GuideLink：
ガイドラインの管理を同時に行うアノテーションツール

大内田賢太 † 金進東 † 辻井潤一 †‡§
† 東京大学情報理工学系研究科コンピュータ科学専攻
‡School of Computer Science, University of Manchester
§National Centre for Text Mining, University of Manchester
{oouchida, jdkim, tsujii}@is.s.u-tokyo.ac.jp

1　はじめに

近年、計算言語学の世界では、我々は様々なテキストデータを使用することが可能になり、それらのコーパスに対して様々な情報を付与 (アノテーション) し、アノテーションされたコーパスから言語処理用知識を得る手法が一般的に用いられている。それゆえ、コーパスへのアノテーションは計算言語学の世界で重要なテーマの１つになっている。現在、コーパスへのアノテーション手法として、人手によるアノテーションが広く行われている。というのも、人手によるアノテーションは、人の言語知識をより正確にコーパスにアノテーションすることができると考えられるからである。しかし、アノテーションの一貫性が崩れることによってエラーが生じる可能性も考えられる。このことから、アノテーションの一貫性の維持が、人手によるアノテーションの重要な観点といえる。

我々は、アノテーション作業を、記述子をコーパス上の単語列に割り振る作業と考える。この記述子はアノテーション作業の前にあらかじめ定義され、どのような単語列に対して割り振られるべきかの基準を持つ。アノテーターはこの基準を基に、記述子を単語列に割り振ることになる。

アノテーションの一貫性が失われる原因となるケースは、いくつか考えられる。一般的に、アノテーターはどのようにアノテーションすべきか難しい事例に直面したとき、どのようにアノテーションするべきか自ら決定を行う。この決定は、アノテーターの頭の中にある、記述子における基準の影響を受ける。しかし、アノテーション作業が複数のアノテーターによって行われる場合はアノテーター間における基準の差異によって、また、１人のアノテーターによってアノテーション作業が行われる場合でもアノテーション作業が長期化することで、基準の一貫性が失われることが考えられる。

アノテーションの一貫性を保つ手法として、アノテーションガイドラインの管理が考えられる。アノテーターは上記のような難しい事例に直面したとき、どのようにアノテーションすべきか判断し、その判断を記録に残す。この判断の記録は全てのアノテーターが参照することができ、この記録が明文化された基準となる。本論文では、このような判断に関する情報をアノテーションガイドラインと呼ぶこととする。一般的に、アノテーションガイドラインは、アノテーションされたコーパスと並行して管理される。しかし、多くのアノテーション作業では、ワードプロセッサーや Wiki などによってガイドラインが管理されているため、このままではガイドラインを体系的に管理することが難しい。

本論文では、アノテーション作業とガイドラインの管理を統合するフレームワークを提案し、体系的にガイドラインを管理できるアノテーションシステムを目指す。現在、さまざまなアノテーションツールが存在する (例: MMAX [1], GATE [2], WordFreak [3], Knowtator [4] など) が、しかし筆者の知りうる限り、アノテーション作業とガイドラインの管理を統合する研究は存在しない。我々は、提案したアノテーションシステムを既存のアノテーションツールのプラグインとして実装した。

2　人手によるアノテーション

提案するフレームワークでは、アノテーション作業を単語列に適切な記述子を割り振る作業と定義する。この定義は、POS タグのアノテーションや固有名詞にタグをつけるアノテーションなど、さまざまなアノテーション作業に対応できる定義である。本論文では、プロテインの固有名詞にタグをつける、プロテインアノテーションを用いて説明を行う。

実際のアノテーション作業において、多くの単語列は容易に記述子を割り振ることができる。しかし度々、どの記述子を割り振ればいいのか、そもそも記述子を割り振ってもいいのだろうか等、困難な問題に直面する。このときアノテーターは、アノテーションガイドラインを参考にし、問題を解決することができる。これがアノテーション作業の一般的な流れになる 。

しかし一般的なアノテーションツールは、単語列を選択し記述子を割り振る機能しか持っていないため、

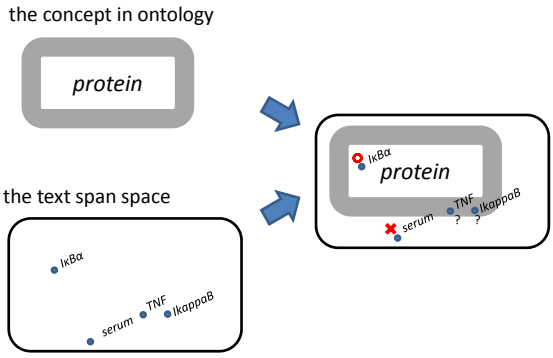

図 1: アノテーション作業の一例

ガイドラインの管理までを行うことができない。我々は、ガイドラインの管理作業を統合したアノテーション作業のフレームワークを提案し、ガイドラインの管理をサポートしたアノテーションシステムを設計する。

