テキスト処理のための固有表現抽出ツール *NExT* の開発

桝井文人†, 鈴木伸哉†, 福本 淳一‡
三重大学 工学部 情報工学科†
{masui, suzuki}@shiino.info.mie-u.ac.jp†
立命館大学 理工学部 情報学科‡
fukumoto@cs.ritsumei.ac.jp‡

1 まえがき

情報検索や自動要約，質問応答など，多くの自然言語処理研究において，固有表現の取扱いの重要性が注目され，情報抽出 (IE) に関する研究も進んでいる [1, 5]．その中でも，固有表現抽出 (NE) は，情報検索や，自動要約，質問応答など，周辺の研究分野との関連性が指摘され，その重要性が認識されている [4, 7]．

こうした状況の中，各分野の研究者が効率的かつ有効に研究を進めるためには，情報抽出で培われた研究成果が，広く，誰もが利用できるツールとして提供されることも重要である．しかしながら，現時点において，そのような条件を満たした日本語固有表現抽出ツールはまだ公開されていない．

そこで，我々は，主に，自然言語処理分野における研究者を支援することを目的として，パターン照合に基づく固有表現抽出ツール "*NExT*(Named Entity extraction Tool)" を開発し，公開を開始した．本ツールは，汎用の形態素解析システム出力に対応し，様々な形式で出力可能な，柔軟なインターフェイスと，簡単な操作性，および，目的に応じて手軽に拡張や改変が可能な拡張性を持つ．

本文では，まず，本ツール開発における基本的な考え方と，その基本仕様について説明し，さらに，本ツールの特徴と基本性能，および入手方法などを紹介する．

2 開発における基本的な考え方

多くのユーザに，広く利用してもらうためには，それらのユーザにとって，使いやすいツールでなければならない．そのための条件として，(1) 柔軟な入出力インターフェイスを備えている，(2) 目的に応じて簡単にカスタマイズできる，(3) 安定した性能がある程度保証される，(4) 繁雑な設定や操作を必要とせず直観的に扱える，のような項目が挙げられる．

性能向上を主目的とした場合，機械学習や統計的手法は有効な手段である．しかしながら，上記 (1)〜(4) に挙げた条件を重視した場合には，これらの手法は，操作に関してある程度の専門知識や熟練が必要であったり，大量の学習データを準備する必要があるなどの理由から，必ずしも最適であるとはいえない．よって，本ツールでは，最も基本的な処理方法である，パターン照合に基づいた固有表現抽出処理を行う．具体的には，「〜大学」，「〜さん」，「国立〜」などの，固有表現に含まれる接頭辞・接尾辞・機能語など (以降「固有表現キー」と呼ぶ) の表記パターンを利用して，文書中に現れる固有表現を探索するものである．

また，システム開発の観点から考えた場合，ツールを公開し，多くのユーザに利用してもらうことによって，ユーザからの意見や要望，バグ報告などのフィードバックが期待できる．これらのフィードバック情報を分析することは，効率的かつ効果的なシステム精緻化につながると期待できる．

3 *NExT* の構成

本章では，*NExT* の基本的な仕様について概観する．*NExT* は，形態素解析処理済のデータを入力として受取り，固有表現キーを用いたパターン照合や形態素の並び，形態素品詞情報の並びなどを利用して処理を進めていく．

本ツールは，基本的には，図 1のような 5 つの処理部と，1 つの知識データによって構成される．以下，各処理部に関して説明する．

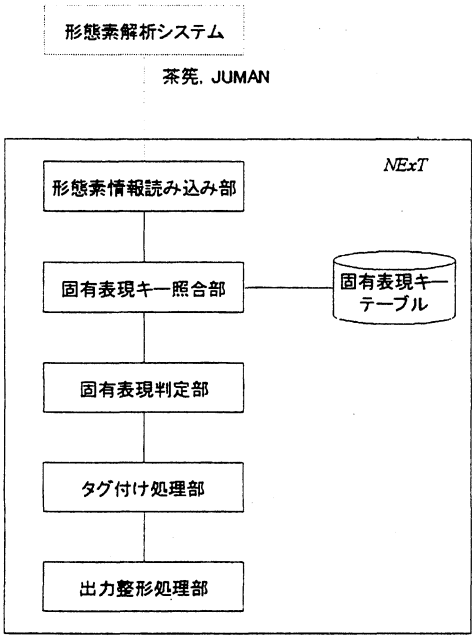

図 1: 固有表現抽出ツール *NExT* の構成図

形態素情報読み込み部では，形態素解析処理済のデータを読み込んで，抽出のための前処理を行う．現在のところ，対応済みの形態素解析プログラムは，茶筅 [8] および，JUMAN[6] の二種類である．読み込んだ情報のうち，処理に必要な情報 (見出し，原型，品詞，品詞細分類) を保持しておく．

