1H1-2

名詞句における用語バリエーションの自動認識

Automatic Recognition of Term Variations in Noun Phrases

岡崎 直観[*1] 辻井 潤一[*1*2*3]
Naoaki Okazaki Jun'ichi Tsujii

[*1]東京大学 The University of Tokyo
[*2]英国マンチェスター大学 University of Manchester, UK
[*3]英国国立テキストマイニングセンター National Centre for Text Mining, UK

This paper presents a supervised machine-learning approach for normalizing terms into their representative forms. This study models the process of term variation with a set of possible operations of string-replacement, that convert a term (e.g., *activities*) into its representative form (e.g., *activity*). We formalize the task of recognizing term variation as a binary classification problem, in which a logistic regression model assigns a positive class to a given term pair only when a term of the pair is the representative form of another term. We use the maximum a posteriori (MAP) method with L_1 regularization for estimating contributions (weights) of string-replacement operations for normalizing terms. Training corpora for normalizing spelling and inflection variants are built from existing lexicons. The experimental results show the effectiveness of string-replacement operations obtained from the training process for both recognizing and normalizing term variations.

1. はじめに

自然言語を扱う様々なアプリケーションは，同じエンティティが表記の異なる複数の用語によって参照される問題（用語バリエーション）に直面する．例えば，病名辞書を用い，テキスト中の病名を識別子に変換する処理では，テキスト中に含まれる表記（例えば *tumour of lung*）が，病名辞書にそのままの収録されているとは限らないので，辞書に収録されていそうな表記（例えば *lung tumor*）に変形する必要がある [Yeganova 04]．一方，複数のデータベースを一つに統合する処理では，同じエンティティを指している複数のフィールド値を認識し，冗長なレコードを生成しないようにすることが望ましい [Bilenko 03]．

ある語を，同じエンティティを指すと思われる別の語に変換する処理は，近似文字列マッチング [Navarro 01]，スペル訂正 [Brill 00]，検索クエリ訂正 [Chen 07] 等のタスクと関連が深い．また，文字列同士の類似度を学習コーパスから獲得する研究も盛んに行われている．McCallum ら [McCallum 05] は，編集距離における操作を条件付き確率場で学習する手法を提案した．Bergsma ら [Bergsma 07] は，語を構成する部分文字列の組み合わせを素性とし，異なる言語間で起源が同じ語の組を認識する手法を提案した．鶴岡ら [Tsuruoka 08] は，語をその標準形に変換するのに必要な文字列置換ルールの汎用性・曖昧性を計る尺度を設計した．荒牧ら [Aramaki 08] は，２つの語の差分文字列や編集距離を素性として，医学用語の表記揺れを識別する分類器を構築した．

本研究では，２つの用語がバリエーションの関係にあるか判別するタスク（バリエーションの識別）に対し，文字列置換ルールを説明変数（素性）としたロジスティック回帰モデルを適用する．回帰モデルのパラメータ推定には，L_1 正則化に基づく事後確率最大化を利用し，バリエーションの識別に貢献する素性を絞り込む．さらに，学習で獲得した置換ルールの重みを活用し，与えられた語に対する標準形の候補を生成するアルゴリズムを提案する．

連絡先: 岡崎直観 <okazaki at is i u-tokyo ac jp>
東京都文京区本郷 7-3-1
東京大学大学院情報理工学系研究科
(03) 5841-4120

