辞書とタグ無しコーパスを用いた固有表現抽出器の学習法

Learning Method of Named Entity Recognizer using Dictionary and Untagged Corpus

土田正明[*1*2] Masaaki TSUCHIDA　水口弘紀[*1] Hironori MIZUGUCHI　久寿居大[*1] Dai KUSUI　大和田勇人[*3] Hayato OHWADA

[*1]NEC 共通基盤ソフトウェア研究所
Common Platform Software Res. Labs., NEC Corp.

[*2]東京理科大学理工研究科経営工学専攻
Department of Industrial Administration, Graduate School of Science and Technology, Tokyo University of Science

[*3]東京理科大学理工学部経営工学科
Department of Industrial Administration, Faculty of Science and Technology, Tokyo University of Science

In this paper, we present a method to learn information extraction rules by dictionary and untagged-corpus for human-annotation-cost reduction. Here, dictionary consists of examples of extraction objective information and those classes information. Our method detects false positive (FP) and false negative (FN) from incomplete training data which contains some errors and missing class data to be generated by dictionary and untagged corpus, and learns information extraction rules using training data which FP and FN were rejected. In experiment with named entity task in Japanese, we had confirmed effectivity of our proposed method compared with baseline method not to reject FP and FN from incomplete training data by evaluation of macro-average f-measure.

1. はじめに

情報抽出は，テキスト情報の活用のため，特定目的の情報を抽出してデータベースなどに入力することで，テキスト情報を構造化する技術である．先行研究で，機械学習の適用によって，高精度な情報抽出が可能なことが示されてきた．例えば，[山田 02] は，日本語固有表現抽出にサポートベクターマシンを適用し，人手作成の抽出ルール [竹元 01] と同等精度を達成している．しかしながら，学習データ作成には高い人手コストを要する．なぜならば，学習データ作成には，多くの文書を読み，抽出する情報に漏れなく正確にアノテーションを付与する必要があるためである．これでは，抽出する種類の追加や抽出ルールの改良などの度に，高い人手コストがかかってしまう．

本稿では，抽出ルール作成の人手コスト削減を目的に，小規模辞書とコーパスから学習データを自動生成し，語句抽出ルールを学習する方法を提案する．また，固有表現抽出タスクで評価を行う．辞書には，抽出したいクラスとその語句の例が登録されている．このような辞書は小規模ならば低コストに作成できる．例えば，各クラスの想定される語句が，コーパス中に存在するか確認しながら登録すれば作成できる．仮に，想定できない場合でも，文書中から抽出すべき語句を見つける作業は，漏れなく正確にアノテーションを付与するより容易と考えられる．このように，辞書作成が低コストであれば，提案法によって抽出ルール作成の人手コストが削減可能と考えられる．

以降，第 2 節では関連研究と比較して本研究の特徴を説明する．第 3 節では提案法を説明する．第 4 節では，固有表現抽出タスクで，精度とコストの両面の評価を行い，結果と考察を述べる．第 5 節で総括し，本研究の今後の方向性を述べる．

2. 関連研究

関連研究には [Collins 99]，[Etzioni 04]，[Whitelaw 08] などが挙げられる．以下，本研究との違いを説明していく．最後に，半教師有り学習，能動学習との関係についても述べる．

連絡先: 土田正明，東京理科大学 理工学研究科 経営工学専攻，j7409701@ed.noda.tus.ac.jp

[Collins 99] は，英語の固有表現抽出で，各クラスのシードルールを用いて学習データを自動生成し，1）語句を分類するための決定リストを学習し，2）決定リストで分類できたデータを新たな学習データとして追加して 1 に戻る，を繰り返すブートストラップ学習法を提案している．ただし，大文字から始まる名詞を固有表現と考えてクラス分類しているため，英語の固有表現抽出に特化している．一方，本研究は，各クラスの正例，負例を各データの特徴量から自動生成するので，**特定の言語やタスクに依存しないフレームワークとなっている**．

