コーパスコンコーダンサ『ChaKi.NET』のラベリング機能

浅原 正幸
人間文化研究機構 国立国語研究所
言語資源研究系・コーパス開発センター

森田 敏生
総和技研

masayu-a@ninjal.ac.jp

1 はじめに

コーパスに対するアノテーションを実施する際に定形的なパターンに基づく規則によりラベルを付与することによりアノテーションの作業コストを削減することができる。この作業を規則に基づくコーパスラベリングと呼ぶ。コーパスアノテーションでは、MATTER サイクル [3] と呼ばれるアノテーションと機械学習を交互に行う作業モデルが提案されている。構文・意味を扱う構造が複雑になるにつれて、アノテーションの作業工数と機械学習の計算量の双方が負担となり、きれいにサイクルを回すことは困難である。しかしながら、言語処理の分野だけでなく、言語学に視野を広げると、規則で記述できる文法的なパターンが文献に記述されており、これらをあらかじめラベル付けしておくことで作業者の負担を軽減することができる。

[4] では形態素解析器 MeCab・係り受け解析器 CaboCha の解析結果を入力として、JSON ベースのクエリ言語で記述されたパターン（部分文字列・形態素列情報･文節係り受けに基づく部分木構造）に適合する形態素位置にラベルを付与するツールを開発し、鳥バンクで用いられている節認定規則を実際に記述し、節境界ラベリングツールを試作した。

実際のラベリングパターンの記述では対象テキストに対する精度・再現率のバランスを検証しながら記述することが多い。具体的には次のサイクルを回しながらパターンの洗練を行う：

1. パターンの記述 (Define)
2. テキスト中のパターン適合候補の枚挙 (Extract)
3. 適合候補の検証 (Evaluate)
4. パターンの制限・緩和 (Restrict or Alleviate)

このパターンの洗練サイクルは、コーパスコンコーダンサにおけるクエリの洗練とほぼ同様の作業である。コーパスコンコーダンサにコーパスラベリング機能を組み込むことにより、ラベリングのためのパターンの記述を簡便にすることができる。

本稿はコーパスコンコーダンサ『ChaKi.NET』[2] にラベリング機能を実装したので紹介する。本機能は既に実装されている文字列検索機能・形態素列検索機能・係り受け部分構造検索検索機能で記述可能なパターンによりラベリングができ、検索機能・WordList 機能を用いることで上に述べたサイクルを回すことが可能になる。サイクルを通して、良い精度・被覆率を与えることも可能であるが、多様なパターンを異なるラベルで記述することで、事後の工程に有用な情報をテキストに付与することができる。

例えば、事後の工程が人手によるアノテーションであるのであれば、作業者の見落としを防ぐために、良い被覆率の高いパターンを記述する。パターンごとの精度を勘案しながらアノテーション作業を進めることも可能になる。

また、事後の工程が機械学習に基づく系列ラベリングであれば、係り受けなどの長距離依存構造を、人手による知識としてパターンにより記述し、ラベルとして記号化することで、局所的な情報しか用いないモデルに対して大域的な特徴量として与えることができる。

以下２節ではパターンの記述方法について述べる。３節ではラベリングの実行方法について述べ、４節ではまとめと今後の展開について示す。

2 パターンの記述と対象の絞り込み

2.1 パターンの記述

パターンは形態素解析器 MeCab 出力に含まれる情報と係り受け解析器 CaboCha 出力に含まれる結果に基づいて記述する。具体的には、検索や Wordlist 機能のクエリ生成部 “Tag Search” (図 1) と “Dependency Search” (図 2) により発行する。

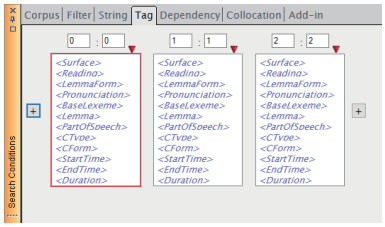

図 1: Tag Search によるクエリ生成部

“Tag Search”(図 1) は 1 つ以上の形態素の相対位置によりパターンを記述することができる。形態素の任意の属性により絞り込み検索ができ、各属性は完全文字列一致もしくは正規表現により指定することができる。