2. 提案手法

Daille ら [Daille 96] は，英語の用語バリエーションを分析し，文字変化（graphical variations），綴り変化（orthographic variations），語尾変化（inflectional variations），文法変化（syntactic variations），語－文法変化（morpho-syntactic variation）に分類した．文字変化は大文字・小文字の違い，ハイフンや記号の利用によって引き起こされる表記揺れであり，大文字から小文字への変換，記号から空白への置換などの簡単な処理で取り扱える．綴り変化は，米国式－英国式綴りの違い（例えば *normalize—normalise*），ギリシャ語，ラテン語，フランス語からの翻字の差（例えば *leukaemia—leukemia*）に由来する．語尾変化は，名詞の複数形，動詞の現在分詞形（名詞化）など，語の活用に基づく．文法変化は，語の挿入や並び替え（例えば *elements of matrix—matrix element*）によって発生する用語バリエーションである．語－文法変化は，形容詞から名詞への変更（例えば *enzymatic activity—enzyme activities*）や，動詞から名詞への変更によって引き起こされる．

用語のバリエーションを，変化する要素という観点でまとめると，文字レベルで生じるもの（綴り変化や語尾変化），単語レベルで生じるもの（文法変化），それらの複合で生じるもの（語－文法変化）に分類できる [Jacquemin 99]．本稿では，用語バリエーションの主要因である語形変化（paradigmatic variations）を機械学習でモデル化する．なお，類義語（例えば *carcinoma* と *cancer*），略語（例えば *estrogen receptor* と *ER*）に由来する用語バリエーションは，本稿で扱わない．

2.1 単語を代表形に変換する置換ルール

単語の語形変化を認識・生成する方法として，ステミング [Porter 80] や文字列距離尺度 [Levenshtein 66] がよく用いられる．ステミングは単語の語尾変化を取り除く有力な手段であるが，語幹に存在する綴り変化を扱えない．文字列距離尺度は，２つの単語の構成文字の一致度や，一方の単語を他方の単語に変更するために必要な置換操作[*1]の数で，単語間の距離を計算する．しかし，*colour—color*, *oestrogen—estrogen* の

[*1] 編集距離 [Levenshtein 66] では，文字の挿入・削除・置換による操作で編集を行うが，これらはすべて置換操作で表現できる．

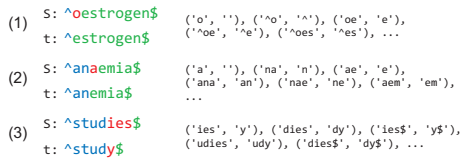

図 1: 置換ルールの生成例

ように，1 文字の差（‘u’ と ‘o’）で綴り変化を解釈できることもあれば，*Finland—inland*, *preservation—reservation* のように，1 文字の差（‘F’ と ‘p’）が別の意味の単語を生成することもある．したがって，置換する部分文字列や，適用できる周辺環境などを考慮し，単語の距離を計算することが望ましい．

さて，単語 w_s =’^anaemia$’ を，代表形 w_t =’^anemia$’ に変換する[*2]過程を考察しよう[*3]．2 つの文字列の差分から，w_s の 4 番目の文字’a’ を空文字列’’ に置換することで，w_t を得ることができる．部分文字列 f を e に置換するルールを，(f, e) と記述することにすると，このルールは (’a’, ’’) と書ける．しかし，この置換操作を w_s の 2 番目，8 番目の文字’a’ に適用してしまうと，w_t が得られない．これは，文字’a’ を消去する操作が，常に適用可能ではないからである．そこで，置換ルールに冗長性を持たせ，(’ae’, ’e’) というルールを考えると，w_s から w_t への変換が，より的確に説明できる．このように，w_s から w_e へ変換を説明できる置換ルールは複数あるが，ルールの正確さ，汎用性が異なる．

本研究では，単語を代表形に変換する際の特徴量として，一方の単語を別の単語に変更するときに要する文字列置換ルールに着目する．本節の残りでは，ある単語 w_s を別の単語 w_t に変更する置換ルールを列挙するアルゴリズムを述べる．2.2 節では，2 つの単語 w_s と w_t が与えられた時，単語 w_s を w_t に置換するルールから，w_t が w_s の代表形であるか識別するモデルを説明し，機械学習に基づいて置換ルールのスコア付けを行う．そして，2.3 節で，与えられた単語 w_s に対して，可能な代表形候補を生成する手法を説明する．

