セグメントとリンクに基づくアノテーションツールの設計と実装

野口 正樹 † 三好 健太 * 徳永 健伸 † 飯田 龍 ‡ 小町 守 ‡ 乾 健太郎 ‡

† 東京工業大学 大学院情報理工学研究科
* 東京工業大学 情報工学科
‡ 奈良先端科学技術大学院大学 情報科学研究科

{mnoguchi, kmiyoshi, take}@cl.cs.titech.ac.jp, {ryu-i, mamoru-k, inui}@is.naist.jp

abstract

これまで，様々な情報が付与されたコーパスが構築されてきたが，付与する情報に応じて個別のアノテーションツールが開発されてきた．これらのツールは，独自のデータモデルや作業モデルを採用しているため，新しい情報を付与する場合には新たにツールを開発する必要があった．本論文では，内部に保持するデータをセグメントとリンクとして抽象化し，それに対する基本操作を定義することにより，コーパスに付与する種々の情報を統一的に扱う枠組みを提案する．これにより，付与する情報に応じてインタフェースを柔軟に設計できる．提案手法を用いた具体例として，述語項構造と句構造という異なる情報を付与するアノテーションツールを紹介する．

1 背景

近年，自然言語処理の分野では，コーパスに基づく統計的手法が研究の中心となっている．そのため，コーパスは統計的手法を用いた解析モデルの学習データや解析システムの評価用テストセットに用いられるなど重要な役割を果たしている．また，コーパスに付与される情報は多様化・複雑化しており，様々なアノテーションツールの開発が行われている．

しかし，既存のアノテーションツールは目的ごとに専用のツールとして開発されてきたため，データフォーマットなどはアノテーションツールごとに異なった形で表現されている．従って，ある目的のために付与された情報を別の目的で使用する場合，その都度変換作業が必要となる．

これまで，様々なアノテーションツールとともに付与される情報の記述形式がいくつか提案されている．例えば，テキストに対し構文や意味の情報を付与した GDA (Global Document Annotation)[2] や，CES (Corpus Encoding Standard)[3] が提案されている．

GDA は，統語的依存関係、代名詞等の照応、共参照、 多義語の語義など様々な情報を XML の形式で表現する．(図 1) CES は，コーパスのメタデータ，文章の構成などの情報を SGML の形式で表現している．しかし，これらのアノテーションでは情報の記述形式が決められているので，情報の相互利用にはコンテンツ間で情報の変換作業が必要になる．

```
<su>
  <np sem="time0">time </np>
  <v sem="fly1">flies </v>
  <adp>like <np>an arrow</np></adp>.
</su>
```

図 1: GDA のデータ例

```
# A-ID:950101003
村山富市首相は...
また、一九九五年中の衆院...
EOT
950101003
np 1.110, 1.113 id="9"
np 2.41, 2.45 id="12"
...
```

図 2: Tagrin のデータ例

一方，Tagrin[5] では，任意の文字列間の関係を表すために，文字列をスタンド・オフ形式で表現している．(図 2)

このように，アノテーションツールごとに付与される情報が異なっているために，情報を相互に利用することができない．

大規模で多様な情報が付与されたコーパスを構築することを考えると，様々なアノテーションで利用できるツールを容易に提供できるように基本となるデータ形式を整備する必要がある．

また，このようなフレームワークの一つに AGTK (Annotation Graph Toolkit)[1] がある．AGTK では，グラフを基本データ構造として，句構造や依存構造など付与する情報ごとにデータに対する操作を提供している．

本論文では，種々のアノテーション情報を抽象化し，それに対する基本操作を定義することで多様な情報を付与する枠組みを提案する．また，この枠組みを利用したアノテーションツールの実装例として，述語項構造のアノテーションツールと句構造のアノテーションツールを紹介する．

2 データの抽象化

データ形式の抽象化およびアノテーションツールの実装にあたり，アノテーションごとに作業のしやす

いインターフェースを採用することは，スムーズなアノテーションにもつながると考えられる．そのため，データの抽象化に当たり，付与する情報とインターフェースの情報とを区別して扱うことにする．

これにより，アノテーションツールの実装においては，付与する情報の操作 (生成・削除) という基本的な操作ができるようにインターフェースを設計すれば良く，分かりやすく柔軟に設計することが可能となる．インターフェースの情報は，アノテーションごとに自由に定義できる．

さらに，付与する情報をセグメントとリンクの 2 つのオブジェクトに限定し，付与する情報の詳細についてはアノテーションごとにコンフィグレーションとして定義する．

以降では，各オブジェクトとコンフィグレーションの役割と表現方法について説明する．また，インターフェースの情報については次節において実装例とともに紹介することする．

