汎用アノテーションツールSLAT

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 計算工学専攻

‡ 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

1 背景

近年、自然言語処理の分野では、大規模なコーパスから得られる統計情報を基にした解析や応用に関する研究が盛んである。特に統計的手法の有用性が確認されてからは、様々な目的に統計的手法が使われるようになった。そのため、コーパスの利用方法が増し、その目的に合わせてコーパスに付与される情報も多様化している。

コーパスに対する情報の付与を全て人手で行うことは非常に多くの時間を費やすうえ、入力ミスなどの誤りが増える原因になる。コーパスに付与される情報の偏りや誤りは、開発する解析器の性能や解析時の評価に大きく影響を与えるため、コーパスには付与された情報の一貫性や精度が求められる。各コーパスの構築プロジェクトでは情報の付与をサポートするツールを開発し [1, 2, 3]、入力の簡略化や制約を加えることで、情報付与のコスト削減を実現した。しかし、これらのコーパスに付与された情報の保存形式はツールによって異なるため、データ形式が統一されておらず、データをそのまま相互利用することができない。そのため、異なるコーパス間での実験や手法の評価のためには変換作業が必要になり、コーパス毎に変換処理を行う必要がある。

また、これまで作成されたコーパスに対して、データ形式や表現方法の規格化などの議論もあり [4, 5, 6]、今後はこれまでのアノテーション形式やこれまでの議論を視野に入れながら汎用的なタグ付けツールを開発する必要がある。

本論文では、コーパスを構築するプロジェクトの視点から見たツールに対する問題点を挙げ、その問題を解決するため開発した汎用アノテーションツール SLAT を紹介する。また、多様なアノテーションに対応するための設定方法について具体例と共に述べる。最後に本論文をまとめ、今後の課題について触れる。

2 問題点

これまで開発されてきたツール [1, 2, 3] では、主にアノテーション作業自体のユーザビリティに注目が注がれていた。しかし、コーパスの構築を目的とする大きな視点で捉えると次のような問題が残されている。

1. アノテーションツール導入のための敷居が高い
   特に言語学者などコンピュータに詳しくない人にとって、インストール作業自体が負担となる。処理をするデータは作業者のマシン内にあり、作業後のデータを渡すプロセスが必要になる。
2. タスクに依存したツール・仕様・データ形式
   各々の利用目的に応じて、その目的に合った情報を付与出来るツールの開発が必要である。付与される情報が様々なので、それに応じてデータ形式も異なり、相互利用が困難である。
3. コーパスの品質維持
   複数の作業者がアノテーションを行う際は、各作業者が仕様書を参照しながらアノテーションをするため、判断基準は各作業者にゆだねられる。その結果、付与される情報の一貫性を保持することが困難である。特に、プロジェクトの初期段階では、付与する情報の仕様改訂が頻繁に起こる。こういった場合には、参照する仕様書が何度も書き変わるため、作業者は対応することが難しい。

3 SLAT

Segment and Link-based Annotation Tool (SLAT) は Tagrin[7] をベースに、第 2 章の問題点を解消するために開発した汎用アノテーションツールである。

SLAT はクライアント/サーバシステムを採用し、アノテーションの設定をサーバ側で管理することにより、Tagrin では必要だった仕様変更等による各作業者のマシンの設定更新作業を省くことができる。また、SLAT は複数の情報が付与された場合、それらを区別