[Etzioni 04] は，大規模文書から，抽出ルールを用いて各クラスの語の候補を抽出し，複数の抽出ルールと各候補との共起に基づく特徴量を用いたナイーブベイズ分類器で，各候補が正解か否か判定する．この方法では，複数の抽出パタンとの共起を特徴量に用いているため，各出現のみからの判定ができず，文脈で変化する意味のあいまい性を扱えない．一方，本研究は，各出現の特徴量に基づき正例，負例を自動生成して，抽出ルールを学習するため，**文脈依存の意味のあいまい性を扱える**．

[Whitelaw 08] は，固有表現辞書とコーパスから学習データを自動生成して，抽出ルールを学習している．各クラスの辞書をウェブ文書から自動増殖し，辞書の語の各出現を正例として学習している．負例（非固有表現）は，数字のみの語，1 語の名詞，などアドホックな基準で与えている．また，辞書の語の各出現をクラス分類するタスクとしているため，抽出対象が辞書の語に制限される．一方，本研究は，**アドホックな負例の基準を用いない**．また，**辞書に登録されていない語も抽出対象**である．

また，学習データ作成の低コスト化には，半教師有り学習，能動学習が関連するが，どちらも本研究とは問題設定が異なる．本研究では，1）教師データに間違いが含まれる，2）1 クラスが未知（辞書定義クラス以外の全てを表すクラス），の 2 点のため，半教師有り学習も能動学習も単純には適用できない．

3. 辞書とコーパスからの抽出ルール学習法

3.1 フレームワークと実現に向けた課題

提案法のフレームワークを図 1 に示す．本フレームワークはブートストラップ学習となっている．すなわち，提案法は，1）

辞書とコーパスから学習データ候補を生成する，2）学習データ候補から正しい学習データを選ぶことで学習データを自動生成する，3）教師有り学習によって抽出ルールを学習する，4）抽出ルールとコーパスから辞書を自動増殖して 1 に戻る，を繰り返すことで，抽出ルールを強化していく．3）には任意の教師有り学習法を適用可能である．本研究では，サポートベクターマシン (SVM) を用いている．

辞書とコーパスから学習データを自動生成するために，まず，1）によって辞書の語句の出現を正例候補とし，その他を負例候補として生成する．しかしながら，それら候補には，**偽の正例**，**偽の負例**が含まれてしまう．偽の正例 (FP:False Positive) は，辞書が正しい場合でも，語句の多義性に起因して発生する．例えば，辞書は「福島：地名」と正しいが，「福島」が人名として使われる場合である．偽の負例 (FN:False Negative) は，辞書に登録されていないが，コーパス中に存在する抽出すべきクラスの語句に起因して発生する．そのため，2）で学習データ候補から **FP,FN を検出し除去する**ことで，正しい学習データを自動生成する．また，4）で抽出ルールとコーパスから辞書を自動増殖するには，抽出ミスが含まれることを考慮して，**辞書に追加する正しい語句を選択**する必要がある．

以降，3）を除き，それぞれの具体的な方法を説明していく．

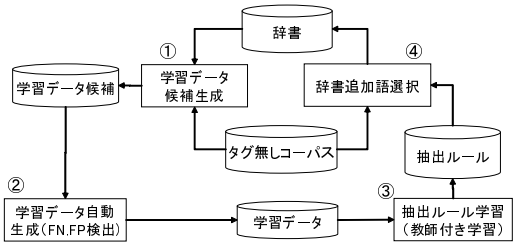

図 1　提案法のフレームワーク

3.2　学習データ候補生成

学習データ候補生成は，1) 図 2 のように，コーパス中の辞書の語句の出現にアノテーションを付与し，2) 次に，辞書アノテーション付きコーパス中の各語句の特徴量を抽出し $D_{cand} = \{D_{dict}, D_u\}$ を生成する．ここで，$D_{dict} = \{(\mathbf{x}_i, y_i)_{i=1}^{n}\}, D_u = \{(\mathbf{x}_j)_{j=1}^{m}\}$ である．$\mathbf{x}$ は特徴ベクトル，y はクラスラベルを表す．アノテーション付与部からラベル付きデータ D_{dict} が生成され，それ以外が D_u となる．