単語 w_s と w_t が共通に持つ最長の接頭文字列を l，最長の接尾文字列を r とする．単語 w_s と w_t の中で，接頭文字列 l，及び接尾文字列 r に含まれない箇所を，それぞれ c_s と c_t で表す．すなわち，単語 w_s と w_t は，それぞれ lc_sr と lc_tr に分解される．単語 w_s を w_t に置換する最短の操作（最短置換ルール）は，(c_s, c_t) である．さらに，最短置換ルールの左側に l の接尾辞，右側に r の接頭辞を追加した置換ルール（拡張置換ルール）も，やはり単語 w_s を w_t に変換できる．図 1 に，変換元と変換先の単語，及びその置換ルールの生成例を示した．単語中の青色と緑色の部分文字列は，それぞれ最長共通接頭文字列，最長共通接尾文字列を表す．例えば，図 1 (2) において，l =’^an’，r =’emia$’，$c_s$ =’a’，c_t =’’ であるから，最短置換ルール (’a’, ’’) を得る．l の接尾辞’n’ を最短置換ルールの先頭に追加すると，拡張置換ルール (’na’, ’n’) を，r の接頭辞’e’ を最短置換ルールの末尾に追加すると，拡張置換ルール (’ae’, ’e’) を得る．

2.2 代表形の識別

ある単語 w_t が，単語 w_s の代表語であるかどうか判別する問題は，与えられた単語ペア $\langle w_s, w_t \rangle$ に対して，w_t が w_s の代表語である（1）か，代表語ではない（0）のラベル $y \in \{0, 1\}$ を割り当てる二値分類問題である．本研究では，与えられた単語ペア $\langle w_s, w_t \rangle$ に対し，ラベル y の条件付き確率 $P(y|w_s, w_t)$ を，ロジスティック回帰モデルで表現する（式 1, 2）．

$$P(1|w_s, w_t) = g\left(\boldsymbol{\lambda}^T \boldsymbol{F}(w_s, w_t)\right), \tag{1}$$

$$P(0|w_s, w_t) = 1 - g\left(\boldsymbol{\lambda}^T \boldsymbol{F}(w_s, w_t)\right). \tag{2}$$

ただし，$\boldsymbol{F} = \{f_1, ..., f_K\}$ は，0 または 1 を返す素性関数群 $f_k(w_s, w_t)$ の値をまとめてベクトルで表現したもの，K は素性の総数，$\boldsymbol{\lambda}$ は素性関数の重み，$g(z)$ はシグモイド関数，

$$g(z) = \frac{1}{1 + \exp(-z)}, \tag{3}$$

である．本研究では，単語 w_t が単語 w_s の代表語であるかどうか調べる素性として，2.1 節で説明した文字列置換ルールを用いる．置換ルールの全体集合を $R = \{r_1, ..., r_K\}$ とし，それぞれの置換ルール r_k を，次式で素性関数 f_k に対応付ける．

$$f_k(w_s, w_t) = \begin{cases} 1 & (w_s \text{ がルール } r_k \text{ で } w_t \text{ に変換される}) \\ 0 & (\text{それ以外}) \end{cases} \tag{4}$$

つまり，$\boldsymbol{F}(w_s, w_t)$ は，w_s を w_t に変換できる置換ルールに対応する要素を 1，それ以外の要素を 0 で埋めたベクトルを返す関数と解釈してもよい．素性の重み $\boldsymbol{\lambda}$ が既知のとき，単語ペア $\langle w_s, w_t \rangle$ に対するラベルの予測 $\hat{y}$ は，式 5 で得られる．

$$\hat{\boldsymbol{y}} = \underset{\boldsymbol{y} \in \{0,1\}}{\operatorname{argmax}} P(y|w_s, w_t) = \begin{cases} 1 & \left(\boldsymbol{\lambda}^T \boldsymbol{F}(w_s, w_t) > 0\right) \\ 0 & (\text{それ以外}) \end{cases} \tag{5}$$