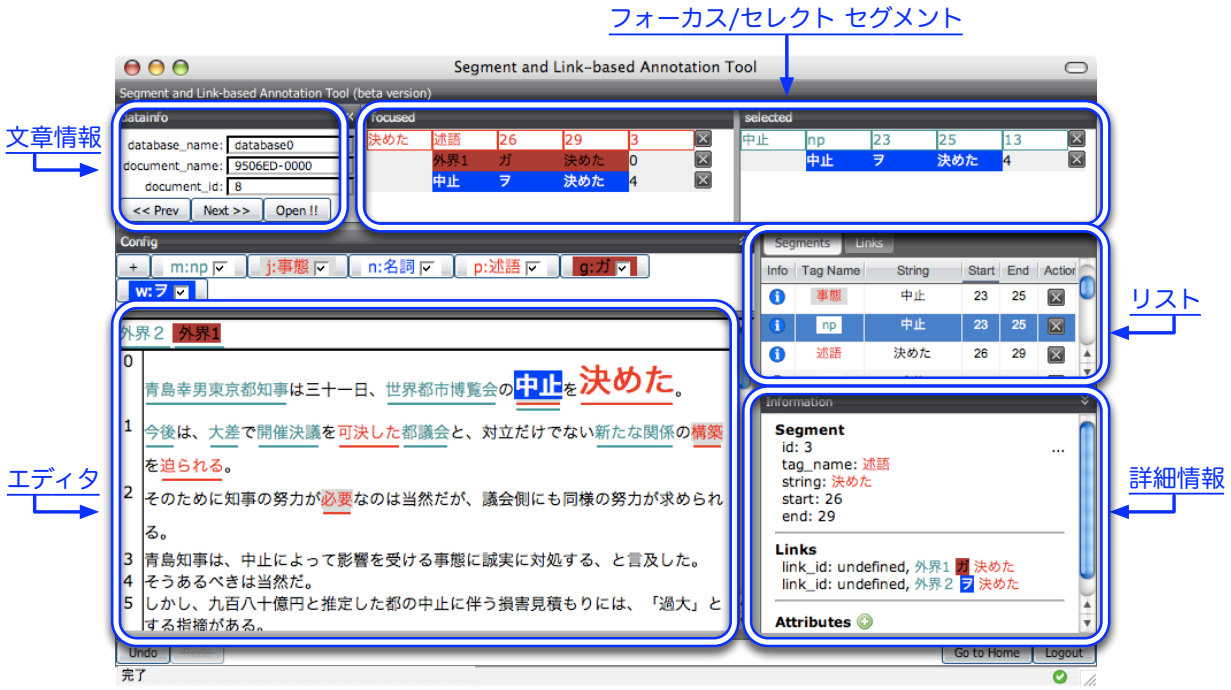

図 1: SLAT のスナップショット

して表示できるように拡張している。

図 1 に SLAT のスナップショットを示す。

3.1 クライアントサーバ型

SLAT はクライアント/サーバ型のアプリケーションである。クライアントに Web ブラウザ (Firefox) を採用することで、作業者がアノテーションツールを導入する際の負担軽減を実現した。これにより、作業者は Web ブラウザで指定された URL にアクセスするだけでアノテーション作業に取りかかることができる。

また、複数の作業者でアノテーションをするコーパス構築プロジェクトにおいて、SLAT ではプロジェクトの管理者はサーバサイドのメンテナンスをするだけで良く、各作業者のマシンやツールを管理する負担を省くことができる。

SLAT は現在は XML 形式でのインポート/エクスポートをサポートしている。今後、個別にデータを追加ができるインターフェースを用いることで、作業者が XML ファイルを用意することなく記事を追加できるよう対応する予定である。

3.2 アノテーション方法

SLAT では汎用性を実現するため、既存のコーパス構築プロジェクトを参考に、付与する情報をセグメントとリンクへ抽象化し、それら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取り入れた [8]。

セグメント

セグメントはテキストの領域を特定しその領域に情報を付与する際に用い、テキスト中の領域に付与する情報を表す文字列 (タグ名) と領域に含まれる文字列、領域の開始位置/終了位置を持つ。形態素、句や節の範囲を指定する場合にも用いることができる。

リンク

リンクはセグメント間の関係を付与する際に用い、リンクの元/先となるセグメントと関係を表すタグ名としての文字列を持つ。セグメント間の関係には、推移性と有向性という少なくとも 2 つの性質があり、その組み合わせにより 4 種類の関係が考えられる。

1. 推移的有向関係
   “車”→“ドア”→“ガラス”のような part-of の関係がこの種類に属する。
2. 推移的無向関係
   並列表現や、“太郎” と “彼” などの同一指示の関係がこの種類に属する。
3. 非推移的有向関係

