Vol.2015-NL-221 No.7
Vol.2015-SLP-106 No.7
2015/5/25

形態素解析の動的な辞書拡張による チャットログからの人名表現抽出

平藤 燎[†1,a)]　牧田 光晴[†2,b)]

概要：インターネット上のコミュニティサービスやソーシャルメディアのチャットログのような崩れた話し言葉テキストでは，形態素解析や固有表現抽出のような基礎的な処理が十分な精度で行えない．本研究ではこれらの基礎的タスクのうち，特に人名表現の抽出を従来より精度よく行う手法を提案する．まず会話のセッション内からヒューリスティクスを複数用いて人名表現候補を列挙する．これらを形態素解析時に辞書へ追加してから解析を行い，実際にその人名表現候補が選ばれた場合に人名表現と判定する．この方法により，既存の形態素解析器や固有表現抽出器では難しいあだ名や難読人名なども抽出できるようになり，Fisher's Randomization Test を用いた統計検定で 1%水準で F-measure の改善が有意なことを確認した．

1. はじめに

近年，コミュニケーション手段として，インターネットのコミュニティサービスや SNS におけるチャット機能が広く使われるようになった．チャット機能を用いたコミュニケーションでは，特徴的な言語表現が使われる．例えば，「もぉしらないぃ」のような崩れた表記，特定のコミュニティでのみ用いられるスラング，新語や造語，あだ名等の表現がここに含まれる．このようなデータが日々アドホックに生成され，ログとして蓄積されている．

また，サービス品質向上等の目的で，チャットログを解析するニーズも増えている．例えば，対話システムの出力候補生成に会話ログ [*1] を用いるものがある．ここでは，ログ中に存在するユーザに関する情報を正しく識別することが求められる．この場合，固有表現抽出を適用することが一般的である．日本語を対象とする場合 [*2]，固有表現抽出のようなテキスト解析技術の性能は，前処理である形態素解析の性能に依存する．特に崩れた表現を多く含む言語データに関しては，既存の形態素解析器では十分な精度を実現できないことが知られている [1]．したがって，形態素解析処理における対象語彙のセグメンテーション精度向上が固有表現の抽出の性能向上にも寄与すると考えられる．

本研究では，上記課題に対処すべく，崩れた話し言葉を含むテキストからの人名表現抽出に焦点を絞る．ここでは，形態素解析処理内部において，会話ログから人名表現を精度よく識別する手法を提案する．対象となる表現には，「ちやちゃん」のようなあだ名や，「心愛(ここあ)さん」のような難読人名，「チミィ〜〜〜〜〜」のような崩れた人称代名詞など既存技術では対処が難しい表現が含まれる．

提案手法の有効性を確かめるため，実サービスの会話ログを用いた実験を行い，その改善結果の有意性を Fisher's Random Test を用いた統計検定法で検証する．

以下，2 章で提案手法について説明する．3 章で提案手法を用いた実験を行い，4 章では改善結果の有意性確認と考察を行う．5 章では関連研究と本研究を比較検討し，終章にて本稿のまとめを行う．

2. 提案手法

本研究では，形態素解析処理内部において，会話ログから人名表現を抽出する手法を提案する．形態素解析器の語彙を拡張するためには，コーパス等から抽出した語を辞書へ追加するのが一般的である．しかし，会話ログで頻出する平仮名のみの語彙等を一律に登録すると，かえって精度が大きく下がってしまう．そこで，本手法では辞書へ追加する語彙をそれぞれの会話内で得た候補のみに限ることで，この問題を避ける．

会話内から人名表現候補となる文字列を探索するために，種々のヒューリスティクスと事前に作成される文字言語モデルを用いる．

以下，詳細について説明する．

†1 現在，東京大学 理学部 地球惑星物理学科
†2 現在，株式会社サイバーエージェント
a) hirafuji_ryo_xa@cyberagent.co.jp
b) makita_mitsuharu@cyberagent.co.jp