ここで、実際のアノテーション作業の例を挙げてみよう。ここでは 4 つの単語列 (*“I B 、”* *“TNF、”* *“I B、”* *“serum.”*) がコーパスに含まれているとする。プロテインアノテーションは、プロテインを示す単語列に記述子 “PROTEIN” を割り振る作業だと定義される。図 1 は、4 つの単語列とプロテインの基準を示す境界線・グレーゾーンを表している。

例えば、単語列 *“I B ”* の場合、アノテーターは容易にプロテインだと判断できる。というのも、*“I B ”* が記述子 “PROTEIN” の基準の内側に入ることが容易に判断できるからである。判断のあと、アノテーターは記述子 “PROTEIN” を *“I B ”* に割り振る。

一般的に、記述子が割り振られた単語列はアノテーションインスタンスとして扱われる。上の例の場合、単語列 *“I B ”* と記述子 “PROTEIN” によって構成されるアノテーションインスタンスが得られたことになる。一方で、単語列 *“serum”* はプロテインではないと容易に判断できる。一般的に、記述子が割り振られない単語列はアノテーションインスタンスとして扱われない。

しかし、単語列 *“TNF”* や単語列 *“I B”* のように、記述子を割り振るべきかどうか判断が難しい単語列が存在する [5][6]。というのも、*“TNF”* や *“I B”* は、単独のプロテインを指し示す単語列ではないが、同じ特性を持つプロテインの集合を表す単語列だからである。このような難しい例では、記述子を割り振るべきと判断するアノテーターもいれば、記述子を割り振る必要がないと判断するアノテーターもいる。そのため、記述子 “PROTEIN” に対してより確かな基準を示すことができる適切なガイドラインが必要になる。

もしそのような適切なガイドラインが見つけられないとき、アノテーターは自分自身でどのようにアノテーションを行うか決めなければならない。しかしこのような場合、そのアノテーター自身あるいは別のアノテーターが同じあるいは類似の事例に直面したとき、同じ判断を行うことができる保証はない。このような問題は、アノテーション作業を行う前に、すべての問題に対応できるだけの十分なガイドラインを用意することができないために生じる。そのため、アノテーターはアノテーション作業中にガイドラインを管理していく必要がある。

具体的には、管理は以下のように行われる。アノテーターが単語列 *“TNF”* のような難しい事例に直面し、適切なガイドラインを見つけられなかったとき、自分自身でどのようにアノテーション行うか判断し、その判断を基にガイドラインを作る。そのガイドラインにはアノテーターの判断の基準が明記され、そのガイドラインを参照すれば同様あるいは類似の事例に直面した時に同じ判断を行えるようにする。

今回の例では、アノテーターは単語列 *“TNF”* に対して記述子 “PROTEIN” を割り振らないこととする。アノテーターは判断の基準を記述し、その基準を示すガイドラインを作成する。ガイドラインはプロテインアノテーションにおいてアノテーションの集合を示す単語列に対して記述子を割り振らないということが書かれる。この新たなガイドラインに従って、アノテーターは単語列 *“TNF”* に対して記述子を割り振らない判断をする。さらに、アノテーターは類似の事例である単語列 *“I B”* にについても、ガイドラインに従って記述子を割り振らない判断を行う。

一般的に、単語列 *“TNF”* のように、ガイドラインの説明にとって有用な具体例がある場合、その具体例をガイドラインの記述に加える。また、ガイドラインの記述と同様あるいは類似の事例に直面した時に容易にこのガイドラインを参照できるようにするため、ガイドラインに (例えば “PROTEIN_FAMILY_OR_GROUP” のような) キーワードを割り振る。しかし、単語列 *“TNF”* に対して記述子が割り振られなかった場合、一般的なアノテーションでは、記述子が割り振られなかった単語列をアノテーションインスタンスとして扱わない。もしこのような単語列をガイドラインの具体例として管理したい場合、その単語列の前後の文や文脈などをメモという形で残すことしかできない。しかし、コーパス上の実際の単語列とガイドラインとが関連付けられるわけではなく、長期間にわたって単語列のコーパス上における場所を覚えておくのは困難である。そのため、具体例としてコーパス上のその単語列を参照することは、容易いことではない。3 章では具体例となる単語列をどのようにガイドラインと関連付けるかについて定義する。