“本を買う”という文の “買う”と “本”のような、意味役割の関係などがこの種類に属する。

4. 非推移的無向関係
   対を表す関係である。

グループ

リンクにおける 2. 推移的無向関係については、同様の関係にあるセグメントの集合をグループとして扱う。

図 1 に示すように、SLAT ではウィンドウの中央左側にはアノテーション対象のテキストとタグを表示するエディタが、中央右側には付与されたタグの一覧がリスト形式で表示される。

作業者はエディタ中の任意の文字列をハイライトしてセグメントを付与することができる。また、フォーカス/セレクトと区別して 2 つセグメントを選択することができ、それぞれ画面上部のエリア (図 1) にセグメントの情報を表示する。フォーカスとセレクトはそれぞれリンクを付与する際にリンクの元と先として扱われ、作業者は 2 つのセグメントを指定して、これらのセグメント間にリンクを付与することができる。

SLAT のエディタ内では、テキスト中の文字列に下線を付与し色づけすることでセグメントが表示される。複数のセグメントが同一文字列に付与された場合は下線が複数本表示される。また、SLAT のエディタではテキスト上に全てのリンクを表示するのではなく、あるセグメントを選択した時にのみ、そのセグメントに関連するリンクを色づけして表示する。

リストには、セグメントやリンク、グループの一覧を、各オブジェクトのプロパティとともに表示する。リストのヘッダをクリックすることでプロパティによる並び替えができる。また、リスト内のセグメントをクリックすることによって、そのセグメントを選択でき、選択されたセグメントを表示するようにエディタが自動的にスクロールする。リスト内のリンクをクリックすると、リンク元およびリンク先のセグメントが選択できる。これにより、作業者は付与された情報を容易に確認できるようになった。

3.3 汎用性

SLAT では設定ファイルを記述することで多様なアノテーションに対応することができる。従って、SLAT を導入するとアノテーションの種類に対して設定ファイルを書き換えるだけでアノテーションの変更に対応することができる。プロジェクトの早期段階での仕様変更にも柔軟に対応可能である。

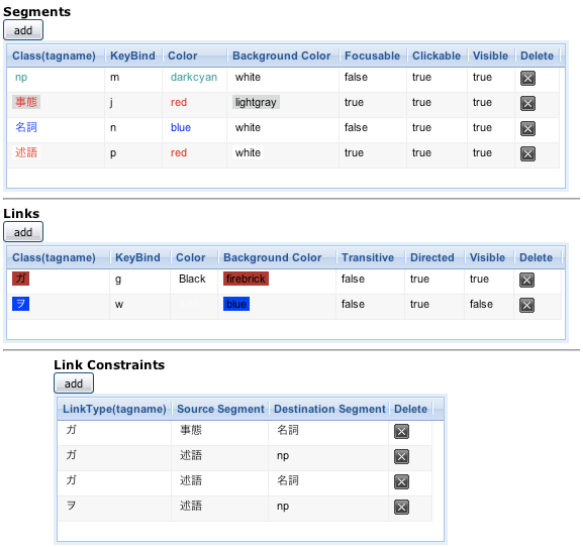

図 2: 設定ファイル編集用の GUI

4 設定方法

SLAT では、付与する情報の種類に応じて様々な設定をすることができる。この設定を適切に行うことで種々のアノテーションに対応することができる。

特定の文字列や表現に対して情報を付与したい場合には、セグメントを設定をする。また、複数の表現間の関係づけを行いたい場合には、関係づけられる表現を表すセグメントとその表現間の関係のためのリンクやグループを設定する。

アノテーションごとにこれらの設定をすることができ、複数種類の設定、例えば、述語項構造と共参照情報の設定を並列に設定することも可能である。この場合、対象とするタグのみを表示して作業を行ったり、作業の対象を変更するなど、設定の変更により作業者が自由にアノテーションを定義することができる。

セグメント、リンク、グループそれぞれどのような設定が可能かについて述べる。

XML ファイルを用いた設定方法と GUI を用いた方法の 2 通りの方法で設定することができる。図 2 に編集用の GUI の一部を示す。

4.1 セグメントに関する設定

セグメントの設定には、付与する “タグ名”とその時に用いる “キーバインド”、表示に用いる “文字色”・“背景色”の他、クリックできるか、フォーカスできるか、操作や表示をするか否かをそれぞれ指定する。

