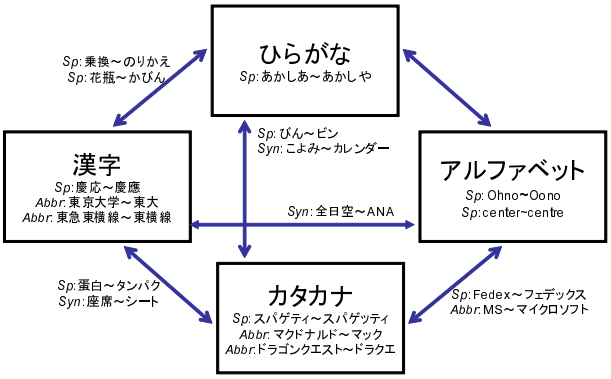

図 1: 日本語における文字種と異表記

3　クエリ書き換えモデル

3.1　問題の定式化

本稿では，クエリ書き換えのタスクを，従来のクエリ訂正と同様に定式化する．本モデルでは，クエリ文字列 q を入力として与え，以下の事後確率を最大化する c^* を正しい書き換え候補として出力する：

$$c^* = \arg \max_{c \in \mathrm{CF}(q) \subset C} P(c|q) \quad (1)$$

C は書き換え候補の全体集合であり，本稿では検索ログから抽出した空白区切りの語およびそのバイグラムから成る．$\mathrm{CF}(q)(\subset C)$ は q の近傍集合であり，$\mathrm{CF}(q) = \{c \in C | \mathrm{ED}(q,c) < \theta\}$，すなわち，$q$ からある一定の編集距離以内にある書き換え候補の集合として求める．実験では正規化無しの編集距離を用い，$\theta = 24$ とした．この編集距離 ED の詳細は次節で述べる．c^* として入力 q と異なるものが出力された場合，クエリの書き換えが起こることになり，本モデルにより誤りの検

*本研究は，萩原の Microsoft Research でのインターンシップ期間中に行われた．

[1]本アプローチによって獲得された書き換え候補は，クエリの書き換え，拡張，および代替クエリの提示などに用いることができるが，獲得された候補を情報検索においてどのように用いるかについてはタスク依存であり，本稿では議論しない．

出および書き換えが同時に扱えるという利点がある．

3.2 雑音のある通信路モデル

雑音のある通信路モデルは，クエリ訂正の定式化に広く用いられている [1, 3]．本モデルもこのアプローチに従い，式 (1) を，ベイズ則を用いて以下のように分解する：

$$c^* = \arg \max_{c \in \mathrm{CF}(q) \subset C} P(c)P(q|c), \tag{2}$$

ここで，言語モデル $P(c)$ は書き換え候補 c 自体の妥当性を，誤りモデル $P(q|c)$ は q と c の類似度を与える．

言語モデルとしては，英語では単語 n グラムによる統計的言語モデルが用いられてきた．しかし，日本語クエリの単語分割は難しいため，本手法ではクエリ文字列全体を書き換えの対象と見なし，$P(c)$ は以下のように検索ログ中の相対頻度を用いて計算する：

$$P(c) = \frac{\mathrm{Freq(c)}}{\sum_{c' \in C} \mathrm{Freq(c')}}. \tag{3}$$

また，誤りモデルには，通常の編集距離の一般化であるアルファ・ベータ法 [1] を用いた．アルファベータ法では，$\alpha \to \beta(\alpha, \beta$ は長さ 0 以上の文字列) の形の文字列書き換えに対して，$P(\alpha \to \beta|\mathrm{PSN})$ を書き換えペアから学習する．ここで，PSN は α の単語中における位置であり，語頭・語中・語末のいずれかである．本モデルを用いて，文字列 w を文字列 s に書き換える確率は，以下のように求められる：

$$P_{\alpha\beta}(s|w) = \max_{R \in \mathrm{Part}(w), T \in \mathrm{Part}(s)} \prod_{i=1}^{|R|} P(R_i \to T_i|\mathrm{PSN}(R_i)), \tag{4}$$

これは，語 w, s の全ての分割 $\mathrm{Part}(w), \mathrm{Part}(s)$ から，書き換え確率を最大にする分割 R, S を見つける問題に相当する．アルファ・ベータ法はスペル訂正 [1] および翻字 [2] のタスクにおいて優れた性能を示した．