2.1 セグメント

アノテーションの対象となるドキュメント中の任意の文字列とその文字列に付与された情報 (ラベル) の対をセグメントと呼ぶ．セグメントにおける文字列はドキュメント中の開始位置と終了位置のオフセット値のペアで表現し，ラベル情報はコンフィグレーションにおいて定義する．セグメントオブジェクトは表 1 に挙げる属性を持つ．

表 1: セグメントの表現

属性名	属性値
segmentID	セグメントオブジェクトの ID
documentID	作業中のドキュメントの ID
label	ラベル
start	文字列の開始位置 (オフセット値)
end	文字列の終了位置 (オフセット値)

2.2 リンク

リンクは 2 つのセグメントオブジェクト間の関係を表すオブジェクトで，2 つのセグメントオブジェクトの組 (source, destination) で表現する．また，ラベル情報は，セグメントオブジェクト間の関係を表す情報で，コンフィグレーションにおいて定義する．リンクオブジェクトは表 2 に挙げる属性を持つ．

表 2: リンクの表現

属性名	属性値
linkID	リンクオブジェクトの ID
documentID	作業中のドキュメントの ID
label	ラベル
source	リンク元のセグメントオブジェクトの ID
destination	リンク先のセグメントオブジェクトの ID

2.3 コンフィグレーション

アノテーションによって付与される情報は，その目的によって異なるため，コンフィギュレーションでは，ラベルをキーとして，付与する情報が持つ属性・属性値を定義する．表 3 に品詞タグ付けにおける定義の例を示す．例えば，label の値が “動詞-接尾-サ変”である場合，セグメントオブジェクトは，セグメントの属性として，conjugation が “true”，conjugationtype が “サ変”という値を持つことを表す．

表 3: セグメントの属性 (例:品詞タグ付け)

属性名	属性値の型
label	品詞
conjugation	活用するか否か
conjugationtype	活用型
...	...

表 4 に述語項構造のアノテーションにおける定義の例を示す．例えば，label の値が “ガ格”であるリンクオブジェクトは，リンクの属性として，directed が “true”，transitive が “false”という値を持つ．

表 4: リンクの属性 (例:述語項構造)

属性名	属性値の型
label	述語と項の関係
directed	有向リンクであるかないか (true/false)
transitive	推移性があるかないか (true/false)
srclabel	リンク元となれるセグメントのラベル
dstlabel	リンク先となれるセグメントのラベル
...	...

従って，リンクのラベルの属性 directed が false であれば無向リンクを表現でき，セグメントオブジェクトの集合は推移的な無向リンクで接続されたセグメントオブジェクトの集合として表現できる．

また，リンクについては，srclabel,dstlabel によって，リンク元，リンク先になれるセグメントオブジェクトを制限することもできる．

3 ツールの実装例

この節では，本論文で提案した枠組みを用いて既存のツールと同様の機能を再実装した例について述べる．アノテーションごとに基本操作を実現する機能を実装している．

再実装したいずれのツールも，ドキュメントの表，セグメントの表，リンクの表を持つデータベースをバックエンドに持ち，インタフェースと通信しながら作業をする構成になっている．(図 3)

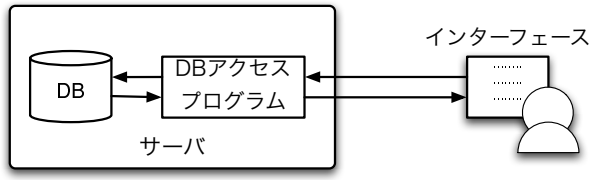

図 3: システム図

3.1 Tagrin

Tagrin は情報抽出や照応解析など多様なコーパス作成を目的としたアノテーションツール [5] である．提案手法を用いることで，任意の文字列をセグメントで表し，文字列の関係をリンクで表すことができる．

例えば，述語に対して格関係の情報を付与する述語項構造のアノテーションの場合，ユーザは対象とする述語を選択した後に，格関係にある文字列を選択し，選択された述語と文字列との間の関係を付与する．再実装したツールでは，述語ごとに格関係を付与する操作を次のように実現する．

1. リンク元となるセグメントをキーボードの shift キーを押しながらクリックで選択する．もしくは，矢印キー [←, →] で選択する．
2. リンク先となるセグメントをクリックで選択する．もしくは，文字列を選択する．
3. リンクを作成するためのキーバインドを入力する．

共参照関係についても同様に，リンク元とリンク先を選択して directed 属性が “false”とコンフィグレーションに記述されたラベルを持つリンクオブジェクトを作成する．

また，これらの機能が使いやすいようにインターフェースを実現するため，コンフィグレーションにインターフェースの情報として次の属性を加えて実装した．(表 5, 図 4)