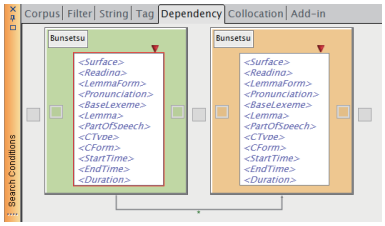

図 2: Dependency Search によるクエリ生成部

“Dependency Search”(図 2) は含む形態素情報により指定する文節の相対位置･係り受け関係によりパターンを記述することができる。文節内の形態素の位置関係として文節頭・文節末・2 形態素の隣接・2 形態素の順序関係が指定できる。文内の文節の位置関係として文頭・文末・2 文節の隣接・2 文節の順序関係・2 文節の係り受け関係が指定できる。

パターンにおいて、セグメントを切り出す範囲を指定する必要がある。現在のところセグメントはクエリ生成部により規定される 1 形態素とする。通常の KWIC 表示で Center 位置に、WordList で 0 位置に表示されるものを対象となる。

このように GUI 上で作成されたパターンは、ChaKi.NET のクエリの保存機能により、XML 形式で保存することができる。

2.2 対象の絞り込み

以降の例では簡単のために図 3 に示すパターンを用いて説明する。具体的には表層形「九州」に対して、固有表現であることを表す “NE” というラベルを付与することを目標とする。

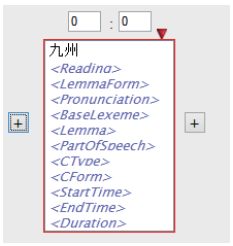

図 3: Tag Search によるパターン事例

検索結果を調べ、Segment 付与対象として問題のない項目にチェックを入れる。あるいは編集メニューの「すべての行をチェック」により全ての項目にチェックを入れ、付与対象としてふさわしくないもののみ、チェックを外すようにしてもよい。チェック後の検索結果の例（KWIC View）を図 4 に示す。

3 ラベリングの実行

ラベリングの実行は Scripting Panel にコマンドを発行することによる。Scripting Panel を開き、Script Engine として “IronRuby” を選択する。

コード入力ウィンドウ（上半分）に下記のようなコードを貼り付ける。

ラベリングの実行スクリプト（概要）

```
...
records.each do |r|
   svc.Open(corpus, s, nil)
   ...
   c = r.GetCenterCharOffset()
   w = r.GetCenterCharLength()
   svc.SetupProject(0)
   svc.CreateSegment(c, c+w, "NE")
   svc.Commit()
   ...
end
```

CreateSegment(startPos, endPos, tagName) により指定した範囲 [startPos, endPos-1] に tagName を付与する。startPos により Center の最左オフセットを得る。検索条件によっては Center の語長がまちまちとなることもあり得るが、GetCenterCharLength() で項目ごとに実際の語長を取得し、適切な範囲に付与する。

このとき、DB がロックされているとエラーとなるので、ロックを外す。（検索中なら検索を終える。Dependency Edit パネルまたは Attribute Edit パネルで編集中なら編集モード解除ボタンを押す。）