*1 本稿では「チャットログ」「会話ログ」を同一概念を指す語として特に区別無く用いる．
*2 日本語以外の分かち書きの無い言語にも当てはまる．

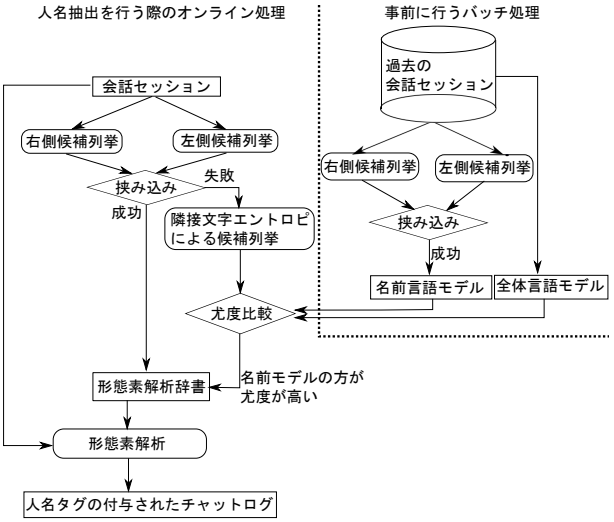

図 1 提案手法の処理概要

2.1 処理の概要

提案手法の処理概要を図 1 に示す.

処理は，事前に行うバッチ処理（図 1 右上）とログ解析時に行うオンライン処理 (図 1 左) の 2 つに分けられる.

以下，それぞれのステップについて説明を行う.

2.2 バッチ処理

バッチ処理では，人名語彙のみから構成される文字言語モデルと全ての語彙からなる文字言語モデルを作成する．そのために，会話ログ中から人名表現である可能性が高い文字列を抽出する．そして，抽出された人名文字列群から人名文字言語モデルを，人名以外の語彙も含んだ全体のログから全体文字言語モデルをそれぞれ作成する.

2.2.1 過去ログからの言語モデルの作成

過去の会話ログから名前を右端・左端に含む可能性が高い文字列を，後述するパターンマッチングを用いて列挙する．これらから両端を確定できた文字列を「名前」であると判断し，これらから「名前らしさ」の尤度を推定する「名前文字言語モデル」を作成する.

また，この名前言語モデルの比較対象として，過去のログ全体から「全体言語モデル」を別に作成する．言語モデルには Kneser-Kney スムージング [2] を行った文字バイグラムモデルを用いた.

2.2.2 会話ログ中からのパターンマッチングによる右側/左側候補の列挙

会話ログ中から人名表現を列挙する．そのために，まずは人名表現を片側に含む可能性が高い文字列を列挙する.

チャットでは，他のユーザーを呼びかける際に「＞ めぐ」のようなイディオムが使われる．また，現実と同じように「めぐ〜！」のような呼びかけも行われる．この事に着目し，会話セッションのうち以下のパターンにマッチする文字列を列挙する.

- ＞ で開始される部分
- 同一文字種（平仮名，片仮名，漢字，アルファベット）のみが出現する 7 文字以下の短い投稿

このパターンによって，「それは明日話そ？＞まさぽんのこと」や「まいってば ！」のような名前を左側に含む可能性が高い文字列を抽出できる.

さらに，一般に「さん」「ちゃん」のような接尾辞は名前の後に置かれる事が多いことを利用し，このような敬称で終わる文字列のパターンを抽出する（例：「ねねななおちゃん [*3]」「ららさんらら さん！ [*4]」.

これらのパターンを抽出することで，左端に人名を含んでいる可能性の高い文字列と，右端に人名を含んでいる可能性の高い文字列がそれぞれ得られる．これらの候補文字列を，以降それぞれ「左端候補」と「右端候補」と呼ぶ.

2.2.3 右端候補と左端候補からの挟み込み

前節で得た右端候補と左端候補を用いると，人名の両端を確定させることができる．例えば,「かいとの」という左端候補だけでは「カイ との」で分割され「カイ」が人名なのか「カイト の」で「カイト」が人名なのか分からない．しかし，ここで「あの かい くん！」という右端候補も得られれば，これらを組み合わせて「カイ」が名前だと分かる.

このように左端候補と右端候補の共通文字列を用いて「挟み込む」ことができた文字列は，人名である可能性が非常に高い．そのため，これらを用いて「名前文字言語モデル」を作成することができる.

