アノテーションガイドラインの管理を行うアノテーションシステムの提案

大内田賢太 † **Jin-Dong Kim**† **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

Abstract

近年、計算言語学の世界では、大量のテキストデータ (コーパス) が蓄積されるようになってきたことから、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、コーパスの人手によるアノテーション作業、アノテーションのガイドラインの管理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。

我々は、アノテーションガイドラインの管理をアノテーション作業自体と同時に行うフレームワークを提案する。アノテーションガイドラインの管理とアノテーション作業を同時に行うことは、いくつかの利点が考えられる。例えば、アノテーションガイドラインを体系的に管理することができ、一貫性の高いアノテーション作業を行うことができるなどである。

我々は、提案したフレームワークの検証のために、AImed コーパスと GENIA コーパスの作成手法を基に、同様のアノテーションしたときのアノテーション手法の手順を示し、その 2 つの手順に沿って実際にどのようなアノテーションガイドラインが作成できるかを示す。また、我々の提案手法を用いた場合と用いなかった場合で、どのような違いが生じるかを検証する。

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。コーパス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。これは、アノテーションされたコーパスをそのままトレーニングデータとして用い、自動的に人の言語知識を獲得することが一般的な手法となってきているからであり、よりエラーの少ないアノテーションされたコーパスが必要になってきているからである。というのも、アノテーションのエラーが自動獲得された言語知識に、直接的に悪影響を及ぼすことが考えられるからである。人手によるアノテーション作業は、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。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ある。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。例えば、品詞を示す記述子を各単語に割り振ったり、固有名詞を示す記述子を文字列に割り振る作業などである。このような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ている。定義された記述子は、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ち、この基準を基に、アノテーターは記述子を単語列に割り振ることになる。もし、その基準があいまいであり、アノテーターが基準を順守できない場合、アノテーションの一貫性が失われる。

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、記述子における定義の影響を受ける。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、アノテーター間における定義の差異によって、一貫性が失われることが考えられる。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、時間が経つにつれて一貫性が失われることが考えられる。

アノテーション作業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ある。アノテーターが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決定し、その決定を記録に残す。この決定の記録は、アノテーター自身、あるいは他のアノテーターが参照することができ、良い指針とすることができる。本論文では、このような決定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、アノテーションガイドラインの管理手順は、あまり研究されていない分野である。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 などによってアノテーションガイドラインが管理されている。そのため、そのようなアノテーションガイドラインは体系的に管理されることは難しい。

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するフレームワークを提案し、体系的にアノテーションガイドラインを管理することを目指す。現在、さまざまなアノテーションツールが存在する (例：WordFreak (Wor, 2003), MMAX (MMA, 2001), Knowtator (Ogr, 2006), GATE (Gat, 2002) 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アノテーション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、このアノテーションシステムの評価のために、実際に我々のアノテーションシステムをプラグインとして実装し、既存のアノテーションツールと結びつけ、アノテーション作業のシミュレーションを行った。

Section 2 では、人手によるアノテーションの流れについて定義し、アノテーションガイドラインの管理に必要なステップを、アノテーションの流れの中に統合する必要があることを示す。Section 3 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とを統合するためのフレームワークを提案する。Section 4 では、プロテインの固有名詞をつけるアノテーション作業を例に、アノテーション作業中にアノテーションガイドラインを作成するシミュレーションを行い、評価を行う。Section 5 では、我々のアノテーションシステムの実装について説明する。Section 6 では、我々のアノテーションシステムを用いたときと用いなかったときで、どのような違いが生じるか比較を行い、我々のアノテーションシステムの利点について議論する。

2 人手によるアノテーションの流れ

我々のアノテーションシステムは、さまざまなアノテーションシステムに用いることができる。本論文では、アノテーションの一例として、プロテインの固有名詞にアノテーションを行うアノテーション作業について考えよう。

Figure 1 は人手によるアノテーションの流れを説明している。一般的にアノテーション作業とは、コーパスと既に定義された記述子を入力し、アノテーションされたコーパスを出力する作業である。入力されたコーパスは、自然言語で書かれたテキストファイルの集合である。そのテキストは、生のテキストであったり、既に品詞などの記述子が付けられているテキストであることもある。基本的にアノテーション作業は、コーパスから単語列を選択し、適切な記述子を単語列に割り振るという作業の繰り返しで行われる。

例として、入力されたコーパスには、以下の4つの単語列が含まれているとしよう：“*IκBα*”・“*IL2R*”・“*IκB*”・“*serum*”。プロテインの固有名詞にアノテーションを行う作業は、記述子 “PROTEIN” をプロテインの一種だと思われる単語列に割り振る作業といえる。Figure 2 は、4つの単語列と、記述子 “PROTEIN” の定義の範囲・ボーダーラインを示している。

