汎用アノテーションツール Slate

Dain Kaplan　　飯田 龍　　徳永 健伸

東京工業大学 大学院情報理工学研究科

{dain,ryu-i,take}@cl.cs.titech.ac.jp

1　はじめに

これまでに様々なプロジェクトを通して，様々なコーパスが作成されており，効率よく，信頼性の高いコーパスを作るために専用のコーパス作成ツールもまた数多く作られてきた．プロジェクトの主眼がコーパスの作成に置かれることが多いことから，コーパス作成ツールは作成するコーパスに特化し，必ずしも汎用性が高くわけではなく，再利用可能なものも少なかった．たとえば，Serengeti [11] はテキスト中の共参照関係のアノテーションのために開発されたツールであり，共参照関係をアノテーションするためには適しているかもしれないが，他の用途には必ずしも適していない．しかし，一般にソフトウェア・システムを作成するにはコストがかかるため，同じような機能を持つツールをコーパスごとに作成するよりは，汎用のツールを用い，ツールの作成のための資源をコーパス作成に割り当てる方が望ましい．また，コーパス作成の初期の段階ではアノテーションの方法が必ずしも厳格に定義できていることは少なく [6]，アノテーションの仕様の変更にともなってツールの変更も必要となる可能性もある．このような観点からも汎用的で柔軟なコーパス作成ツールが望まれている．これまでにも汎用のアノテーション・ツールを開発する試みはいくつかあったが [1, 8, 10]，必ずしも成功しているとはいえない．

Dipper らはコーパス作成のためのアノテーション・ツールを汎用性に関する以下の7つの観点から分類している [3]．(1) 扱うデータの多様性，(2) 多層のアノテーション，(3) アノテーションの多様性，(4) 簡便さ，(5) カスタム化可能性，(6) 品質の保証，(7) 相互変換可能性．我々もこれらの (1) から (5) の観点を重視して汎用性の高いアノテーション・ツール SLAT (Segment and Link-based Annotation Tool) [9] 開発を進めてきた．しかし，これらの観点はいずれも，アノテーションの対象となるコーパス中心の観点であり，コーパス作成のプロセスをどのように管理するかという視点が欠けている．コーパスの規模はますます大きくなり [2]，またアノテーションされる情報もますます複雑化・多様化している．今日ではすでにアノテーションされた既存のコーパスにさらに別の情報をアノテーションするような多層的なアノテーションが一般的となっている [4, 7]. 前述の汎用ツールはこのような多層的なアノテーションを扱うことは考えていない．以上のような背景から，アノテーションの汎用性を重視した SLAT を拡張し，コーパス作成のプロセスの管理まで視野に入れた枠組を開発し，Slate (SLAT Enhanced) として実装を進めている．本稿では，まず，コーパス作成過程において必要となる機能を洗い出し，それを基礎として設計した枠組と，その枠組に基づいて実装したアノテーション・ツール Slate について紹介する．

2　コーパス作成に求められるもの

コーパスに対する要求は，量的な拡大と同時に，互いに関連した様々な情報を重層的に付与するといった質的な拡大も高まっている．この要求を満たすためには，アノテーション・ツール自身の柔軟性やインターフェースの洗練が必要であることはもちろん，複数のアノテーションの間の関係や作成プロセスに関わるアノテータやデータの管理までも視野に入れる必要がある．ここでは，前述の Dipper らの考察に加え，より広い視野からコーパス作成においてシステムが支援すべき項目を検討する．

(1) ユーザ管理・役割管理
ある程度の規模のコーパス作成には複数の人間が関与するのが普通である．これらのユーザはコーパス全体の設計やアノテーションすべき情報を設計する管理者と実際のアノテーション作業をおこなうアノテータに大別することができる．システムはこれらのユーザ種別とその権限を管理できなければならない．

(2) タスクの割り当てと進捗管理
管理者はアノテータにタスクを割り当て，アノテータの進捗を管理する．進捗によってはタスクを別のアノテータに割り振ることも必要かもしれない．