本稿では，英語と日本語の対応が取れた Wikipedia 記事のタイトル 59,754 組から本モデルを学習する．また $|\alpha|, |\beta| \leq 3$ とし，カタカナをヘボン式によりアルファベットへと変換した後，$ED_{\alpha\beta}(q|c) = -\log P_{\alpha\beta}(q|c)$ によって編集距離を計算する．q および c には英語・カタカナの両方が考えられるため，両方向の編集距離を考慮した．また，翻字の表記揺れは，「スパゲティ(spageti)」と「スパゲッティ(spagetti)」のように，ほとんどが文字の重複 $a \to aa$ および縮退 $aa \to a$ の違いでしかないため，これらについてペナルティを与えないように修正した編集距離 $\mathrm{ED_{hd}}$ を導入する．以上をふまえて，最終的な誤りモデルとして，

$$\begin{aligned} \mathrm{ED}(q,c) &= \min[\mathrm{ED}_{\alpha\beta}(q|c), \mathrm{ED}_{\alpha\beta}(c|q), \mathrm{ED_{hd}}(q,c)], \\ P(q|c) &= \exp[-\mathrm{ED}(q,c)] \end{aligned} \tag{5}$$

を用いた．ただし，ここでの編集距離はクエリと書き換え候補の文字列長 $2/(|q||c|)$ によって正規化する．

3.3 カーネル法による意味的類似度

前節で述べた雑音のある通信路モデルでは，クエリと書き換え候補間の意味的な類似度を扱うことができない．そこで，文脈から計算される語の意味的類似度

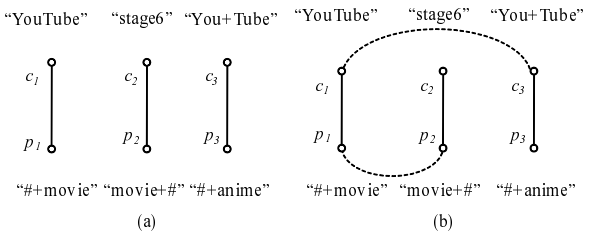

図 2: (a) 書き換え候補とパターンの接続グラフの例 および (b) 補完後のグラフ

である分布類似度を用いた．意味的類似度の計算に用いる文脈は，[9] と同様に，検索ログにおいて対象の文字列を除いた残りとして定義した．例えば，クエリ「JAL+航空券」における「JAL」の文脈は「#+ 航空券」となる．文脈を抽出した後，書き換え候補 c_i に対してベクトルを $\boldsymbol{c}_i = [\mathrm{pmi}(c_i, p_1), \ldots, \mathrm{pmi}(c_i, p_M)]'$ として構築する．ここで，M は異なり文脈の数，pmi は以下で与えられる自己相互情報量である：

$$\mathrm{pmi}(c_i, p_j) = \log \frac{|c_i, p_j|}{|c_i, *||*, p_j|}, \tag{6}$$

ただし，$|c_i, p_j|$ は書き換え候補 c_i とパターン p_j が共起する回数であり，*はワイルドカードを表す．$\boldsymbol{c}_i$ を L2 正規化したベクトル $\hat{\boldsymbol{c}}_i$ から構成される，書き換え候補とパターンの共起行列を $X = \{\hat{\boldsymbol{c}}_i\}$ とする．

この文脈定義は，分かち書き等を用いないためロバストで汎用的であるが，パターン側にもスペルミスや表記揺れ，語順の入れ替え等が含まれる可能性があり，スパースネスの問題が生じる．例えば，「YouTube+movie」「movie+YouTube」「YouTube+movii」(movie のスペルミス) の 3 つのクエリからは，「YouTube」に対して「#+movie」「movie+#」「#+movii」の 3 つの異なったパターンが抽出されるが，これらはベクトル空間において完全に独立の次元として扱われてしまう．

そこで，類似度グラフ上での伝播をモデル化し，書き換え候補間やパターン間の相関を扱うことのできる意味カーネル法 [6] を用いる．Kandola et al. [6] は，減衰係数の異なる 2 つの意味カーネル，**ノイマンカーネル** $\hat{K}$ および**拡散カーネル** $\tilde{K}$ を提案しており，それぞれ以下のように計算される：

$$\hat{K}(\beta) = K(I - \beta K)^{-1} = \sum_{t=1}^{\infty} \beta^{t-1} K^t \tag{7}$$

$$\tilde{K}(\beta) = K \sum_{t=1}^{\infty} \frac{\beta^t K^t}{t!} = K \exp(\beta K) \tag{8}$$

ここで，$K = X'X$ は書き換え候補間の類似度行列，β は減衰係数である．

これらの意味カーネルを使用することにより，スパースネスはある程度軽減されるが，スパースな接続グラフにおいては依然として問題が発生する場合がある．図 2(a) はその典型的な例であり，$K = X'X = I$ となり書き換え候補間の相関が考慮できない．そこで，書き換え候補間およびパターン間の綴りの類似度を利