これらの単語列にアノテーション作業を行うとき、アノテーターはアノテーション作業に対して適切なアノテーションガイドラインを探すことができる。アノテーションガイドラインには、記述子に対する定義や、どのようにアノテーションすればいいか手法が書かれてある。もし、アノテーターが適切なアノテーションガイドラインを見つけることができたなら、容易にアノテーション作業を行うことができるだろう。

例えば、アノテーターが単語列 “*IκBα*” にアノテーション作業を行おうとしたとする。アノテーターは記述子 “PROTEIN” に関するアノテー

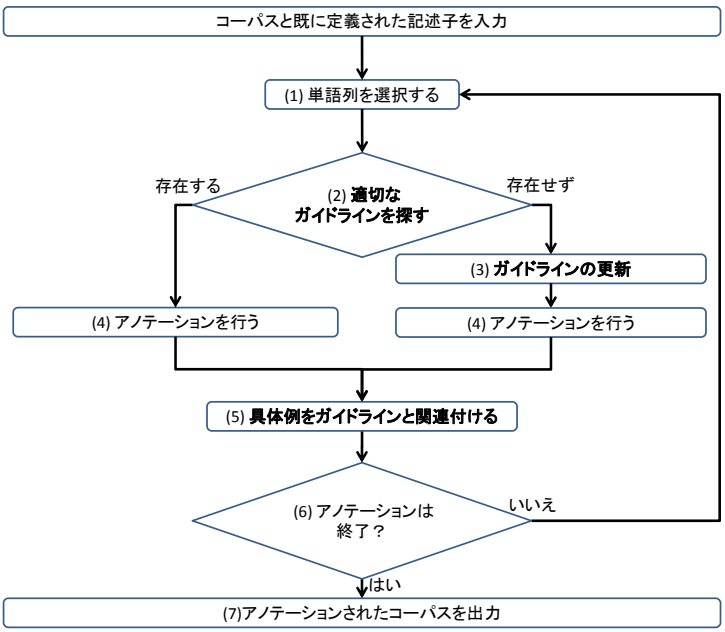

Figure 1: 人手によるアノテーションの流れ

ションガイドラインを探し、記述子 “PROTEIN” の定義を理解することができる。これにより “*IκBα*” は明らかに記述子 “PROTEIN” の定義の内側にいることが分かるので、“*IκBα*” に記述子 “PROTEIN” を割り振ることができる。一般的に、アノテーションされた単語列は、アノテーションインスタンスと呼ばれる。この例では、アノテーターは単語列 “*IκBα*” が記述子 “PROTEIN” によってアノテーションされたというアノテーションインスタンスを得ることになる。同様に、“*serum*” は明らかに記述子 “PROTEIN” の定義の外側にいることが分かるので、アノテーターは “*serum*” に記述子 “PROTEIN” を割り振らない。また、一般的には、このように記述子が割り振られなかった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。

以上のように、明らかに定義の内側･外側であることが判断できる単語列が存在する一方で、判断しにくい単語列も存在する。例えば、“*IL2R*” や “*IκB*” である。“*IL2R*” や “*IκB*” のような単語列は、“PROTEIN” の定義のボーダーライン上に存在する。“*IL2R*” は固有のプロテインを指す単語列ではなく、同じ特性をもったプロテインの集合を指し示す単語列である。“*IκB*” も同様である。このような難しい事例では、“PROTEIN” の定義についてより詳細な情報を持つアノテーションガイドラインが必要になる。しかし、そのようなアノテーションガイドラインが見つからない場合がある。Figure 3(a) は、このような

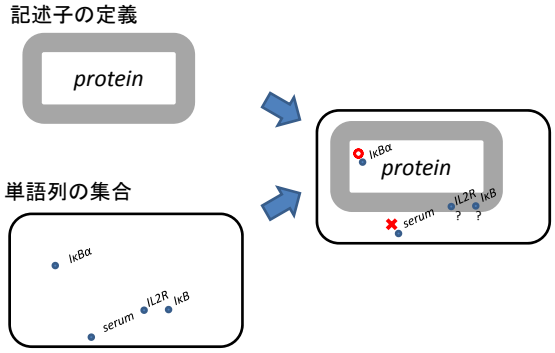

Figure 2: アノテーション作業の例

事例を示している。

また、アノテーターが適切なアノテーションガイドラインを見つけられなかった場合、アノテーター自身が、どのようにアノテーションするかを決める必要がある。しかし、たとえどのような理由で決定したとしても、自分自身あるいは他のアノテーターが同様あるいは類似した状況に直面したとき、同じ決定をするという保証はない。