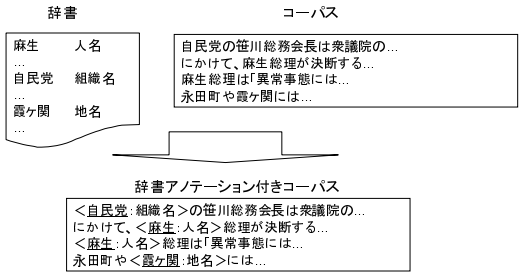

図 2　辞書によるアノテーション例

3.3　学習データ自動生成 (FP,FN の検出)

学習データ自動生成では，学習データ候補 $D_{cand} = \{D_{dict}, D_u\}$ を入力に 1)D_u から D_{dict} の数倍をランダムサンプリングして D_{smp} を生成し，2)$D_{cand2} = \{D_{dict}, D_{smp}\}$ から FN,FP を検出して各クラスの正例，全クラス共通の負例とすることで，最終的な D_{train} を生成する．

1) は，正例，負例の数のバランスを調整するために行う．提案法は，D_{dict} から各クラスの正例を，D_u から全クラス共通の負例を生成する．そのため D_u が D_{dict} に比べて多すぎると正例，負例の数がアンバランスになってしまう．アンバランスな学習データからの機械学習は困難である．そのため，D_u からサンプリングすることで，正例，負例のバランスを調整している．

2) では，正例，負例を生成するために FP,FN 検出を行う．**FP 検出**では，D_{dict} から，実際には y_i(辞書によるクラスラベル) ではない x_i(語句の出現) を検出し，残りをそのクラスの正例とする．**FN 検出**では，D_{smp} から辞書に定義されている各クラスの語句の出現を検出し，残りを全クラスの共通の負例とする．これにより，全データにラベルが付き，$D_{train} = \{(\mathbf{x}_i, y_i)_{i=1}^{N}\}$ が生成される．FP，FN 検出の基本的な考え方を以下に示す．

- **FP 検出**：自クラスよりも他クラスのデータに近い D_{dict} のデータは，自クラスの FP らしい
- **FN 検出**：D_{dict} のいずれかのクラスに近い D_{smp} のデータは，そのクラスの FN らしい

提案法では D_{cand2} をクラスタリングし，クラスタ内の D_{dict} のデータ数に基づき FP,FN を検出する．図 3 を用いて説明する．辞書には A，B，C のクラスの語句が定義されている．図 3 のように，クラス A が多く含まれるクラスタの他クラスのデータをクラス A の FN と見なす．また，クラス A が，他クラスのデータが多いクラスタに含まれている場合，それらデータをクラス A の FP と見なす．D_{smp} が多く含まれるクラスタ内の D_{smp} は，どのクラスにも類似しないので全クラスに共通の負例と見なす．ただし，クラスタ内の各クラスの D_{dict} や D_{smp} のデータ数の大小は，D_{cand2} 全体での各データの数で割って，相対化した上で比較する．理由は，D_{cand2} 内で数が多いデータほど各クラスタに多く含まれやすいため，数を手がかりにする本手法では，相対化しないと元々数が多いデータに支配されてしまうためである．

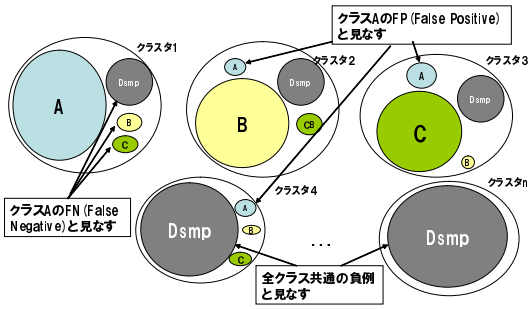

図 3　クラスタリング結果からの FP,FN 検出のイメージ．楕円の面積がデータ数を表している．

仮に，クラスタ内のラベルの分布が，データ全体のラベルの分布と同じ場合は，そのクラスタのデータは信頼できないので用いない．理由は，ラベルの分布がデータのランダム分割と同じなので，クラス毎にまとまる傾向を利用する本手法にとって信頼できないデータ群と考えられるためである．