用して，グラフを図 2(b) のように補完する．この補完は，類似度行列 $K = X'X$ の代わりに，

$$K^+ = \gamma S_C + (1-\gamma)X'[\delta S_P + (1-\delta)I]X \quad (9)$$

を用いることにより実現できる．ここで，$S_C = \{s_c(i,j)\}$ は，書き換え候補間の綴りの類似度行列であり，(i,j) 要素は $s_c(i,j) = \exp[-\mathrm{ED}(c_i,c_j)]$ によって与えられる．パターンの綴りの類似度行列 S_P も同様に計算する．K^+ を用いたカーネルは，補完グラフ上のランダムウォークモデルによって説明することができ，パラメータ γ, δ はそれぞれ，書き換え候補間とパターン間の補完の混合比を調節する．この補完類似度行列を K の代わりに用い，式 (7), (8) と同様に**補完ノイマンカーネル**および**補完拡散カーネル**を計算する．

3.4 ブートストラップによる書き換え候補の抽出

クエリ訂正の従来手法は，綴りの類似度のみを手がかりとしているため，意味的にしか類似していない書き換え候補を抽出できないという問題点がある．そこで本手法では，文脈パターンに基づいた意味カテゴリ抽出アルゴリズムである *Espresso*[10] の拡張版 *Tchai*[9] を用いて，綴りの類似度だけでは獲得できない書き換え候補を抽出した．

具体的には，雑音のある通信路モデルにより選択された上位 50 個のインスタンスを，予めシード C_0 として与える．シードインスタンスの信頼度は $P(c)P(q|c)$ により計算する[2]．次に，C_0 を用いて 100 個のパターン P_0 を抽出する．この際，200 個以上の異なりインスタンスと共起するものはジェネリック・パターンと見なし，P_0 には含めない．続いて，P_0 を用いてインスタンスを抽出し，C_1 を得る．最後に，C_1 に対してパターン集合 P_1 を求め，和集合 $C_0 \cup C_1$ と P_1 を意味カーネルの計算に用いる．ただし，信頼度 0.0001 以下のパターンは P_1 には含めず，また，P_1 の最大サイズは 2,000 に固定した．

3.5 最大エントロピー法

本手法では，統一的な確率モデルを構築するために，[8] と同様に最大エントロピー法を用いた．最大エントロピー法では，条件付き確率 $P(c|q)$ を，素性 $f_1, \ldots, f_K$ を用いて以下のように求める：

$$P(c|q) = \frac{\exp \sum_{i=1}^{K} \lambda_i f_i(c,q)}{\sum_c \exp \sum_{i=1}^{K} \lambda_i f_i(c,q)} \quad (10)$$

ここで，$\lambda_1, \ldots, \lambda_K$ は各素性に対する重み係数であり，訓練セットの対数尤度を最大化するように決定される．係数の最適化には GIS (Generalized Iterative Scaling) アルゴリズムを用いた．なお，GIS においては，全ての書き換え候補 C に関する正規化ステップが必要であるが，検索ログから抽出される候補数は非常に多い．そこで，[8] と同様，近傍集合 $C(q)$ に関する正規化によりこのステップを近似した．

素性としては，以下の 4 つのカテゴリを用いた：

1. 言語モデル確率素性：$f_{\mathrm{lang}}(q,c) = \log P(c)$
2. 誤りモデル確率素性：

$$f_{\alpha\beta}(q,c) = -\mathrm{ED}_{\alpha\beta}(q|c) \quad (11)$$

$$f_{\beta\alpha}(q,c) = -\mathrm{ED}_{\alpha\beta}(c|q) \quad (12)$$

$$f_{\mathrm{hd}}(q,c) = -\mathrm{ED}_{\mathrm{hd}}(q,c) \quad (13)$$

3. 類似度素性：$f_{\mathrm{sim}}(q,c) = \log \mathrm{sim}(q,c)$．ここで，$\mathrm{sim}(q,c)$ は $K, \hat{K}, \tilde{K}, \hat{K}^+, \tilde{K}^+$ のいずれかであり，定数 $\varepsilon = 1.0 \times 10^{-5}$ を加えた後 [0, 1] に正規化した．
4. 類似度に基づく訂正可能性素性：これは，c の頻度が q よりも高く，かつ c と q との分布類似度がある閾値よりも高い場合にのみ 1 となるような二値素性である．この素性値が 1 の場合，q が c の典型的なスペルミスであることを示唆している [8]．分布類似度の閾値が 0.5, 0.6, ..., 0.9 の計 5 個の素性を用いた．