固有表現キー照合部では，固有表現キーテーブルに保存されている，固有表現キーの情報を読み込む．さらに，パターン照合によって，形態素情報読み込み部で保持された形態素解析済データ中から，固有表現候補を探索する．このとき，探索範囲の上限は一文とし，左から右方向へ探索する．

固有表現判定部では，固有表現キーが発見された箇所を起点として，固有表現の先頭箇所を決定するためのバックトラック処理を行う．基本的には，固有表現キーの左側の形態素を調べ，固有表現の範囲を確定する．調べた形態素が，名詞や形容詞語幹，助動詞語幹などであれば接続させて範囲を拡張し，そうでない場合は接続せずに範囲を確定する．形態素の品詞情報が固有名詞である場合は，その品詞情報を採用する．また，キーに前置するカタカナの連続は接続，キー直後に記号対によって英文字列が現れた場合は短縮形とみなして抽出，など，文字種の連続性や短縮語の判定なども考慮している．確定した固有表現の属性は，固有表現キーが持つ属性によって決定される．固有表現キーが連続する場合は最も右のキーが持つ属性に変更する．

タグ付け処理部では，決定された固有表現の前後に，固有表現の属性を示すタグ情報を付与する．タグ情報は，コード化された，内部タグ形式である．形態素解析結果へのタグ埋め込みが指定されている場合は，形態素解析結果への復元処理を行う．

出力整形処理部では，タグが付与された固有表現に対する整形処理を行う．基本的には，内部タグ形式を，ユーザ指定に応じたタグ形式へ変換し，出力する．必要に応じて，固有表現が連続している場合の統合処理や，若干のヒューリスティック処理も行う．

固有表現キーテーブルには，接辞見出し，属性種別が記述される．属性種別は，コード化されて制御され，対応した処理規則によって細かなパターン照合制御に利用される．初期状態で設定されている属性種別は，“人名・地名・組織名・日付・時間・金額・割合” の 7 種である．

4 *NExT* の特徴

本章では，*NExT* の特徴について詳述する．

第一に，入力の柔軟性について述べる．*NExT* の入力情報には，形態素解析済のテキスト情報が必要である．したがって，本ツールを利用するためには，形態素解析システムの入手が必要である．

現在，本ツールは，形態素解析システム『茶筅』および『JUMAN』の出力形式 [6, 8] に対応している．具体的には，“chasen -c” のようなオプション制御によって得られる，品詞コード付きの出力結果が入力対象となる．両システムは，既に多くの研究者の間で利用されており，入手も簡単である．

第二に，出力の柔軟性があげられる．出力情報としては，茶筅，JUMAN，IREX タスク定義，MUC タスク定義，HTML，XML の，計 6 種類の出力が可能である．茶筅および JUMAN の出力形式については，各形態素解析結果に固有表現タグを埋め込んで出力する．これは，構文解析のように，本来の形態素解析出力形式を破壊せずに固有表現抽出の効果を利用したい場合に有効である．IREX[5]

およびMUC[1]のタスク定義出力は，両評価型ワークショップで提案された形式である．よって，これらの形式を用いることで，多くのシステムとの比較評価や，上記ワークショップに基づいた技術との連携が可能である．また，各タスク定義に対応した，スコアラーと呼ばれる評価プログラムも存在するので，これらが入手できれば，簡単に性能評価ができる．

第三に，使いやすさについて述べる．本ツールは，Perl 言語で実装されている．Perl は，Unix, Windows, Machintosh など，様々な OS 環境に移植された汎用的なプログラミング言語である．したがって，Perl 環境が設定されていれば，OS や機種の依存性を意識せずに利用可能である．また，本ツールはコマンドライン上での利用を前提として設計されている．入出力や対象ファイル指定などは，コマンド引数やオプション引数の形で制御される．したがって，ある程度コマンドプロンプト上でのド操作経験があれば，簡単に操作できる．

第四に，拡張性である．本ツールは、フリーソフトウェアとして，ソースコードを公開している．したがって，利用する固有表現キーの拡張や修正はもとより，ツール本体のカスタマイズを行うことも可能である．さらに，本ツールは，パターン照合の規則を基本としているため，複雑な計算過程は含まれていない．そのため，必要に応じて，比較的簡単に規則の修正追加が可能である．また，固有表現キーについては，接辞および属性種別を示す内部コード記述するだけでよい．キーの属性種別は，コード化されて制御されるため，予約コード番号の範囲内であれば，ユーザが自由に設定できる．また，キーパターン照合部の記述を拡張し，連携することによって，より細かなパターン照合の制御も可能である．