また，提案法では，辞書のクラス情報を有効利用するために，クラスタリングに，制約付きクラスタリングと距離関数学習を組み合わせた MPCK-means[Billenko 04] を用いる．MPCK-means は，制約に must-link (同クラスタになるべきデータ対)，cannot-link(異クラスタになるべきデータ対) を与えると，制約を守るように距離関数を学習しながらクラスタリングを実行する．これら制約を D_{dict} のクラス情報から作成することで，距離関数学習を通して辞書のクラス情報の知識を活用できる．

3.4　辞書追加語選択

辞書追加語選択では，D_{train} から学習した抽出ルールとコーパスを用いて，辞書に追加する語を選択する．抽出ルールは，決定木やサポートベクターマシンなど，任意の教師有り学習法で

学習する．

まず，抽出ルールを用いてコーパスから 1 回以上抽出された語句をそのクラスの追加語候補とする．次に，各候補に対して，そのクラスの語句として追加すべき度合いを表すクラススコアを計算する．最後に，各クラスで，クラススコアの上位 n 語を辞書に追加する．クラススコア $Clscore(w,c)$ の計算式は，1）語句 w がクラス c の語句として正しい，2）語句 w がコーパス中で c 以外のクラスの語句として使われていない，の 2 点を満たすほど高くなるよう定める．具体的には以下の式で計算する．

$$ClScore(w,c) = ext\_cnt(w,c) \times \frac{\sum_{\mathbf{x}\in D(w)} f(\mathbf{x},c)}{|D(w)|}$$

$$f(\mathbf{x},c) = \frac{f_c(\mathbf{x})}{\sum_{c_i \in C} f_{c_i}(\mathbf{x}))}$$

$D(w)$ は, 語句 w の各出現の特徴ベクトルの集合, $ext\_cnt(w,c)$ は，語句 w がクラス c として抽出された数，$f_{c_i}(\mathbf{x})$ は，入力 $\mathbf{x}$ がクラス c_i である確信度を返す関数である．本手法の $f_{c_i}(\mathbf{x})$ では，SVM の出力をシグモイド関数で変換した値を用いる．

$ext\_cnt(w,c)$ は，語句 w がクラス c として抽出された回数なので，大きいほどそのクラスの語句として正しいと考えられる．また，$\frac{\sum_{\mathbf{x}\in D(w)} f(\mathbf{x},c)}{|D(w)|}$ は，平均的に各出現でクラス c のみが尤もらしい場合に高くなる．そのため，それらの積である $ClScore$ は，各クラスの語句として正しほど，かつ，他クラスの語句として使われる可能性が低いほど高くなる．

4. 評価実験

本節では 1）提案法とベースライン法の比較による精度評価, 2) 人手学習データ作成と比較したコスト評価，について述べる．

4.1 実験設定

実験データには，固有表現抽出の評価データである CRL_NE を用いた．CRL_NE は，毎日新聞 95 年 1 月 1 日から 10 日までの全 1174 記事，約 1 万文からなる．学習用に 1 日から 5 日までの記事，テスト用に 6 日から 10 日までの記事を用いた．固有表現は 8 クラス，組織名，人名，地名，人工物名，日付表現，時間表現，金額表現，割合表現，が定義されている．

本実験では，各文節を 9 クラス（8 クラスと非固有表現）に分類することで，文節単位の固有表現抽出を行った．それに伴って，オリジナルの文字単位ではなく，本実験では文節単位で正解を判定した．例えば，元の正解が「＜麻生：人名＞総理は」でも「＜麻生総理は：人名＞」など，文節単位で合えば正解とした．

言語解析には JAna[佐藤 03] を用いた．JAna は日本語テキストを入力に，形態素解析，文節処理，文節間の係り受け解析を行う．各形態素では，表層文字列，原型文字列，品詞情報が出力される．また，各文節では主辞とその意味属性が出力される．意味属性は，機械翻訳向けの情報で人，組織，時間，などが定義されている．ただし，意味属性は一般名詞にも振られているため，固有表現か否かを判断できる情報ではない．