表 5: リンクの追加設定

属性名	属性値
label	セグメントオブジェクト間の関係
...	...
keybind	リンク作成時のキーバインド
srcfcolor	リンク元セグメントの文字色
srcbcolor	リンク元セグメントの背景色
dstfcolor	リンク先セグメントの文字色
dstbcolor	リンク先セグメントの背景色

これにより，キーごとに付与するラベルを定義しているので，人手で入力する必要は無い．また，文字列を選択してリンクを作成した場合には，リンク先に対応するラベルを持つセグメントを作成する．

3.2 eBonsai

eBonsai は句構造のアノテーションツール [4] である．eBonsai のアノテーション作業は，文に対してパーザが出力した複数の句構造候補から正しい句構造を選択するタスクが中心になる．提案手法を用いると，非終端記号をセグメントで，親子関係をリンクで表現できる．

eBonsai での基本操作は，不要なセグメント，リンクを削除することである．再実装したツールではこれらの操作を行うために次の 2 つの機能を実装した．

1. 複数の候補セグメントが存在する文字列 (非終端記号) から正しいセグメントを選択し，それ以外のセグメントを削除する．
2. 複数の候補リンクが存在するセグメントから正しいリンクを選択し，それ以外のリンクを削除する．

また，階層構造が重要になるため，候補を句構造表示するインターフェースを採用している．(図 5)

4 まとめと今後の課題

本論文では，種々のアノテーション作業時に保持するデータをセグメントとリンクに抽象化し，それらに対する基本操作を定義することで多様な情報を付与する枠組みを提案した．また，この枠組みを利用したアノテーションツールの実装例として，述語項構造のアノテーションツールと句構造のアノテーションツールを紹介した．

この枠組みにより，データを共通で管理することは可能になった．しかし，依然として作業者間のアノテーション結果の精度は，作業者のアノテーションスキル (学習具合) に依存している．そのため，より良い精度のコーパスを作成するためには，作業者が正しいアノテーションが可能になるような支援を実現することなどが今後の課題である．

参考文献

[1] http://agtk.sourceforge.net/. *AGTK: Annotation Graph Toolkit*, 2002.

[2] http://i-content.org/gda/. *Global Document Annotation*, 2002.

[3] http://www.cs.vassar.edu/CES/. *Corpus Encoding Standard*, 2000.

[4] 市川宙, 野口正樹, 吉田恭介, 橋本泰一, 徳永健伸, 田中穂積. 構文木付きコーパス作成支援統合環境：ebonsai. 言語処理学会 第 11 回年次大会, Mar 2005.

[5] 高橋哲郎, 乾健太郎. アノテーションツール”tagrin”の紹介. 言語処理学会 第 12 回年次大会, Mar 2006.