(3) 新しいタスクの生成
管理者は新しいアノテーション・タスクを容易に生成できなければならない．

(4) タスクの修正
アノテーション・タスクの初期の段階では，アノテーションの設計が流動的でアノテーションの設計自身の修正が発生することがある．このような場合，システムはアノテーション・タスクの修正に柔軟に対応できなければならない．

(5) アノテーションの分析・統合
複数のアノテータがコーパスを分割してアノテートすることを考えると，同じテキストに異なるアノテータがアノテーションした際の一致率などの分析や，テキストを分割してアノテーションした際の結合などを支援する必要がある．

(6) 版管理
アノテーションの設計変更によりコーパスに複数の版ができる可能性がある．いわゆる版管理の機能が必要である．

(7) 多層アノテーション
すでにアノテーションされたコーパスにさらに別の情報をアノテーションするような多層的なアノテーションが最近では多く試みられている．このためには，現在作業中のアノテーションから既にアノテーションされた情報を参照するなどの機能が必要となる．

(8) 他システムとの連携
アノテーションの種類によっては，自動的にアノテーションをおこない，それを人手で修正した方が効率がよい場合もある．これを実現するためには，他のシステムの出力を柔軟に受け入れるなどの機構が必要である．

(9) 入出力の拡張性
種々の入出力フォーマットに対応できる必要がある．

(10) 多言語処理
依然としてその量においては英語のコーパスが主流であるが，現在では様々な言語のコーパスが作成されるようになってきた．多言語への対応は必須である．

3 枠組みの概要

前節で述べた項目は大別すると，枠組みの問題 (1)~(7) と実装の問題 (8)~(10) に分類することができる．図 1 に我々の枠組みの概要を UML で記述したものを示す．この枠組みで直接的に対応しているのは前述の項目のうち (1)~(4) と (7) である．

文字列から成る Document は，内容を格納するための最小の要素である．複数の Document をまとめて Document set を構成できる．たとえば，ひとりのアノテータがタグ付けする Document のまとまりを Document set として定義するのは自然な使い方であろう．Project はアノテーション・タスクを定義するもので，ひとつ以上の Document set を参照できる．Document set は Project とは独立に定義されているので，複数の Project から同じ Document set を参照することもできる．この構造によって多層的なアノテーションを自然に実現できる．User (アノテータ) はひとつ以上の Project に属することができ，Project ごとに異なる Document set のタグ付けをおこなうことができる．また，Project はひとつ以上の Tagset を使うことができるので，たとえば，形態素レベルの Tagset と同時に統語レベルの Tagset を用い，多層的なアノテーションをおこなうことができる．Tagset には高橋らが提案した Segment と Link [12] の基本要素が含まれる．Segment はどのような Attribute や Constraint を持つかを規定することによって定義し，実際のアノテーションによってそのインスタンスが作成され，label，Attribute の値が設定される．

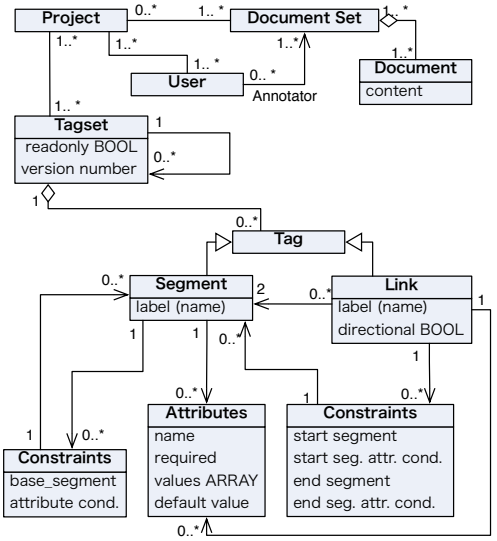

図 1: 枠組みの UML 表現 (簡略版)