辞書は，CRL_NE から固有表現を取得し，学習用コーパス内の頻度に比例するようサンプリングして自動生成した．これは 1）高頻度語ほど見つけやすい，2）高頻度語を登録することで辞書アノテーションの量が増える，の理由から，辞書への高頻度語の登録が自然であるためである．実験 1 では各クラス 10 語，実験 2 では各クラス 10，30，50，100 語の辞書を作成した．

SVM の多クラス分類への拡張には 1 vs rest 法を用いた．1 回の繰り返しで辞書に追加する語数は 3 とした．D_{smp} のサンプリング数は，予備実験の結果から D_{dict} の 3 倍[*1]とした．MPCK-means のクラスタ数は，定義された 8 クラスと非固有表現を考慮して 9 とした．MPCK-means の実装には，Weka-UT[*2]を用いた．また，MPCK-means での距離学習には，重み付きユーグリッド距離を用いた．それ以外のパラメタは Weka-UT のデフォルトを用いた．

特徴量には 1）抽出対象の文節，2）1 の直後の文節，3）1 の直前の文節，4）1 の係り先の文節，5）1 に係る文節，の 5 種の文節をそれぞれ区別し，各文節内の形態素情報と意味属性の出現の有無を用いた．各文節の主辞は他の形態素と区別した．

各実験では，8 クラスの適合率，再現率からマクロ平均を求め，2 つのマクロ平均から F 値を測定した．

4.2 実験１：精度面の有効性評価

提案法とベースライン法で 9 ターン学習を行い，各ターンの抽出精度を測定する．1 ターン目の抽出精度の比較から，提案の学習データ自動生成の有効性を評価する．また，2 ターン目以降の抽出精度の推移から，辞書追加語選択の有効性を評価する．

ベースライン法は，学習データ自動生成において FP,FN の検出による除去を行っていない点でのみ提案法と異なる．すなわち，辞書アノテーションからの D_{dict} を各クラスの正例，ラベル無しデータ D_u のランダムサンプルである D_{smp} を全クラス共通の負例 (非固有表現) として D_{train} を生成する．

ベースライン法の妥当性は，1）辞書アノテーションでもほぼ正しいので FP が無視できる，2）全体における固有表現の割合は少ないので，ランダムサンプリングでもほぼ非固有表現と考えられるため，FN が無視できる，という仮定に基づく．一方の提案法は，FP,FN を除去するため，無視できる程度に混入が少ないのならば，学習データを減る分悪影響の可能性もある．

結果と考察

実験結果を図 4，5 に示す．横軸がターン数で，縦軸が適合率，再現率，F 値のマクロ平均である．図 4，5 の 1 ターン目の F 値を比較すると，提案法が約 0.1 ポイント上回っていることから，提案の学習データ自動生成法の有効性が確認できた．また，繰り返し学習で，F 値が向上していることから，辞書追加後選択の有効性が確認できた．ただし，9 ターンでの F 値の向上は，ベースライン法で 0.08 ポイント，提案法で 0.03 ポイントと小さかった．

学習データ自動生成と辞書追加語選択の組み合わせの効果を見ると，ベースライン法のほうが F 値の向上が大きい．ただし，詳細を見るとベースライン法は，1 ターン目から時間表現，日付表現，割合表現，金額表現の 4 種類しか追加されていなかった．これは，ベースライン法では，辞書の語のみが抽出される適合率重視の厳しいルールが学習されやすいので，他の 4 クラスの新規語句が抽出できなかったためと考えられる．2 ターン目以降は，追加された 4 クラスの性能が向上し，他の 4 クラスの抽出精度が同じであるため，マクロ平均が向上する結果となった．

一方，提案法は「人工物」を除く全クラスの語句が追加されていた．ただし，途中で不適切な語が追加されたクラスで性能低下が起こったために，マクロ平均ではあまり向上しない結果となっていた．9 ターン全てで正しい語句が追加されていたクラスでは F 値が約 0.05〜0.08 ポイント向上していた．