3 ガイドラインの管理を行うアノテーションフレームワーク

アノテーション作業は、生の文章のレイヤーに新たな情報をもつレイヤーを付加する作業とみなすことができる。我々は、生の文章のレイヤーを Text Layer、新た

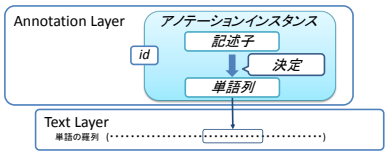

図 2: Annotation Layer におけるデータ構造

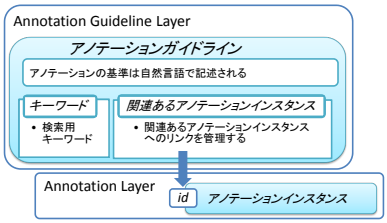

図 3: Annotation Guideline Layer におけるデータ構造

に付加するレイヤーを Annotation Layer と呼ぶこととする。一般的なアノテーションツールでは、Text Layer に Annotation Layer を付加する作業のみをサポートしている。これに対し、我々のアノテーションツールでは Annotation Guideline Layer を付加し、このレイヤーでアノテーションガイドラインを管理することにする。本章では、上記の 3 つのレイヤーについて説明し、ガイドラインの管理作業を統合したアノテーションフレームワークについて提案する。

3.1 Text Layer

本手法では、アノテーションされるテキストは Text Layer によって管理される。テキストは一般的に、文字の羅列によって表現される。

3.2 Annotation Layer

Annotation Layer は、どの単語列に対して記述子を割り振るかについての情報が含まれている。アノテーションインスタンスは、単語列へのリンクと単語列に割り振られた記述子に関する情報を持つ。アノテーションインスタンスによって定義されたリンクが示す単語列はコーパス上に存在するため、Annotation Layer は Text Layer に依存することになる。

2 章で説明したとおり、我々は実際には記述子を割り振られなかった単語列も、ガイドラインの管理には必要だと考えている。しかし、一般的なフレームワークではこのような単語列を管理することはできない。というのも一般的には、アノテーションインスタンスは “単語列” と “記述子” の 2 つの要素によって構成されるからである。そこで我々は、アノテーションインスタンスを “単語列” と “記述子” と “決定” の 3 つの要素によって構成する手法を提案する (図 2)。“決定” という要素を用いることで、記述子を割り振られた単語列だけでなく、記述子を割り振られていない単語列を管理することができるようにした。

3.3 Annotation Guideline Layer

Annotation Guideline Layer はアノテーションガイドラインに関する情報が含まれている。アノテーションガイドラインはアノテーターによって共有され、アノテーター間でのコミュニケーションをサポートを行う。

アノテーションガイドラインは、アノテーションの基準を共有化するために明文化されたものである。一

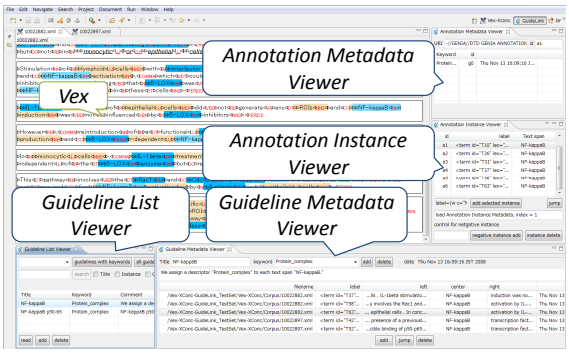

図 4: アノテーションツール *GuideLink*

般的に、人手によるアノテーション作業の際、いくつかのアノテーションインスタンスがガイドラインと関連付けられる。これは提案手法でも同様である。このため、Annotation Guideline Layer は Annotation Layer に依存することになる (図 3)。

4 GuideLink

提案手法を基に、我々はアノテーションガイドラインを管理するツール “*GuideLink (Guideline + Link)*” を実装した。*GuideLink* は既存のアノテーションツールの拡張を行い 、アノテーションガイドラインの管理をサポートする。4.1 章ではガイドラインの参照する方法について、4.2 章ではガイドラインの更新する方法について、4.3 章ではガイドラインに具体例を関連付ける方法ついて説明する。

また、*GuideLink* によってガイドラインに具体例を関連付けるためには、既存のアノテーションツールがいくつかの API を有している必要がある。本論文では、我々は Eclipse[1] プラットフォーム上で動作するアノテーションツールの一つである Vex[2] のプラグインとして、*GuideLink* の実装を行った (図 4)。4.4 章では、アノテーションツールの拡張性について説明する。