ロジスティック回帰モデルの学習は，N 件の学習インスタンス $\left((w_s^{(1)}, w_t^{(1)}, y^{(1)}), ..., (w_s^{(N)}, w_t^{(N)}, y^{(N)})\right)$ が与えられたとき，モデルの対数尤度，

$$\mathcal{L}_{\boldsymbol{\lambda}} = \sum_{i=1}^{N} \log P(y^{(i)}|w_s^{(i)}, w_t^{(i)}) \tag{6}$$

$$= \sum_{i=1}^{N} \left\{ y^{(i)} \log p^{(i)} + (1 - y^{(i)}) \log\left(1 - p^{(i)}\right) \right\}, \tag{7}$$

を勾配法などで最大化する（最尤推定）．ここで，$p^{(i)}$ は $P(1|w_s^{(i)}, w_t^{(i)})$ の省略記法である．対数尤度 $\mathcal{L}_{\boldsymbol{\lambda}}$ は微分可能であり，その勾配は式 8 で与えられる．

$$\frac{\partial \mathcal{L}_{\boldsymbol{\lambda}}}{\partial \lambda_k} = \sum_{i=1}^{N} \left(y^{(i)} - p^{(i)}\right) f_k(w_s, w_t) \tag{8}$$

最尤推定は過学習を起こすことが知られており，通常はパラメータ $\boldsymbol{\lambda}$ に対して何らかの事前分布を仮定し，過学習を防止する．本研究では，素性関数が置換ルールで構成されていることから，モデルのパラメータを学習で推定することは，置換ルールの重み付けを行うことと等価である．また，2.1 節で説明した置換ルールの生成アルゴリズムは，単語を変形しうるルールをすべて列挙しているため，単語を代表形に変換するときに貢献できないルールや，汎用性が低く滅多に利用されないルールがたくさん含まれている．そこで，本研究では事前分布

*2 語尾変化に対しては単語の原形，綴り変化に対しては短い文字数，もしくはアルファベット順で若い単語を代表形とする．

*3 本研究で扱うすべての単語の先頭と末尾には，それぞれ記号^と$を追加することとする．

```
# ^anaemia$ ^anemia$
1 ('a','’'), ('ae','e'), ('na','n'), ('aem','em'),
  ('ana','an'), ('nae','ne'), ('^ana','^an'), ...
# ^dimerize$ ^dimerise$
1 ('z','s'), ('iz','is'), ('ze','se'), ('ize,'ise'),
  ('riz','ris'), ('ze$','se$'), ('eriz','eris'), ...
# ^spurt$ ^sport$
0 ('u','o'), ('pu','po'), ('ur','or'), ('spu','spo'),
  ('pur','por'), ('urt','ort'), ('^spu','^spo'), ...
```

図 2: 綴り変化コーパスの学習データ例

としてラプラス分布を仮定し，対数尤度関数を重みベクトル $\boldsymbol{\lambda}$ の L_1 ノルムで正則化する．学習時に最小化する損失関数は，

$$E_{\boldsymbol{\lambda}} = -\mathcal{L}_{\boldsymbol{\lambda}} + \frac{|\boldsymbol{\lambda}|}{\sigma} \tag{9}$$

となる．σ は L_1 正則化の影響をコントロールするパラメータであり，値を小さくすると，重み 0 が割り当てられる素性，つまりモデルから削除される素性の数が増える．なお，式 9 の第 2 項は $\lambda_k = 0$ において微分不可能であるため，最小化には Orthant-Wise Limited-memory Quasi-Newton (OW-LQN) 法 [Andrew 07] を用いた[*4]．