5 基本的な性能

NExT の基本性能を検証するために，IREX 評価で用いられた評価コーパス 100 記事 [5] を用いて，評価実験を行った．茶筅 [8] を用いて対象コーパスを形態素解析処理し，それに対して固有表現抽出処理を行い，正解コーパスとの比較を行った．ただし，*NExT* では，固有物名 (ARTIFACT) の抽出は行わなかった．評価には，IREX タスク定義 [5] に基づいて，適合率 (Recall)，再現率 (Precision) および F-measure を用いた．評価結果を表 5に示す．

表 1: *NExT* の性能評価結果 (茶筅出力利用)

	再現率	適合率	F-measure
組織名	55.66	56.95	56.30
人名	83.86	67.09	74.54
地名	67.40	71.00	69.15
日付	79.34	80.13	79.73
時間	88.57	70.45	78.48
金額	83.61	77.27	80.32
割合	87.18	97.14	91.89
合計	72.48	69.90	71.17

総合的には，再現率が 72.48%，適合率が 69.90%，F-measure が 71.17 であった．IREX ワークショップによれば，分野非限定タスクにおける全参加システムの F-measure 平均が 69.54，中間位システムの F-measure 平均が 70.34 である [5]．ことを考慮すると，本ツールは，IREX 評価基準において平均レベルの性能を示したといえる．

さらに，各表現の内訳をみると，組織名と地名の精度がやや低い．この原因として，「関西空港」や「三重大学」のように，組織名にも地名にもなり得る固有名詞に対して，うまく対応ができなかったことが考えられる．これらの固有名は，分野依存知識によって解決できるケースも多い．したがって，分野依存知識によるチューニングや，分野に応じた固有表現キー属性の切替えを行えば，精度は向上すると思われる．

6 おわりに

本論文では，自然言語処理分野や情報検索分野の，多くの研究者や技術者の研究開発を支援することを目的として開発した，パターン照合と品詞接続規則に基づく固有表現抽出ツール *NExT* について報告した．*NExT* が，拡張性や柔軟性を重視し，パターン照合に基づく処理を行うこと，複数の入出力形式に対応していることを述べた．

現在，*NExT* は，GNU General Public License version2(GPL2)[3] に基づいて，ベータ版が公開・配布されている．配布パッケージは，開発プロジェクトのホームページ (参考文献に URL を記載) [2] より入手可能である．配布パッケージには，抽出プログラム本体，抽出用固有表現キーテーブル，および，簡単なプログラム仕様と使用方法を記述したドキュメントが含まれている．ユーザに対して

[1] *http://www.itl.nist.gov/iaui/894.02/related_projects/muc/* を参照されたい．

は，ユーザメーリングリストを準備し，保守・管理に関する情報を提供するとともに，ユーザからのフィードバック情報を収集している．

今後は，固有表現キーの初期設定データの充実と，キー登録設定機能の強化，固有表現の種類の拡張，などをすすめていく予定である．

参考文献

[1] DARPA. *Proceedings of the Tipstar Text Program Phase III 18 month workshop.* DARPA, 1998.

[2] NExT 開発プロジェクト. Next - a named entity extraction tool. http://irmscher.shiino.info.mie-u.ac.jp/next/, 2001.

[3] GNU Project. *GNU's Not Unix!: License.* http://www.gnu.org/licenses/, 2001.

[4] 福本淳一, 関根聡, 江里口善生. MUC-7,Tipster 参加報告. 情報処理学会研究報告: FI51-14,NL127-14, pp. 101–108, 1998.

[5] 関根聡, 江里口善生. IREX-NE の結果と分析. 言語処理学会第6回年次大会ワークショップ論文集, pp. 25–32, 2000.

[6] 黒橋貞夫, 長尾眞. 日本語形態素解析システム JUMAN version 3.6 使用説明書. 1998.

[7] 桝井文人, 関根聡. TIPSTER Text Program Phase III 24th Month Workshop 参加報告. 電子情報通信学会技術報告: NLC98-57, pp. 98–32, 1999.

[8] 松本裕治, 北内啓, 山下達雄, 平野善隆, 松田寛, 浅原正幸. 形態素解析システム『茶筅』version 2.0 使用説明書 第二版. Naist technical report, 奈良先端科学技術大学院大学, 1999.