図 1 の枠組を使ってアノテーション作業がどのように管理できるか例を用いて説明する．Penn Treebank [6] のように形態素・統語情報を付与するプロジェクトを考えよう．たとえば，これを Project A とする．Project A では Tagset X を使うものとしよう．次に Project A の成果の上にさらに PropBank [5] のような述語-項構造を付与することを考える．これを Project B として，Tagset Y を使うものとする．これらのプロジェクトをまったく独立におこなうと，2 つのアノテーションの間で情報を付与する対象に矛盾をきたす可能性がある．我々の枠組では，Tagset X と Y の間に適切な制約を記述することにより，不用意なミスを防ぐことができる．この場合，Project B は Tagset Y と同時に Tagset X も参照

することになる．

大規模で複雑なコーパスの構築には複数のアノテータが関与することになる．つまり，ひとつの Project には複数の User が関連付けられる．User には種々の権限が付与されるのが普通であり，その管理も支援対象として考慮すべきであろう．User は，Project を企画し監督する立場の管理者と，指示にしたがって Document に対してアノテーションをおこなうアノテータに大別できる考えられる．管理者は Project で用いる Tagset を決めたり，Document set をシステムにアップロードし，アノテータに割り当てる．場合によっては作業の途中で Tagset に修正を施すかもしれない．一方，アノテータにはこのような権限はないのが普通であるが，タグ付けのインタフェースにおける「見え」をカスタマイズすることは許されるかもしれない．これらの複雑な権限管理は Project と User の属性として定義することができる．

すでに述べたように我々の枠組では，ある Document set に対して複数の Tagset を関連付けることができるので，多層的なアノテーションを容易に実現できる．逆に，ひとつの Tagset を複数の Document set に関連付けることができるので，異なる Document のセグメント間にリンクを付与することも可能である．これはたとえば，複数文書とそれらの複数文書から生成された要約の間の関係を記述するのに有用である．

我々の枠組ではアノテーションの定義をユーザがアノテーション対象とは独立におこなえることが最大の特徴である．これによって，既存のアノテーションを利用したアノテーションを定義すれば多層的なアノテーションを実現できる．

4 Slate

SLAT では Web ブラウザベースのインタフェースでセグメントとリンクによるアノテーションを提供していたが，Slate (Segment and Link-based Annotation Tool Enhanced) では，User，Project，Document set などの管理も含めた実装となっている．また，SLAT では Javascript を用いて実装していたために，動作速度が遅いという問題が指摘されていたが，Slate では Adobe Flash と Google Web Toolkit (GWT) を用いて全面的に実装をやり直したために，動作が高速化できた．また，Adobe Air 環境を用いて Web ブラウザとは独立のアプリケーションとしても動作できる．

コーパス作成作業の流れの概略は以下のようになる．

管理者の作業

- Project の作成
- Tagset の定義
- アノテーションする文書のアップロード
- Document set の作成
- Users (アノテータ・アカウント) の作成
- Project の定義 (User，Tagset，Document set の関連付け)
- アノテーション作業の割当

アノテータの作業

- 割当られた作業の選択
- アノテーション作業の開始

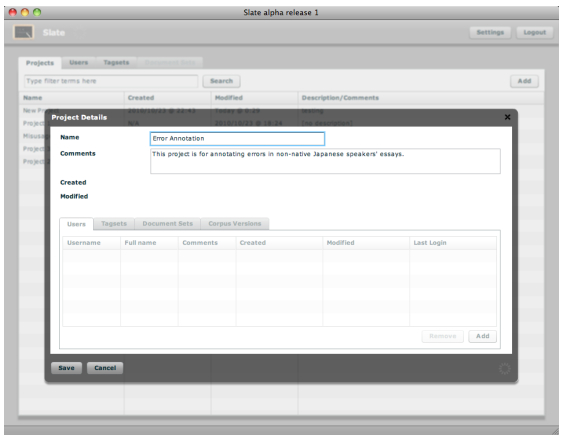

図 2: Slate の管理画面の例 (プロジェクトの作成)

図 2 に Slate の管理画面の例を示す．これは新規プロジェクトを作成する画面で，タブを切り換え User，Tagset，Document set を指定することにより，これらを関連付けることができる．