4 評価実験

4.1 実験条件

実験には，2007 年 11 月と 12 月に Live Search に入力された検索クエリログの一部を用いた．頻度が 8 回未満のクエリは取り除き，残った延べ 83,080,257 個 (異なり数 1,038,499 個) のクエリを使用した．

クエリ訂正に関する従来手法では，クエリに対して正解となる書き換え候補を人手により作成し，モデルの訓練と評価を行っている．作成の際には，クエリを評価者に提示し，書き換えが必要な時はその正解を付与する．しかしこの作成法では，元のユーザー意図が分からないため，正解を与えるのが非常に難しいという欠点がある．例えば，*gogle* というクエリを提示された場合，それを *google* と *goggle* のどちらに訂正すべきかは不明である．そこで [3] では，検索ログを用い，同じユーザーによって連続して入力されたクエリの組 (q_1, q_2) に対して，q_1 と q_2 がある一定の編集距離以内のものを抽出し，それを評価者に提示することにより正解ペアを作成した．この手法はユーザー意図を考慮することができ信頼性が高いが，綴りの類似していないペアを排除してしまう．そこで本稿では，この手法を改善し，以下のように正解セットを作成した．

まず，検索ログから，クエリのペア (q_1, q_2) に対して，(1) 同じユーザーが 3 分以内に連続して入力したものであり，(2) q_1 の検索結果のクリック数が 0，q_2 は 1 以上であるようなものを集める．条件 (2) によって，q_2 は q_1 よりも適切なクエリであることが期待される．次に，こうして抽出されたクエリのペアに対し，q_1 と q_2 の従属性の指標である対数尤度比 (LLR; log-likelihood ratio)[4, 5] を計算する．最後に，LLR ≥ 200 であるようなクエリのペア 10,000 個を無作為に抽出し，評価者に提示する．評価は，提示されたペアに対して正解か不正解かを判断するだけであるため，1,000 ペアにつき 1 時間ほどで実施でき，非常に高速であった．また，2 人の評価者間の一致率も 95.7%と高かった．書き換えとして不適切なペアを除いた後，そのうち 1,243 個，628 個，4,618 個をそれぞれテストセット，dev セット，訓練セットとして使用した[3]．

[2] 実際には，言語モデルの確率に偏るのを防ぐため，$P(c)$ に係数 0.1 をかけて使用している．

[3] カーネルのパラメータ β, γ, δ は，dev セットを用いて最適化した．最終的に用いた値は，$\hat{K}, \tilde{K}, \hat{K}^+$ に対しては $\beta = 0.3$，$\tilde{K}^+$ に対しては $\beta = 0.2$，また，$\hat{K}^+$ に対しては $\gamma = 0.2, \delta = 0.4$，$\tilde{K}^+$ に対しては $\gamma = 0.35, \delta = 0.7$ である．

表 1: 各モデルの性能比較 (%)

モデル	正解率	再現率	精度
SC	75.30	48.61	55.78
ME-NoSim	79.49	56.17	66.17
ME-Cos	79.32	**58.19**	64.35
ME-vN	79.24	57.18	65.42
ME-Exp	78.52	56.42	63.64
ME-vN+	**79.89**	55.67	66.57
ME-Exp+	79.81	54.91	**66.67**

なお，書き換えには一般に複数の正解 (e.g.,「2tyann」に対して「2 ちゃん」「2 ちゃんねる」等) が考えられるため，テストセットによる評価の際には，書き換えの推移関係も考慮した．具体的には，正解セットにおいて q_1 が q_2 に書き換えられ，さらに q_2 が q_3 に書き換えられる場合，$q_1 \rightarrow q_3$ も正解ペアとした．

評価には，以下の 3 つの指標を使用した [8].

- 正解率：テストセット中のクエリのうち，システムによって正しく書き換えられたものの割合．
- 再現率：テストセット中の書き換えが必要なクエリのうち，システムによって正しく書き換えられたものの割合．
- 精度：システムによって書き換えられたクエリのうち，正しいものの割合．

4.2 結果

表 1 に，各モデルの性能を示す．SC は雑音のある通信路モデルであり，その他は最大エントロピー法 (ME) である．ME-NoSim はその中でも最も単純なモデルであり，SC と同じ素性を用いているが，3 つの評価指標全てにおいて有意に性能が向上しており (マクネマー検定, $p < 0.0001$)，最大エントロピー法による学習が有効であることを示している．3 種類の編集距離を別々の素性として用いていることも，性能向上の理由として考えられる．