2.3 オンライン処理

図 1 左の「オンライン処理」で示した形態素解析実行フェーズでは，まずバッチ処理と同様「挟み込み」処理によって人名表現である可能性が高い文字列を得る.

しかし，短い会話セッションの場合には十分な数の右端・左端候補を得られず，挟み込みだけでは十分な数の人名表現を得ることができない．そのため，片側の候補しか得られなかった場合は後述するステップも合わせて用い，より多くの人名候補を得る.

最後に，獲得した人名表現候補を形態素解析処理時に辞書に追加し，解析した結果追加した候補が選ばれた時に「人名である」と判定する.

2.3.1 隣接文字エントロピーを用いた片側文字列のみからの人名候補列挙

前節で得られた右側候補と左側候補のうち，片側しか得られなかったものだけを用いて人名候補を探索する．簡単のため，左側候補（名前から始まる可能性が高い文字列）を用いて説明するが，右側候補についても文字列を反転すれば同じ説明が当てはまる.

まず，$w_{1..k}$ を二候補以上で現れる k 文字の共通接頭辞

*3 ねね，ナナオちゃん
*4 ララさん，ララさん！

として，これに対する隣接文字エントロピー $H(w_{1..k})$ を次で定義する．

$$H(w_{1..k}) = \sum_{w_{k+1}} \frac{c(w_{1..k}w_{k+1})}{c(w_{1..k})} log_2 \left[\frac{c(w_{1..k}w_{k+1})}{c(w_{1..k})} \right] \quad (1)$$

ここで $c(w_{1..k})$ は k 文字の接頭辞 $w_{1..k}$ から始まる名前の候補数とする．言い換えれば，隣接文字エントロピーは候補集合から最初の k 文字が同じものを選んだときの，次の文字の（経験）エントロピーである．また，w_{k+1} に関しては文字列の終端も特殊な文字の一種であると見なす．

この隣接文字エントロピーがある閾値をこえる２文字以上の全ての接頭辞 $w_{1..k}$ をさらに次節のフィルタに渡す．この時のエントロピーの閾値は予備実験から 1bit とした．

2.3.2 人名候補の文字言語モデルによるフィルタリング

前節で得た人名候補にはまだ人名以外の語が多数含まれている．例えば，「それ」「それな〜」「それだ！」という３つの左端候補があった場合，この中で隣接文字エントロピーが閾値を超える共通接頭辞である「それ」が（本来は普通代名詞だが）人名候補として抽出される．このような無関係な文字列を除いて適合率を上げるために，バッチ処理で作成した文字言語モデルを用いて人名としての尤度が低い候補を除外する．

具体的には，前節の隣接文字エントロピーを用いて得た候補について人名言語モデル・全体言語モデルの両方で尤度を計算し，人名言語モデルよりも全体言語モデルの方が尤度が高かった場合は人名ではないと判断し候補から除外する．

2.3.3 人名候補の形態素解析ラティスへの追加

前段までに得た，挟み込みに成功した候補文字列と，隣接文字エントロピーと言語モデルで絞り込んだ候補文字列を，形態素解析時にラティス構造へ人名として追加する．この後ビタビアルゴリズムを用いて最短コスト経路探索を行い，この時に追加した人名候補を通るパスが実際に選択された場合に，その部分を名前として決定する．これによって，文章内での局所的な文脈を考慮した上で人名かどうかを判定できる．例えば，「なぁちゃん」という名前が列挙され辞書に追加されていた場合，「なぁと明日会える？」の「なぁ」は人名が入っていても不自然でない位置なので「なぁ/と明日会える？」のように区切られ「なぁ」が人名であると判定されるが，「それはちょっとどうかなぁと思う」の中の「なぁ」は人名が入るには不自然な位置なので人名として判定されることはない．

3. 実験

提案手法の効果を測定するため，株式会社サイバーエージェントの提供するコミュニティサービス「アメーバピグ [*5]」の会話ログ用いて人名表現の抽出性能を評価した．比較する既存手法には提案手法と同様に新たに教師データを作成する必要がない３手法を用いた．