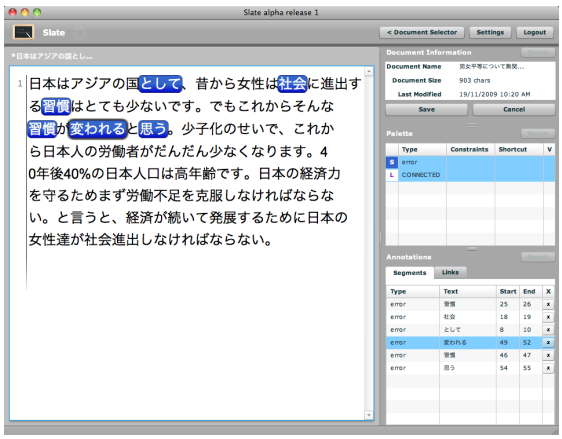

図 3: アノテーション・インタフェース

アノテータは図 3 に示すインターフェース画面からアノテーション作業をおこなう．基本的なインターフェー

スは SLAT を踏襲しているが，タグ・セットの表示，タグ一覧，文書情報などを右側に集約し，これらをサイズ変更が可能なペインとして用意した点が大きく異なる．

5 おわりに

本稿では，アノテーション・タスクの複雑な構造をアノテーション・タスクに関わる実体の関係を関係–実体モデルでモデル化する枠組みを提案した．この枠組によれば，これらの実体とその関係の定義によりアノテーションを定義することができる．また，近年のアノテーションの主流である多層的なアノテーションも自然に扱うことができる．この枠組の実装として Slate を紹介した．

謝辞

この研究は文部科学省科学研究費補助金特定領域研究「代表性を有する大規模日本語書き言葉コーパスの構築」の一貫としておこなわれた．

参考文献

[1] Hamish Cunningham, Diana Maynard, Kalina Bontcheva, and Valentin Tablan. GATE: an architecture for development of robust hlt applications. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pp. 168–175, 2002.

[2] Mark Davies. The 385+ million word corpus of contemporary american english (1990–2008+) design, architecture, and linguistic insights. *International Journal of Corpus Linguistics*, Vol. 14, No. 2, pp. 159–190, 2009.

[3] Stefanie Dipper, Michael Götze, and Manfred Stede. Simple annotation tools for complex annotation tasks: An evaluation. In *Proceedings of the LREC Workshop on XML-based Richly Annotated Corpora*, pp. 54–62, 2004.

[4] Ryu Iida, Mamoru Komachi, Kentaro Inui, and Yuji Matsumoto. Annotating a Japanese text corpus with predicate-argument and coreference relations. In *Proceedings of the Linguistic Annotation Workshop*, pp. 132–139, 2007.

[5] Paul Kingsbury and Martha Palmer. From treebank to propbank. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC 2002)*, pp. 1989–1993, 2002.

[6] Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of english: The Penn Treebank. *Computational Linguistics*, Vol. 19, No. 2, pp. 313–330, 1993.

[7] Eleni Miltsakaki, Rashmi Prasad, Aravind Joshi, and Bonnie Webber. The Penn Discourse Treebank. In *Proceedings of 4th International Conference on Language Resources and Evaluation (LREC 2004)*, pp. 2237–2240, 2004.

[8] Christoph Mueller and Michael Strube. MMAX: A tool for the annotation of multi-modal corpora. In *Proceedings of the 2nd IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems*, pp. 45–50, 2001.

[9] Masaki Noguchi, Kenta Miyoshi, Takenobu Tokunaga, Ryu Iida, Mamoru Komachi, and Kentaro Inui. Multiple purpose annotation using SLAT – Segment and link-based annotation tool. In *Proceedings of 2nd Linguistic Annotation Workshop*, pp. 61–64, 2008.

[10] Constantin Orăsan. PALinkA: A highly customisable tool for discourse annotation. In *Proceedings of the 4th SIGdial Workshop of Discourse and Dialogue*, pp. 39–43, 2003.

[11] Maik Stührenberg, Daniela Goecke, Nils Diewald, Alexander Mehler, and Irene Cramer. Web-based annotation of anaphoric relations and lexical chains. In *Proceedings of the Linguistic Annotation Workshop*, pp. 140–147, 2007.

[12] 高橋哲朗，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 回年次大会発表論文集, pp. 228–231, 2006.