言語モデルおよび誤りモデルの素性に加え，コサイン類似度を用いた ME-Cos では，ME-NoSim に比べて再現率が上昇している．しかし，クエリ書き換えのタスクにおいてより重要であると考えられる正解率・精度は低下している．この傾向はノイマンカーネル (ME-vN) および拡散カーネル (ME-Exp) を用いた手法においても同じである．ここから，意味的類似度を含めることにより必ずしも性能が向上するわけではないことが分かるが，これは検索ログから得られた文脈がスパースであることが原因であると考えられる．

一方，補完ノイマンカーネル (ME-vN+) および補完拡散カーネル (ME-Exp+) を用いた場合，分布類似度や通常のカーネル法と比較して再現率の低下を抑えつつ正解率・精度が向上している．特に ME-Exp+は，ME-Exp と比較して有意に高い性能を示した ($p < 0.01$)．

なお，本手法における正解率は，従来手法 ([8] で 80%以上) と比較すると一見低いが，これは本手法で用いたデータセットの作成法が，書き換えの必要なペアを多く選ぶ傾向があるためである．入力と同じクエリを出力する最も単純なベースラインの正解率は 67.3%であり，[8] の 83.4%に対して低くなっている．

具体的な出力に注目すると，意味的類似度を用いないモデルでは，綴りの類似度のみに依存しているため，「ハリポタ」→「ハリーポッター」などの書き換えに失敗しているのに対し，意味的類似度を用いたモデルでは正しく書き換えられている．一方で,「フィギア」→「フィギュアスケート」,「サンドイッチ」→「サンドイッチマン」などの書き換えは，場面限定的な書き換えであり，いずれのモデルも失敗している．また,「マック」には,「マクドナルド」「マッキントッシュ」の 2 つの正解が考えられるが，このような曖昧性のあるクエリの扱いは今後の課題である．

5 おわりに

本稿では，日本語クエリ書き換えの統一的なアプローチを提案した．本手法は英語のクエリ訂正の従来手法に基づいているが，カタカナの表記揺れや翻字等に限らず，同義語や翻訳など，表記・綴りの上では異なるが意味的に類似した候補への書き換えを扱うことができる．また，コサイン類似度のような単純な意味的類似度では，意味的相関を捉えられない場合があることを示し，綴りの類似度によって補完した意味カーネル法が書き換え性能の向上に有効であることを示した．

なお，スパースネスの問題を解消するための手法として，pLSI のような潜在意味モデルが考えられるが，これらの手法との比較は今後の課題である．また，本手法の利点として，モデルやデータを追加することにより容易に拡張できるという点がある．今後，漢字の読みモデルや検索ログからの統計情報などを素性として追加することにより，さらに高精度なクエリ書き換え手法を実現する予定である．

参考文献

[1] Eric Brill and Robert C. Moore. An Improved Error Model for Noisy Channel Spelling. *Proc. of ACL 2000*, pp. 286–293, 2000.

[2] Eric Brill, Gary Kacmarcik and Chris Brockett. Automatically Harvesting Katakana-English Term Pairs from Search Engine Query Logs. *Proc. of NLPRS 01*, 393–399, 2001.

[3] Silviu Cucerzan and Eric Brill. Spelling Correction as an Iterative Process That Exploits the Collective Knowledge of Web Users. *Proc. of EMNLP 2004*, pp. 293–300, 2004.

[4] Ted Dunning. Accurate Methods for the Statistics of Surprise and Coincidence. *Computational Linguistics*, vol. 19, num. 1, pp. 61–74, 1993.

[5] Rosie Jones, Benjamin Rey, Omid Madani, and Wiley Greiner. Generating Query Substitutions. *Proc. of WWW 2006*, pp. 387–396, 2006.

[6] Jaz Kandola, John Shawe-Taylor, and Nello Cristianini. Learning semantic similarity. *Neural Information Processing Systems (NIPS 15)*, pp. 657–664, 2002.

[7] Takeshi Masuyama, Satoshi Sekine, and Hiroshi Nakagawa. Automatic Construction of Japanese KATAKANA Variant List from Large Corpus. *Proc. of COLING 2004*, pp. 1214–1219, 2004.

[8] Mu Li, Muhua Zhu, Yang Zhang, and Ming Zhou. Exploring Distributional Similarity Based Models for Query Spelling Correction. *Proc. of COLING/ACL 2006*, pp. 1025–1032, 2006.

[9] Mamoru Komachi and Hisami Suzuki: Minimally Supervised Learning of Semantic Knowledge from Query Logs. *Proc. of IJCNLP 2008*, pp. 358–365, 2008.

[10] Patrick Pantel and Marco Pennacchiotti: *Espresso*: Leveraging Generic Patterns for Automatically Harvesting Semantic Relations. *Proc. of ACL 2006*, pp. 113–120, 2006.