4.3 実験２：コスト面の有効性評価

提案法で人手コストが削減できるか評価する．提案法で，各クラス 10，30，50，100 語の辞書を用いた場合の 1 ターン目と，同等 F 値の達成に必要な人手アノテーションの記事数を測定し，記事へのアノテーション付与と辞書作成時間を見積もって比較する．具体的には CRL_NE を用いて人手アノテーションの記事数を 20 記事づつ増やし，各辞書サイズの F 値を初めて超えた記事数と比較する．

---

*1 2 倍でも 4 倍でも F 値はほぼ変わらなかった．ただし，傾向として，大きくすると高適合率，低再現率となった．

*2 http://www.cs.utexas.edu/users/ml/risc/code/

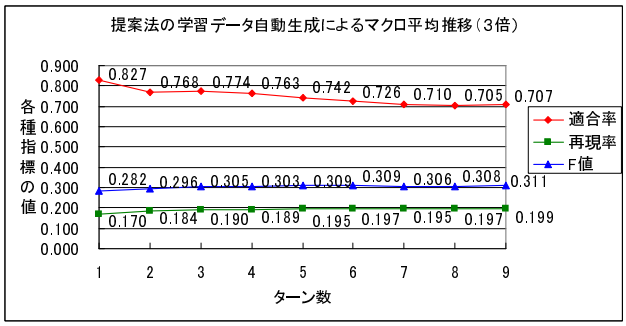

図 4　提案法

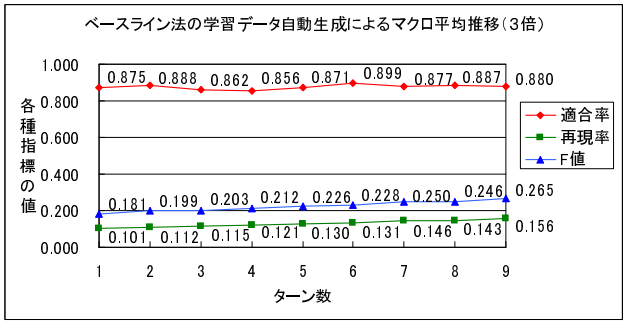

図 5　ベースライン法

結果と考察

実験結果を表 1 に示す．各 10 語の辞書では，40 記事の F 値が 0.217 で 60 記事が 0.300 であったため 50 記事と見なす．

記事アノテーションに必要な時間を見積もるため，筆者がタグ付けツール Tagrin[*3]を用いて 10 記事にアノテーション付与作業を行った．結果，1 記事平均 4 分であった．本作業では，判断に迷った部分は主観で決めたので，仕様を確認しながら厳密な作業を行うとさらに時間がかかると考えられる．そのため，ほぼ最小の見積もりと考えられる．上記のようにアノテーションの時間コストを見積もると，それぞれの辞書サイズと同等の F 値に達成するためには，各 10 語で 50 記事なので 200 分，各 30 語で 100 記事なので 400 分，各 50 語で 200 記事なので 800 分，各 100 語で 260 記事なので 1040 分，となる．

辞書作成に必要な時間は，1 語登録あたりの時間を t_w とすると，各クラスの語数$\times$ 8 $\times$ t_w 分と見積もれる．一般的な t_w の見積もりは困難だが，以下のように考えて見積もった．作業としては，事前に知っている語といくつかの文書中で見当たった語をリスト化し，ある程度たまったらコーパス中での高頻度語が含まれるかを確かめながら作成すれば良い．また，金額表現や割合表現など，機械的に語句を生成可能なクラスも存在する．すると t_w は平均 30 秒程度あれば十分と考えられる．$t_w = 30$ の場合の作成コストは，各 10 語で 40 分，各 30 語で 120 分，各 50 語で 200 分，各 100 語で 400 分となる．

比較すると，各辞書のサイズで 80 %，70 %，75 %，62 %の削減となる．また，小規模辞書を入力に提案法で辞書を自動増殖し，自動増殖された辞書を確認して辞書を作成することで，一層の人手コストの削減が可能となると考えられる．