付録

例 1: IREX NE タスク定義に基づく出力例

```
<DATE>三　十　一　日</DATE><TIME>午　前　七時半</TIME>ごろ、<LOCATION>山梨県</LOCATION>の<LOCATION>甲斐駒ケ岳</LOCATION>八合目を登山中の<LOCATION>滋賀県草津市野路東</LOCATION>、演奏家、<PERSON>立命太郎</PERSON>さんと<LOCATION>三重県津市上浜町</LOCATION>、赤福店経営、<PERSON>三重次郎</PERSON>さんの二人が足を滑らせて沢に滑落、<PERSON>立命</PERSON>さんが行方不明になった。
```

例 2: MUC NE タスク定義に基づく出力例

```
<TIMEX TYPE="DATE">三十一日</TIMEX><TIMEX TYPE="TIME">午　前　七　時半</TIMEX>ごろ、<ENAMEX TYPE="LOCATION">山梨県</ENAMEX>の<ENAMEX TYPE="LOCATION">甲斐駒ケ岳</ENAMEX>八合目を登山中の<ENAMEX TYPE="LOCATION">滋賀県草津市野路東</ENAMEX>、演奏家、<ENAMEX TYPE="PERSON">立命太郎</ENAMEX>さんと<ENAMEX TYPE="LOCATION">三重県津市上浜町</ENAMEX>、赤福店経営、<ENAMEX TYPE="PERSON">三重次郎</ENAMEX>さんの二人が足を滑らせて沢に滑落、<ENAMEX TYPE="PERSON">立命</ENAMEX>さんが行方不明になった。
```

例 3: HTML 形式での出力例

```
<font color="#800000">
三十一日</font><font color="#800000">午前七時半</font>ごろ　、<font color="#ff0000">山梨県</font>の<font color="#ff0000">甲斐駒ケ岳</font>八合目を登山中の<font color="#ff0000">滋賀県草津市野路東</font>、演奏家、<font color="#008000">立命太郎</font>さんと<font color="#ff0000">三重県津市上浜町</font>、赤福店経営、<font color="#008000">三重次郎</font>さんの二人が足を滑らせて沢に滑落、<font color="#008000">立命</font>さんが行方不明になった。
```

例 4: XML 形式での出力例

```
<?xml version="1.0" encoding="EUC-JP" ?>
<!DOCTYPE tagged-document [
<!ELEMENT tagged-document (
    org,person,loc,num,date,time,money,percent)>
<!ELEMENT org (#PCDATA)>
<!ELEMENT person (#PCDATA)>
<!ELEMENT loc (#PCDATA)>
<!ELEMENT num (#PCDATA)>
<!ELEMENT date (#PCDATA)>
<!ELEMENT time (#PCDATA)>
<!ELEMENT money (#PCDATA)>
<!ELEMENT percent (#PCDATA)>
]>
<tagged-document>
<date>三十一日</date><time>午前七時半</time>ごろ、<loc>山梨県</loc>の<loc>甲斐駒ケ岳</loc>八合目を登山中の<loc>滋賀県草津市野路東</loc>、演奏家、<person>立命太郎</person>さんと<loc>三重県津市上浜町</loc>、赤福店経営、<person>三重次郎</person>さんの二人が足を滑らせて沢に滑落、<person>立命</person>さんが行方不明になった。
</tagged-document>
```

例 5: 茶筅形式での出力例 (IREX タグ埋め込み)

```
<DATE>
三十一 サンジュウイチ 三十一 19 0 0
日 ニチ 日 33 0 0
</DATE>
<TIME>
午前 ゴゼン 午前 16 0 0
七 ナナ 七 19 0 0
時半 ジハン 時半 33 0 0
</TIME>
ごろ ゴロ ごろ 32 0 0
、 、 、 76 0 0
<LOCATION>
山梨 ヤマナシ 山梨 11 0 0
県 ケン 県 2 0 0
</LOCATION>
の ノ の 68 0 0
<LOCATION>
甲斐 カイ 甲斐 2 0 0
駒ケ岳 コマガタケ 駒ケ岳 11 0 0
</LOCATION>
八 ハチ 八 19 0 0
合 ゴウ 合 33 0 0
目 メ 目 28 0 0
を ヲ を 59 0 0
登山 トザン 登山 17 0 0
中 チュウ 中 32 0 0
の ノ の 68 0 0
<LOCATION>
滋賀 シガ 滋賀 11 0 0
県 ケン 県 2 0 0
草津 クサツ 草津 11 0 0
市 シ 市 28 0 0
野路東 ノジヒガシ 野路東 11 0 0
</LOCATION>
、 、 、 79 0 0
演奏 エンソウ 演奏 17 0 0
家 カ 家 28 0 0
、 、 、 79 0 0
<PERSON>
立 タチ 立 11 0 0
命 イノチ 命 2 0 0
太郎 タロウ 太郎 8 0 0
</PERSON>
さん サン さん 29 0 0
と ト と 65 0 0
：　：　：　：
```