[*5] https://pigg.ameba.jp/

3.1 ベースライン

ベースライン手法として固有表現抽出器である CaboCha[*6]ver0.69 と KNP[*7]ver4.12，さらに形態素解析器である MeCab[*8]ver0.996 を用い，辞書には ipadic[*9]ver2.7.0 を用いた．CaboCha，KNP，MeCab はそれぞれ教師ありによる識別器を用いて抽出を行うが，学習済みのパラメータセットが同時に配布されており，教師データを別途用意せずとも固有表現抽出や形態素解析を行うことができる．

固有表現抽出器では人名タグが付与されたものをそのまま用い，形態素解析器では IPA 品詞体系で「名詞 - 固有名詞 - 人名」以下として出力された形態素を人名とした．

3.2 実験に用いたプログラムと設定

提案手法の実装にはベースラインでも用いた MeCab ver0.996 を改変したものを使用し，辞書も同様に ipadic ver2.7.0 を用いた．人名候補の追加時には品詞を「名詞, 固有名詞, 人名, 名」として追加し，スコアは予備実験によって得た結果から 7000 とした．ipadic に含まれる同品詞の単語群は 7000 から 8000 程度のものが多く，それらの中では比較的低いスコアであるといえる．実験設定として，ipadic に含まれる人名はすべて除外し今回の手法で列挙した人名候補のみを用いた設定（提案手法 1）と ipadic に含まれる人名も候補として使った設定（提案手法 2）で評価を行った．

3.3 評価方法

上記の会話ログから 6000 発言をランダムに選び，ベースライン手法と提案手法を用いて人名部分を抽出した．抽出結果について予め人手で作成した正解と比較し，以下の尺度で評価した．

$$\text{再現率} \equiv \frac{\text{抽出器が正しく抽出した人名の数}}{\text{正解データ中のすべての人名数}}$$

$$\text{適合率} \equiv \frac{\text{抽出器が正しく抽出した人名の数}}{\text{抽出器が人名と判定した総数}}$$

$$\text{F-measure} \equiv \frac{2 \cdot \text{再現率} \cdot \text{適合率}}{\text{再現率} + \text{適合率}}$$

「ななやん [*10]」「なおちゃ [*11]」などのように敬称接尾辞もあだ名の一部を成していると考えて差し支えないケースがある．このため，正解を判定する際，「ななやん」の場合は「なな」のように人名と考えられる最小の部分を正解と

[*6] http://taku910.github.io/cabocha/
[*7] http://nlp.ist.i.kyoto-u.ac.jp/
[*8] http://taku910.github.io/mecab/
[*9] http://sourceforge.jp/projects/ipadic/
[*10] なな＋やん
[*11] なお＋ちゃ　なお＋ちゃんが幼児語的に訛った用法

表 1 実サービスの会話ログを用いた実験結果

設定	再現率	適合率	F-measure
MeCab	0.188	0.362	0.247
KaboCha	0.166	0.587	0.259
KNP	0.117	0.356	0.176
提案手法 1(既存人名辞書なし)	0.502	0.287	0.365
提案手法 2(既存人名辞書あり)	0.565	0.266	0.362

表 2 既存手法と比較した際の p 値

	提案 1	提案 2
MeCab	6.00×10^{-4}	1.00×10^{-5}
KaboCha	3.60×10^{-3}	2.50×10^{-3}
KNP	1.00×10^{-5}	1.00×10^{-5}

して与え，抽出器の答えがその部分を含んでいれば正解であるとした．

3.4 結果

実験によって得られた結果を表 1 に示す．提案手法では再現率が倍以上に改善し，適合率は下がったものの，F-measure ではすべての既存手法に比べ 0.1 以上改善している．また，ipadic の人名項目の有無については，人名項目を使わない場合に比べ使った場合の再現率が高かったが，逆に適合率は下がり F-measure としては微減となった．