2.3 単語異表記から代表形への変換

前節の学習で得られる重みベクトル $\boldsymbol{\lambda}$ は，単語の代表形を認識するタスクにおいて，それぞれの置換ルールの性能を反映したものになる．すなわち，ある置換ルールに正の重みが割り当てられれば，その置換ルールは語形変化を正しく捉えており，負の重みが割り当てられた置換ルールは，語形変化を間違って表現していることになる．表 3 に，実際の学習によって高い重みが割り当てられた置換ルールを示した．例えば，最も高い重みを持つルールは，単語が `uss` という部分文字列を含む場合，その箇所を `us` に置換することを薦めている．

従って，与えられた単語 w の代表語の候補集合 $C(w)$ を求めるタスクにも，学習で獲得した素性の重みが利用できる．

$$C(w) = \{r(w) | r \in R, \boldsymbol{\lambda}^T \boldsymbol{F}(w, r(w)) > 0\} \tag{10}$$

ただし，$r(w)$ は単語 w に対して置換ルール r を適用して得られる単語を表す．式 10 は，単語 w にすべての置換ルール $r \in R$ を適用し，2.2 節で獲得した識別モデルが $C(w)$ を単語 w の代表形と判別したものだけを集める．なお，式 10 では，学習に用いたすべての置換ルール $r \in R$ を適用して，w に対して可能な代表語を列挙しているが，実際には学習において正の重みが割り当てられたルール $R_+ = \{r_k | r_k \in R, \lambda_k > 0\}$ だけを適用すればよい．学習時に L_1 正則化の影響を大きくして，有効な素性の数を絞り込んでいる場合は，少ない計算量で $C(w)$ を求めることができる．

3. 実験

本稿で提案した識別モデルを学習するには，一方の語が別の語の代表語である事例（正例）と，一方の語が別の語の代表語ではない事例（負例）を大量に含む学習コーパスが必要である．本研究では，生命・医学分野を向けに構築された英語語彙リソースである UMLS SPECIALIST Lexicon[*5]を用い，綴り変化コーパス（正例 15,830 件，負例 33,296 件）と，語尾変化コーパス（正例 113,215 件，負例 124,747 件）を自動獲得した．綴り変化コーパスの正例として，LRSPL 辞書（spelling variants table）に収録されている単語ペア，語尾変化コーパスの正例として，LRAGR 辞書（agreement/inflection table）に収録されている単語ペアを採用した．各コーパスの負例は，LRWD 辞書（word table）に収録されている語彙のすべてのペアのうち，正例にならなかったものを用いた．ただし，ある事例の単語ペアから生成されるすべての素性（置換ルール）が，負例からのみ利用される場合は，その事例は学習の結果に依らず負例と判別されるので，学習データから削除した．図 2 に綴り変化コーパスと生成される置換ルールの例を示す．`#`で始まる行は w_s と w_t を，後続の行はその事例のラベルと置換ルール（の一部）を表している．

System	P	R	F1
提案手法 ($\sigma = 5$)	.899	.873	.886
ED ($\theta_e = 1$)	.319	.871	.467
ED ($\theta_e = 2$)	.323	.999	.488
NED ($\theta_n = 0.118$)	.440	.835	.576
ステミング [Porter 80]	.084	.074	.079

表 1: 綴り変化コーパスにおける評価結果

System	P	R	F1
提案手法 ($\sigma = 5$)	.979	.983	.981
ED ($\theta_e = 1$)	.484	.679	.565
ED ($\theta_e = 3$)	.479	.988	.646
NED ($\theta_n = 0.308$)	.495	.964	.654
ステミング [Porter 80]	.926	.839	.881

表 2: 語尾変化コーパスにおける評価結果

評価実験として，ある単語が別の単語の代表形であるかどうかを識別するタスクにおける提案手法の性能（適合率，再現率）を測定した．提案手法は学習に基づく手法であるため，実験では 10 分割交差検定を行い，学習データとテストデータが重ならないようにした．性能比較のためのベースライン手法として，編集距離（ED），正規化編集距離（NED），ステミング [Porter 80] を用いるシステムを用意した．編集距離では，与えられた 2 語の編集距離がある閾値 θ_e 以下である場合，その 2 語を単語–代表形の関係にあると見なす．正規化編集距離では，編集距離を長い方の語の文字数で割り，その値がある閾値 θ_n 以下である場合，単語–代表形の関係にあると判別する．ステミングでは，与えられた 2 語の両方に Porter のステマーを適用し，得られた語幹が同一であれば，元々の単語ペアが単語–代表形の関係にあったと見なす．