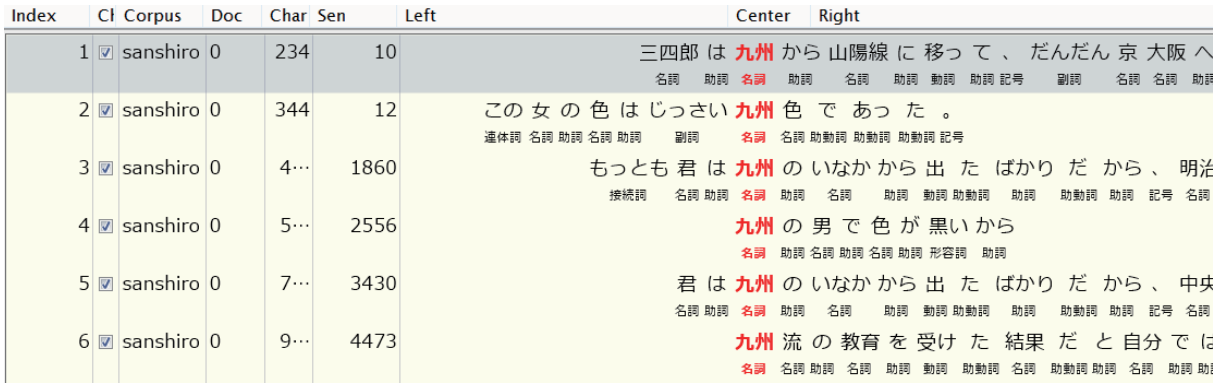

図 4: チェック後の検索結果の例（KWIC View）

正しく Segment が付与されたかどうかを Scripting Panel の出力メッセージおよび、Dependency Edit Panel 等で確認する。

その都度スクリプトツールボタンより実行スクリプトを入力するのは大変だが、スクリプトをあらかじめ準備しておき、登録済みスクリプトを呼び出すようにすればよい。

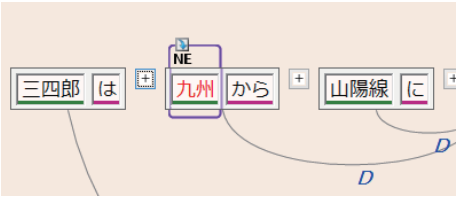

図 5: Segment 付与結果

上記の Dependency Edit Panel の表示は、正常に “NE” Segment が付与されたことを示している。

4 おわりに

本稿では ChaKi.NET のラベリング機能について紹介した。コーパス言語学において、研究者がコーパスを自分で構築することが増え、検索や統計処理による分析のみならず、規則に基づいて多様な情報を付与することが重要になりつつある。

今後の展開として、有用なパターンの蓄積が考えられる。『鳥バンク』[1] で記述されている文型パターンのうち、節定義部分を ChaKi.NET 上で付与できるように規則の記述・変換を行い、系列ラベリング学習モデルの特徴量に係り受けレベルの情報を与えるようにする。これらのラベリング機能に基づき、BCCWJ に対する節情報アノテーションを行う。新聞記事サンプルを中心に鳥バンクの節分類の第 3 段階レベルのアノテーションを進める。

また、同機能の利用者を増やすために、ChaKi.NET に用いたコーパスに対する統計処理や情報付与の講習会を開催していきたい。

謝辞

本研究の一部は科研費萌芽「近代語コーパスに対する統語情報アノテーション基準策定」(15K12888) および基幹型共同研究プロジェクト「コーパスアノテーションの基礎研究」および国語研「超大規模コーパス構築プロジェクト」によるものです。

参考 URL

- ChaKi.NET ダウンロードページ:`https://osdn.jp/projects/chaki/releases/`
- ChaKi.NET 用データダウンロードページ : `http://chaki-data.ninjal.ac.jp/`
- BCCWJ 関連データダウンロードページ : `https://bccwj-data.ninjal.ac.jp/mdl/`

参考文献

[1] 池原悟. 意味類型パターン記述言語仕様書. Technical report, 独立行政法人科学技術振興機構, 戦略的基礎研究事業, 高度メディア社会の生活情報技術, 2007.

[2] Yuji Matsumoto, Masayuki Asahara, Kou Kawabe, Yurika Takahashi, Yukio Tono, Akira Otani, and Toshio Morita. ChaKi: An Annotated Corpora Management and Search System. In *Proceedings from the Corpus Linguistics Conference Series*, 2005.

[3] J. Pustejovsky and A. Stubbs. *Natural Language Annotation*. O'Reilly, 2012.

[4] 浅原正幸, 小西光, 田中弥生, 加藤祥. 品詞列・係り受け部分木に基づくラベリングツールの設計と実装–節境界ラベリングを例に–. 第 8 回コーパス日本語学ワークショップ予稿集, pp. 83–92, 2015.