4. 統計学的検定と考察

本章では，まず 4.1 で実験における F-measure の改善の有意性に関する検定について述べ，続く 4.2 で実験結果に対する考察を行う．

4.1 実験結果の検定

性能評価はあくまでランダムにサンプリングされた実験データを用いて行っているため，測定された F-Measure やその他の指標には必ずゆらぎが存在する．そこで統計学的検定により，結果における 2 群の差が偶然によるものか否かについて定量的に示す．ここでは先行研究 [3] でも使われている Fisher's Random Test[4][5] を実施し，「結果の差が偶然観測される確率」である p 値によって F-measure の増加を評価した．

今回の実験結果を片側検定した結果を表 2 に示す．列を比較元（ベースライン）として行を比較先とした時の p 値を示している．今回の提案手法は MeCab/KaboCha/KNP に対しては 1%以下の水準で有意に改善していることが分かる．なお，既存手法 2（ipadic 人名あり）に比べ既存手法 1（ipadic 人名なし）の方が F-measure がわずかに高いが，この 2 群に対して同様に検定を行うと p 値は 0.36 となり帰無仮説は棄却されない．つまり，ipadic 人名語彙を追加しても性能が低下するとは言えない結果となった．

4.2 考察

まず，提案手法で正しく抽出できた例を表 3 に示す．既存の手法では抽出できなかったり誤って一部のみを抽出していた場合でも，提案手法では正しく抽出できたことが分かる．また，完全に正解ではないが，正解より長い部分文字列を人名と判定したケースはあった (表 4)．本研究が想定する，対話システムの返答候補生成等での利用を考慮した場合，一部のみ抽出する既存手法より有用性が高いと考えられる．

次に，既存手法では誤抽出され，提案手法では誤抽出されない例を表 5 に示す．既存手法では形態素解析時の辞書の品詞エントリに影響を受け「バレインタイン」「笑」を人名として判定してしまうが，提案手法では会話内の他の発言でパターンにマッチした場合にのみ人名候補とするため人名と判定しない．

また，提案手法でも抽出できなかった例を表 6 に示す．これらの人名表現は敬称がない状態でのみ使われており，単体での呼びかけも行われなかったためパターンにマッチせず，候補とすることが出来なかった．

しかし「りつ だいすき」や「しぇる ｓん[*12] とあったけど」のような手がかりとなりうる発言は他に存在しており，これらの情報が利用できれば抽出可能になると考えられる．例えば「だいすき へ続く文字列は人名である可能性が高い」や「ｓん の左端は人名である可能性が高い」というような「敬称ほど確信度は高くないが人名抽出の手がかりとなる可能性が高い」複数のルールで何度もマッチした文字列を人名候補として補足的に用いる等の方法が考えられる．

これらのルール構築には，漸近的な知識獲得を行う Espresso[6] やとくに非わかち書き文を対象とした g-Monaka[7] などのアルゴリズムを用いる事ができると考えられる．自動獲得されたルールが十分に存在すれば，片側パターンしか得られなかった時のフォールバックである隣接文字エントロピーと言語モデルへの依存が低減することも期待できる．

最後に，提案手法で誤って人名と判断された例を表 7 に示す．右端候補抽出に用いた敬称パターンは比較的カバレッジが高く実際に名前で終わっている可能性が高い．これに対して，左端候補に用いた「＞人名」のパターンや単発言での呼びかけは，カバレッジは高いものの実際には人名ではないものが多く含まれ，これらが疑陽性を引き起こし適合度を下げている一因となっている．

例えば，「＞」のパターンは人名だけではなく「昨日見たよ＞でんしゃ」のように返答時の話題の限定に行われることもある．こういったパターンを人名候補として扱ってしまうことにより「あの でんしゃ 乗りたい」のような入力

[*12] "さん"のタイプミス

表 3 正解語彙を提案手法で正しく抽出できた例

正解	提案手法による結果	形態素解析器	既存の固有表現抽出器
ももか のいえきて	ももか のいえきて	もも（桃） か（助詞）	(抽出しない)
心愛(ここあ) さんを一人にさせないでね	心愛(ここあ) さんを一人にさせないでね	心（名詞）愛さ（動詞）ん（助動詞	(抽出しない)
きぃー ちゃんこんにちはw	きぃー ちゃんこんにちはw	き（動詞）ぃ（名詞）ー（名詞）	きぃ ー ちゃんこんにちはw