表 1 に，綴り変化コーパスにおける適合率 (P)，再現率 (R)，F1 スコア (F1) を示した．提案手法の F1 スコアは 0.886 であり，評価したシステムの中では最も良い性能を示した．編集距離は，閾値 $\theta_e = 2$ のときに F1 スコアが最大となるが，適合率が低く，綴り変化の関係にある語のペアと，似た綴りで意味の異なる語のペアを区別できなかった．正規化編集距離は，閾値 $\theta_n = 0.118$ のときの F1 スコアが最大となったが，正規化しない編集距離よりは若干良い性能を示す程度であった．ステミングは，綴り変化に対処するアルゴリズムではないため，このコーパスで性能の評価を行うのは不適切であるが，参考として結果を掲載した．

表 2 は，語尾変化コーパスにおける適合率 (P)，再現率 (R)，

∗4 本研究では OW-LQN 法の実装として，libLBFGS (`http://www.chokkan.org/software/liblbfgs/`) を利用した．

∗5 SPECIALIST: `http://specialist.nlm.nih.gov/`

順位	置換元	置換先	重み	適用例
1	uss	us	9.81	focussing
2	aev	ev	9.56	mediaeval
3	aen	en	9.53	ozaena
4	iae$	ae$	9.44	gadoviae
5	nni	ni	9.16	prorennin
6	nne	ne	8.84	connexus
7	our	or	8.54	colour
8	aea	ea	8.31	paean
9	aeu	eu	8.22	stomodaeum
10	ooll	ool	7.79	woollen

表 3: 高い重みが割り当てられた置換ルール（綴り変化）

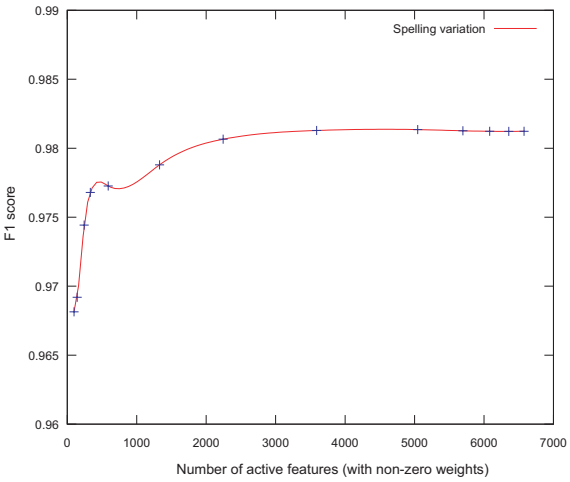

図 3: 非 0 の重みが割り当てられた素性数と F スコアの関係

F1 スコア (F1) である．こちらのコーパスでも，提案手法がすべての評価尺度において最も良い性能を示した．このコーパスでは，ステミングの真価が発揮されるはずであるが，再現率が低く，提案手法と比較しても見劣りする結果となった．ステミングが語尾変化コーパスにおいて失敗する事例を調べてみると，-ses で終わる語（例えば *analyses*）と，-sis で終わる語（例えば *analysis*）から，それぞれ別の語幹（例えば *analys* と *analysi*）を生成するケースが目立った．