フォーカスを指定すると、キーボードの “矢印キー” の左右を使うことで、文章中のフォーカスが指定してある全セグメントを先頭から順に移動することが出来る。これにより、情報を付与する対象を順に辿ることができ、アノテーションすべき対象の取りこぼしを防ぐことができる。

例えば、固有表現のアノテーションの場合には、付与する情報は固有表現の ‘範囲’ と ‘種類’ である。設定ファイルには、セグメントの “タグ名” に ‘人名’ や、‘組織名’ といった固有表現の種類を指定することで対応できる。

4.2 リンクに関する設定

リンクの設定には、セグメント同様に付与する “タグ名” とその時に用いる “キーバインド”、表示・操作をするか否かの他、表示に関してリンクの元のセグメントとリンクの先のセグメントそれぞれの表示に用いる “文字色” ・“背景色”、そのリンクの “推移性” と “有向性” も指定する。この他、リンク元/先として選択できるセグメントのタグ名の組を一組以上指定する必要がある。複数指定した場合には、選言として解釈され、リンク元/先の制約として用いる。この制約により、リンクが意図しないセグメント間に付与されることを防ぐことができる。

例えば、述語項構造のアノテーションの場合には、図 2 のように、‘’ 述語 ’ とその述語の項に対応する名詞句となるセグメント ‘np’ の設定と述語と項の関係を表すリンクの設定を記述すれば良い。

4.3 グループに関する設定

グループの設定には、セグメントやリンク同様に付与する “タグ名” とその時に用いる “キーバインド”、表示・操作をするか否かの他、要素となるセグメントの表示に用いる “文字色” ・“背景色” を指定する。この他、グループの要素になりうるセグメントのタグ名を一つ以上指定する必要がある。複数指定した場合には、選言として解釈され、要素の制約として用いる。この制約により、グループが意図しないセグメントを要素と取ることを防ぐことができる。

例えば、共参照情報のアノテーションの場合には、共参照関係になるセグメントの設定を記述し、グループの要素となるセグメントのタグ名を ‘名詞句’ や ‘代名詞’ など設定すれば良い。

5 まとめと今後の課題

コーパス構築プロジェクトの作業内容を考慮した、導入の容易な汎用アノテーションツール SLAT を紹介した。SLAT は設定ファイルの記述によって様々なアノテーションに対応することが可能である。

今後は、コーパスの品質維持を実現するために、アノテーションフローを利用したヘルプの提示、コーパスの品質を保つためのバッチ処理の導入、仕様の変更や問題点の報告のための機能などの課題に取り組む予定である。

参考文献

[1] NEGRA project. @nnotate. http://www.coli.uni-saarland.de/projects/sfb378/negra-corpus/annotate.html

[2] GATE, A General Architecture for Text Engineering. http://gate.ac.uk/

[3] Constantin Orăsan. (2003). PALinkA: A highly customisable tool for discourse annotation. In Procceedings of the 4th SIGDIAL Workshop on Discourse and Dialogue

[4] Steven Bird, et al. (2000). ATLAS: A flexible and extensible architecture for linguistic annotation In Proceedings of the Second International Conference on Language Resources and Evaluation.

[5] Nancy Ide, Keith Suderman. (2006). Merging Layered Annotations In Procceedings of Workshop “Merging and Layering Linguistic Information”

[6] Udo Hahn, Ekaterina Buyko, Katrin Tomanek, et al. (2007). An Annotation Type System for a Data-Driven NLP Pipeline. In Proceedings of the Linguistic Annotation Workshop

[7] 高橋哲郎, 乾健太郎. (2006). アノテーションツール “Tagrin” の紹介. 言語処理学会第 12 回年次大会予稿集.

[8] 野口 正樹, 三好 健太, 徳永 健伸, 飯田 龍, 小町 守, 乾 健太郎 (2007). 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. 言語処理学会第 13 回年次大会予稿集.