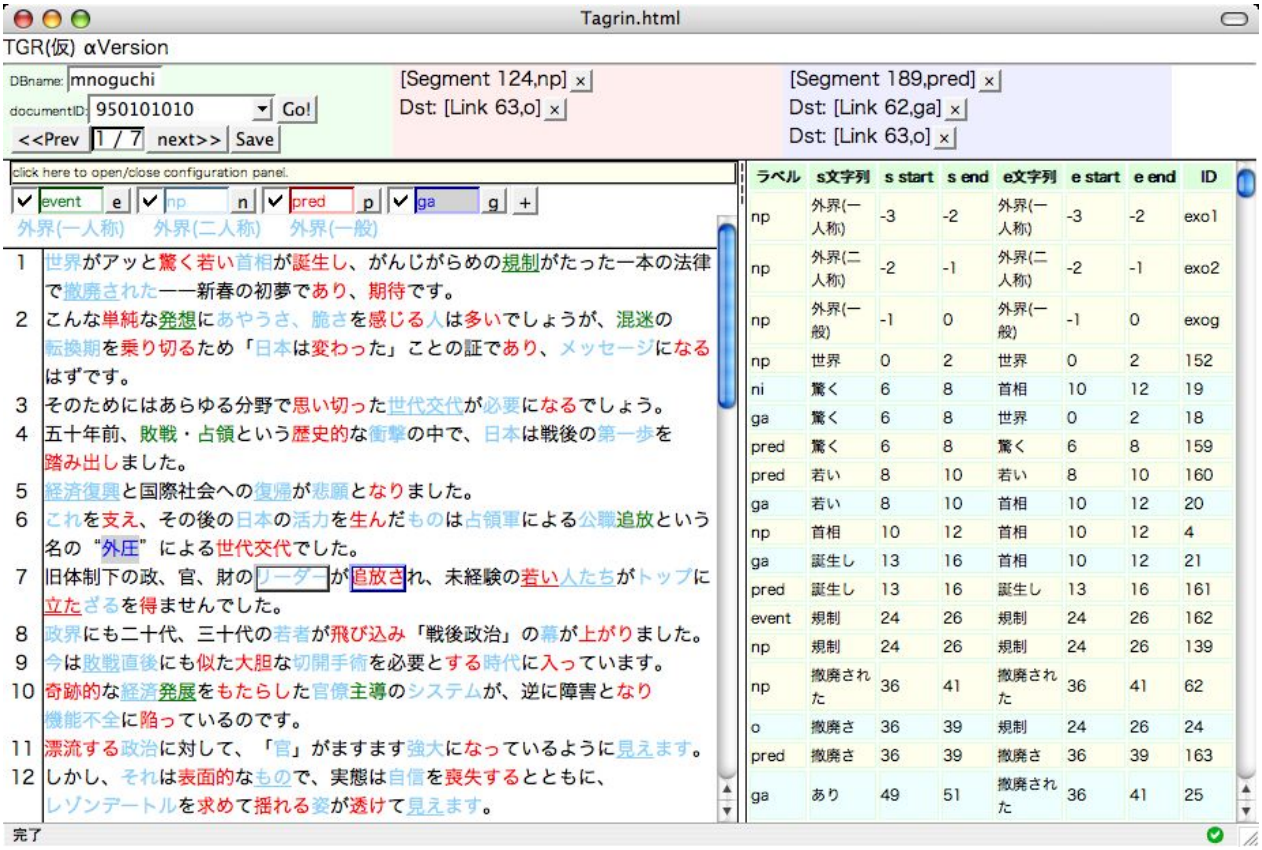

図 4: Tagrin の実装例

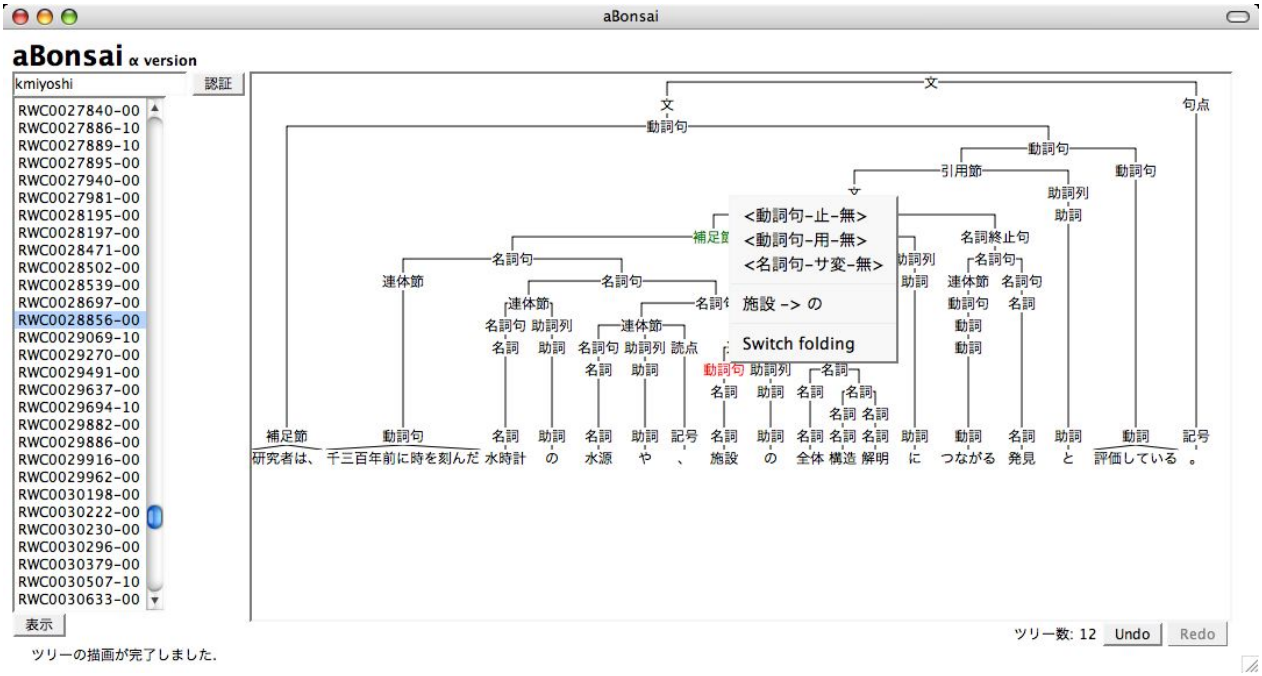

図 5: eBonsai の実装例