表 4 正解語彙を含む文字列を提案手法で抽出できた例

正解	提案手法による結果	形態素解析器	既存の固有表現抽出器
また くらげ くん ねてないのかwww	まだくらげ くん ねてないのかwww	まだ（副詞） くらげ（名詞） くん（接尾）	(抽出しない)

表 5 既存手法での誤抽出が提案手法ではみられなかった例

正解	提案手法による結果	形態素解析器の結果	既存の固有表現抽出器の結果
バレインタインのちょこ よーいした？	(抽出しない)	バレインタイン（人名）	バレインタイン のちょこ よーいした？
そんなことないよ（笑）	(抽出しない)	笑（人名：えみ）	そんなことないよ（笑）

表 6 提案手法でも既存の手法でも抽出できない例

正解	提案手法による結果	形態素解析器の結果	既存の固有表現抽出器の結果
もう しぇる とつきあうしかないな	(抽出しない)	もうし（動詞）＋未知語	(抽出しない)
りつ は悪くないよ	(抽出しない)	りつ（一般名詞）	(抽出しない)
み〜しゃ ちゃのこと	(抽出しない)	み（接頭）〜（記号）し（動詞） ＋未知語	(抽出しない)

表 7 提案手法で誤って人名と判断された例

入力	提案手法による結果	形態素解析器	既存の固有表現抽出器
あのでんしゃ乗りたい	あの でんしゃ 乗りたい	でん（副詞）しゃ（動詞）	(抽出しない)
ちょ，もうやめろってw	ちょ，もうやめろってw	ちょ（名詞, 動詞非自立的）	(抽出しない)

に対して人名と誤判定してしまう．また，単発言での呼びかけには人名だけでなく「ちょw」「おは」「うん」など短い頷きや同意，挨拶なども含まれる．これらを一律に人名候補として扱ってしまうことで，疑陽性を引き起こすと考えられる．

前述のようにパターンを自動で複数導出し，複数回マッチしたもののみを用いるアプローチはこの擬陽性にも有効であると考えられる．

5. 関連研究

固有表現抽出の手法として，従来から CRF[8][9][10] や SVM[11] などの手法が英語や日本語，中国語，タイ語など種々の言語で提案されている．これらは識別モデルであるため，識別のための素性を設計した上で教師データを用意して学習を行う．用いる素性として，福島ら [8] は前後の文字種，単語，品詞を，山田ら [11] は前後の形態素，形態素の品詞，文字種を用いており，同じ分類器を用いる手法においても異なる素性が選択されている．また，高瀬ら [12] はアニメ関連用語を抽出するためには従来の素性に加えて文字数も素性として用いることが有効であるとしており，対象のドメインによっても有効な素性は異なる事が示唆される．

実装が公開され一般に用いられる固有表現抽出器である CaboCha[13] や KNP[14] には調整済みのパラメータセットが同梱されており，実用する際にはまずは同梱されたパラメータセットを用いる事が一般的である．しかし本研究の会話ログのような崩れた話し言葉を多く含む場合，十分な性能が得られない．

この問題に対し，CaboCha では自ら用意した学習データを用いて，素性に関しても自由に設計した上で再学習を行うことができるが，作成コストが掛かる上に得られる性能向上の予測は難しい．KNP ではパラメータの調整に非常に大規模な Web コーパスを用いており，会話ログだけで同等の規模のコーパスを用意することは現実的ではない．

一方，既存の形態素解析器で対象メディアに特有な単語等を追加したユーザ辞書を整備することは一般的である．それに加え，近年，形態論的制約から未知語をオンラインで獲得する手法 [15] や，正規化を行うことで「みたぁぁぁぁい（見たい）」のような崩れた表記に対しても正しく形態素解析を行う手法 [1] のように，動的に形態素解析を行う手法が提案されている．本研究で対象とした人名表現には，比較的語彙数が限られるスラングとは異なり無限に語彙が

存在する．このため，手動で事前に辞書を整備するだけでは上手く対処できない．さらに，「なぁ（ちゃん）」「やや（ちゃん）」など他の機能語と重複した短い人名表現も多く，これらを辞書に一律に追加してしまうと副作用が発生する．したがって，提案した手法のようにオンラインで人名を一時的に増やすアプローチは，この副作用を避けつつ語彙を無数に獲得できるため有効であると考えられる．