[1] http://www.eclipse.org/
[2] http://vex.sourceforge.net/

4.1 ガイドラインの参照

アノテーターが必要とするガイドラインを探し出そうとき、2つの方法が提供されている。1つ目は、必要とするガイドラインに関連付けられているであろうキーワードを基に探し出す方法である。図4の *Guideline List Viewer* がこの機能を提供している。*GuideLink* は木構造のデータ構造によってキーワードを管理し、各ノードはその親ノードのサブクラスのキーワードになっている。例えば、キーワード "PROTEIN" を管理するノードがあり、そのノードの葉ノードとして、キーワード "PROTEIN_FAMILY_OR_GROUP" を管理する葉ノードが存在する。

また2つ目の方法として、アノテーションインスタンスの具体例からガイドラインを検索する方法がある。図4の *Guideline Instance Viewer* や *Annotation Metadata Viewer* がこの機能を提供している。この2つのビューは、各アノテーションインスタンスと関連付けられたアノテーションガイドラインを表示することができる。例えば、記述子 "PROTEIN" についてのアノテーションガイドラインが必要なとき、記述子 "PROTEIN" が割り振られたアノテーションインスタンスを探し出し、そのアノテーションインスタンスと関連付けられたガイドラインを参照すればいいのである。

4.2 ガイドラインの更新

図4の *Guideline Metadata Viewer* は、アノテーションガイドラインの編集を行うためのビューである。図の例では、キーワード "PROTEIN_FAMILY_OR_GROUP" が付けられたガイドラインである。このガイドラインは記述子 "*I B* " が割り振られたアノテーションインスタンスの具体例へのリンクを持っている。アノテーターがガイドラインを参照して、アノテーションインスタンスを作るとき、*GuideLink* はアノテーションインスタンスへのリンクをガイドラインの中に格納する。

4.3 ガイドラインに具体例を関連付ける

アノテーションガイドラインは単語列にアノテーションするときに用いる。もし、我々がある単語列に対して記述子を割り振るかどうか、ガイドラインを用いて決定した時、その単語列、記述子、決定で構成されるアノテーションインスタンスはガイドラインをより明確なものにする具体例となる。そのため、ガイドラインに具体例となるようなアノテーションインスタンスへのリンクを格納し、必要な時にアノテーションインスタンスを参照できるようにしておく。

アノテーションインスタンスへのリンクは、以下のような手順で作られる。まず、既存のアノテーションツールを用いてアノテーション作業を行う。このとき、単語列に記述子を割り振る毎に、既存のアノテーションツールによって単語列に ID を割り振る。*GuideLink* でこの ID を受け取り、ID をアノテーションガイドラインに格納する。

GuideLink では、図4の *Guideline Metadata Viewer*、*Annotation Instance Viewer*、*Annotation Metadata Viewer* によってアノテーションインスタンスとアノテーションガイドラインのリンクを管理する。Guideline Metadata Viewer はアノテーションガイドラインに関連付けられたアノテーションインスタンスを表示し、Annotation Metadata Viewer はアノテーションインスタンスに関連付けられたアノテーションガイドラインを表示する。

4.4 拡張性

Vex は既存のアノテーションツールの一種である。既存のアノテーションツールが *GuideLink* が必要とする機能や API を持つとき、*GuideLink* はアノテーションガイドラインの管理を行う機能を提供することができる。

GuideLink が必要とする機能や API は以下の2つである。(1) 単語列に対して ID を割り振り、単語列からアノテーションインスタンスを作り出す機能。(2) アノテーションインスタンスを作成、編集、削除を行った場合、*GuideLink* へその作成、編集、削除したという情報を送る機能。図4は、*GuideLink* によって Vex の機能を拡張した例である。純粋な Vex そのものでは *GuideLink* が必要とする API を持たないため、本実験のために Vex を拡張し、Vex に必要な API を持たせ、*GuideLink* と接続できるようにした。

5 まとめ

本論文では、アノテーション作業とアノテーションガイドラインの管理を統合したアノテーションシステムを提案し、そのアノテーションシステムに基づいたアノテーションツールを実装した。アノテーションフレームワークの提案を行う前に、我々は人手によるアノテーション作業の流れを説明し、その流れの中でアノテーションガイドラインの管理が必須であることを説明した。アノテーションガイドラインの管理のために、アノテーションに関する3つのレイヤーを定義し、そのレイヤーごとにフレームワークを設計した。最後に、提案されたフレームワークを基に、アノテーションツールの実装を行った。

参考文献

[1] *MMAX: A tool for the annotation of multi-modal corpora*, 2001.

[2] *GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications*, 2002.

[3] *WordFreak: An Open Tool for Linguistic Annotation*, 2003.

[4] *Knowtator: a plug-in for creating training and evaluation data sets for biomedical natural language systems*, 2006.

[5] H. Pearson. Biology's name game. *Nature*, 411(6838):631–632, 2001.

[6] G.A. Petsko. What's in a name. *Genome Biol*, 3(4):1–1005, 2002.