上記は簡易的な見積もりで厳密性には欠けるが，コスト削減は十分に見込めると考えられる．ただし，人手アノテーションの記事数が増えるにつれて，同等の抽出性能を達成するために必要な辞書サイズの増え幅は，記事数のそれよりも大きくなる傾向にあることがわかった．そのため，抽出性能が収束し，実質的に同コストになるか，もしくは，人手アノテーションの記事数が増えると，辞書作成コストの方が高くなることも考えられる．今後，より詳細な評価が必要である．

*3　http://kagonma.org/tagrin/

表 1　各語数の辞書を用いた場合と同等の F 値に必要な記事数

辞書	適合率	再現率	F値	記事数	適合率	再現率	F値
各10語	0.798	0.145	0.246	60記事	0.707	0.190	0.300
各30語	0.686	0.255	0.371	100記事	0.805	0.243	0.373
各50語	0.687	0.281	0.398	200記事	0.743	0.275	0.401
各100語	0.658	0.306	0.418	260記事	0.797	0.284	0.418

5.　まとめと今後の方向性

本稿では，抽出ルール作成の人手コスト削減を目的に，辞書とコーパスから語句抽出ルールを学習する方法について述べた．提案法は，辞書とコーパスから生成した偽のデータを含む学習データ候補から，偽の正例 (語の多義性に起因するデータ) と偽の負例 (辞書の不完全性に起因するデータ) を除去して学習データを自動生成する．また，抽出ルールとコーパスを用いて辞書を自動増殖し，再度学習データを自動生成して抽出ルールを学習することを繰り返すことで，抽出ルールを強化していく．

固有表現抽出タスクでの評価実験では，辞書の語の各出現を正例，それ以外のランダムサンプルを負例とするベースライン法と比較し，F 値のマクロ平均で，約 0.1 ポイント向上したことから，提案法の有効性が確認できた．辞書自動増殖についても，9 ターンで F 値のマクロ平均が 0.04 ポイントと，向上幅は少ないが，効果を確認できた．向上幅が少ない理由に，辞書に不適切な語句が登録されたクラスでの抽出精度の低下があった．また，辞書作成時間と同等 F 値の達成に必要な記事アノテーションの作業時間を比較すると，提案法で人手作業時間を 60 %から 80 %程度削減できる可能性があることがわかった．

本研究は，今後の方向性として，1）**半教師有り学習，能動学習の適用**，2）**他の語句抽出タスクへの適用と評価**，を考えている．1）は，本手法で自動生成された学習データを教師有りデータ，それ以外を教師無しデータとすることで適用可能である．また，2）では，提案法の汎用性の評価と共に，提案法の得意，または不得意なタスクの性質を明らかにしていきたい．

参考文献

[Billenko 04] Billenko, M., Basu, S., and Mooney, J. R.: Integrating constraints and metric learning in semi-supervised clustering, in *Proc. of ICML'04*, pp. 839–846 (2004)

[Collins 99] Collins, M. and Singer, Y.: Unsupervised models for named entity classification, in *Proc. of the Joint SIGDAT Conference on EMNLP in Natural Language Processing and Very Larg Corporand*, pp. 100–110 (1999)

[Etzioni 04] Etzioni, O., Cafarella, M., Downey, D., Kok, S., Popescu, A. M., Shaked, T., Soderland, S., Weld, S. D., and Yates, A.: Web-scale information extraction in knowitall: (preliminary results), in *Proc. of WWW'04*, pp. 100–110 (2004)

[Whitelaw 08] Whitelaw, C., Kehlenbeck, A., Petrovic, N., and Ungar, L.: Web-scale named entity recgnition, in *Proc. of CIKM'08*, pp. 123–132 (2008)

[佐藤 03] 佐藤 研治, 池田 崇博, 中田 貴之, 長田 誠也：CRM 分野へ向けた日本語処理機能のミドルウェア化, 言語処理学会第 9 回年次大会発表論文集, pp. 109–112 (2003)

[山田 02] 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)

[竹元 01] 竹元 義美, 福島 俊一, 山田 洋志：辞書およびパターンマッチルールの増強と品質強化に基づく日本語固有表現抽出, 情報処理学会論文誌, Vol. 42, No. 6, pp. 1580–1591 (2001)