6. 終わりに

本稿では崩れた話し言葉が多く含まれるチャットログから人名表現を抽出する手法を提案した．ここでは解析対象の会話から人名候補を抽出し，これを形態素解析処理中に辞書へ追加するアプローチを採った．

実サービスの会話ログを用いた実験の結果，提案手法は，既存の固有表現抽出器に比べ２倍以上の再現率で幅広い人名表現を抽出できた．F-measure も 1%水準で有意に改善したことを Fisher's Random Test で示した．

提案手法ではヒューリスティクスにより抽出パターンを作成した．今後はこれを自動獲得することで再現率の改善が期待できる．また，自動獲得によって得た十分な数のパターンを組み合わせることで，言語モデルを用いたフォールバックへの依存も軽減され，適合率も向上できると考えられる．

参考文献

[1] 笹野遼平，鍛治伸裕：不自然言語処理-枠に収まらない「リアルな」言語処理-：2. 新しい語・崩れた表記の処理，情報処理，Vol. 53, No. 3, pp. 211–216 (2012).
[2] Kneser, R. and Ney, H.: Improved backing-off for m-gram language modeling, *Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on*, Vol. 1, IEEE, pp. 181–184 (1995).
[3] Chinchor, N., Lewis, D. D. and Hirschman, L.: Evaluating message understanding systems: an analysis of the third message understanding conference (MUC-3), *Computational linguistics*, Vol. 19, No. 3, pp. 409–449 (1993).
[4] Fisher, R. a.: Design of Experiments (1935).
[5] Smucker, M. D., Allan, J. and Carterette, B.: A Comparison of Statistical Significance Tests for Information Retrieval Evaluation, *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management*, CIKM '07, New York, NY, USA, ACM, pp. 623–632 (2007).
[6] Pantel, P. and Pennacchiotti, M.: Espresso: Leveraging generic patterns for automatically harvesting semantic relations, *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, pp. 113–120 (2006).
[7] 萩原正人，小川泰弘，外山勝彦：グラフカーネルを用いた非分かち書き文からの漸次的語彙知識獲得，人工知能学会論文誌，Vol. 26, No. 3, pp. 440–450 (2011).
[8] 福島健一，鍛治伸裕，喜連川優：日本語固有表現抽出における超大規模ウェブテキストの利用，電子情報通信学会第 19 回データ工学ワークショップ/第 6 回日本データベース学会年次大会 (DEWS2008) (2008).
[9] Gao, J., Li, M., Wu, A. and Huang, C.-N.: Chinese word segmentation and named entity recognition: A pragmatic approach, *Computational Linguistics*, Vol. 31, No. 4, pp. 531–574 (2005).
[10] Tirasaroj, N. and Aroonmanakun, W.: Thai named entity recognition based on conditional random fields, *Natural Language Processing, 2009. SNLP '09. Eighth International Symposium on*, pp. 216–220 (2009).
[11] 山田寛康，工藤　拓，松本裕治：Support Vector Machine を用いた日本語固有表現抽出，情報処理学会論文誌，Vol. 43, No. 1, pp. 44–53 (2002).
[12] 高瀬真記，古宮嘉那子，小谷善行：CRF を用いたアニメ関連用語の固有表現抽出，CRF を用いたアニメ関連用語の固有表現抽出」第三回コーパス日本語学ワークショップ予稿集，pp. 179–182 (2013).
[13] 工藤　拓，松本裕治：チャンキングの段階適用による日本語係り受け解析，Vol. 43, No. 6, pp. 1834–1842 (2002).
[14] 笹野遼平，黒橋禎夫：大規模格フレームを用いた識別モデルに基づく日本語ゼロ照応解析，情報処理学会論文誌，Vol. 52, No. 12, pp. 3328–3337 (2011).
[15] 村脇有吾，黒橋禎夫：形態論的制約を用いたオンライン未知語獲得，自然言語処理 = Journal of natural language processing，Vol. 17, No. 1, pp. 55–75 (2010).