表 3 に，綴り変化コーパスで学習後に，高い重みが割り当てられた素性 10 個を示した．このように，学習で得られたモデルから，綴り変化のプロセスを文字列の置換として直接的に解釈できるのが，提案手法の特徴である．図 3 に，語尾変化コーパスにおいて，L_1 正則化パラメータ σ を 0.01 から 100 まで変化させ，学習後に非 0 の重みが割り当てられた素性の数（横軸）と，F1 スコア（縦軸）の関係をプロットした．正則化パラメータ σ を大きくし，素性の数を増やしていくと，F1 スコアは上昇していくが，素性数 5,000（$\sigma = 5$）付近で，性能改善が頭打ちとなる．パラメータ $\sigma = 0.01$ としたときは，ほとんどの素性がモデルから削除され，有効な素性数が 97 まで減少したが，0.961 の F1 スコアを達成しており，単純な置換ルールだけで語尾変化の特徴がかなり捉えられると推察される．

4. 結論

本稿では，文字列置換ルールとロジスティック回帰モデルに基づいた用語バリエーション認識手法を提案した．今後は，本研究で得られた知見をもとに，与えられた語の代表形を生成するシステムを構築し，辞書引きなどの処理に活用したい．

謝辞

本研究は，科学技術振興調整費・重要課題解決型研究等の推進「日中・中日言語処理技術の開発研究」の支援によるものである．

参考文献

[Andrew 07] Andrew, G. and Gao, J.: Scalable training of L_1-regularized log-linear models, in *Proceedings of the 24th International Conference on Machine Learning (ICML 2007)*, pp. 33–40 (2007)

[Aramaki 08] Aramaki, E., Imai, T., Miyo, K., and Ohe, K.: Orthographic Disambiguation Incorporating Transliterated Probability, in *International Joint Conference on Natural Language Processing (IJCNLP2008)*, pp. 48–55 (2008)

[Bergsma 07] Bergsma, S. and Kondrak, G.: Alignment-Based Discriminative String Similarity, in *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pp. 656–663 (2007)

[Bilenko 03] Bilenko, M. and Mooney, R. J.: Adaptive duplicate detection using learnable string similarity measures, in *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 39–48 (2003)

[Brill 00] Brill, E. and Moore, R. C.: An improved error model for noisy channel spelling correction, in *ACL '00: Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pp. 286–293 (2000)

[Chen 07] Chen, Q., Li, M., and Zhou, M.: Improving Query Spelling Correction Using Web Search Results, in *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pp. 181–189 (2007)

[Daille 96] Daille, B., Habert, B., Jacquemin, C., and Royauté, J.: Empirical observation of term variations and principles for their description, *Terminology*, Vol. 3, No. 2, pp. 197–258 (1996)

[Jacquemin 99] Jacquemin, C.: Syntagmatic and paradigmatic representations of term variation, in *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pp. 341–348 (1999)

[Levenshtein 66] Levenshtein, V. I.: Binary codes capable of correcting deletions, insertions, and reversals, *Soviet Physics Doklady*, Vol. 10, No. 8, pp. 707–710 (1966)

[McCallum 05] McCallum, A., Bellare, K., and Pereira, F.: A Conditional Random Field for Discriminatively-trained Finite-state String Edit Distance, in *Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence (UAI 2005)*, pp. 388–395 (2005)

[Navarro 01] Navarro, G.: A guided tour to approximate string matching, *ACM Computing Surveys (CSUR)*, Vol. 33, No. 1, pp. 31–88 (2001)

[Porter 80] Porter, M. F.: An algorithm for suffix stripping, *Program*, Vol. 14, No. 3, pp. 130–137 (1980)

[Tsuruoka 08] Tsuruoka, Y., McNaught, J., and Ananiadou, S.: Normalizing biomedical terms by minimizing ambiguity and variability, *BMC Bioinformatics*, Vol. Suppl 3, No. 9, p. S2 (2008)

[Yeganova 04] Yeganova, L., Smith, L., and Wilbur, J.: Identification of related gene/protein names based on an HMM of name variations, *Computational Biology and Chemistry*, Vol. 28, No. 2, pp. 97–107 (2004)