この問題は、我々がアノテーション作業を行う前に、すべてのボーダーラインの事例に対する十分に適切なアノテーションガイドラインを用意することができないために起こる。もし我々がそのようなアノテーションガイドラインを持っていれば、すべての事例において適切な判断を

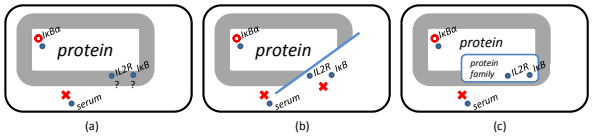

Figure 3: アノテーションガイドラインの直観的な概念

行うことができ、完璧なアノテーションを行うことができるだろう。我々はそのようなアノテーションガイドラインをアノテーション作業の前に用意することができないため、アノテーション作業の過程の中で、アノテーションガイドラインの管理を行う必要がある。

管理は以下のように行われる。例えば *"IL2R,"* のような単語列に出会ったとき、まずは現在のアノテーションガイドラインから適切なガイドラインを探し出す。もし適切なガイドラインが見つかった場合、ガイドラインに従ってアノテーション作業を行う。そうでなければ、アノテーションをいかに行うか決定するための理由となる基準を考える必要がある。そのあと、その基準を用いてアノテーションガイドラインを作成し、既存のアノテーションガイドラインに追加・更新する。これにより、同じあるいは類似の事例に遭遇したとき、同じ決定を行うことができる。

Figure 3(b) は、ボーダーライン上にある単語列 *"IL2R"* に対してアノテーションを行わないと決定した例である。まずは、決定するための基準を記述し、それを新しいアノテーションガイドラインとしてまとめる。この基準は、図上では青いラインで指示している通り、プロテインのグループを示す単語列はアノテーションを行わないというものである。この新しいアノテーションガイドラインに従って、アノテーターは *"IL2R"* はアノテーションしないと決定する。また、類似の事例である *"IκB"* もまた、アノテーションガイドラインに従い、アノテーションしないと決定できる。

逆に、Figure 3(c) は、新たな記述子 "PROTEIN_FAMILY_OR_GROUP" を作り、*"IL2R"* にアノテーションする例である。同様に、*"IκB"* に対してもアノテーションを行う。

いずれの場合においても、*"IL2R"* はアノテーションガイドラインにおいて、とてもよい具体例である。一般的には、アノテーターはこのような具体例をアノテーションガイドラインの説明に加える。また、類似な事例に対してアノテーションすることを考えて、このアノテーションガイドラインに "PROTEIN_FAMILY_OR_GROUP" のようなキーワードを登録し、探しやすくする場合がある。ただし、この *"IL2R"* は実際にはアノテーションされない単語列である。そのため、既存のアノテーションツールでは、*"IL2R"* のようなアノテーションされない文字列を、ガイドラインの具体例として残しておくことが難しい。Section 3 で、我々はこのようなアノテーションされていない文字列を具体例として残す手法について提案する。

3 アノテーションフレームワーク

Section 2 では、人手によるアノテーション作業とアノテーションガイドライン管理の流れを説明した。本章では、アノテーション作業のプロセスとアノテーションガイドライン管理のプロセスを統合するための、2 種類のフレームワークを提案する。1 つ目は手法に関するフレームワークである。これは、アノテーション作業手法、アノテーションガイドライン管理手法の両方に関するフレームワークである。2 つ目はデータ構造に関するフレームワークである。これは、アノテーションインスタンス、アノテーションガイドライン両方に関するフレームワークである。Section 3.1 では、手法に関するフレームワークを、Section 3.2 では、データ構造に関するフレームワークを説明する。

3.1 手法に関するフレームワーク

既存のアノテーションツールはアノテーションインスタンスのみを扱うことが多い。これに対し我々のアノテーションシステムは、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両方を扱う。さらに、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とをリンクで結び、2 つの関連付けを行う。このリンクは、ガイドラインの説明をより詳しくするために用いられる。もし、リンクによってアノテーションガイドラインにアノテーションインスタンスが関連付けられていたら、関連付けられているアノテーションインスタンスがアノテーションガイドラインの良い具体例となるだろう。

また、我々の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される。我々は、より多くの既存のツールで適応できるように、我々のアノテーションシステムでのみがアノテーションガイドラインとリンクを扱うことができることを想定している。そのため、リンクはアノテーションガイドラインからアノテーションインスタンスへ一方方向に指し示すものだと定義する。

Figure 4: アノテーションインスタンスのデータ構造

3.2 データ構造に関するフレームワーク

この章では、我々はアノテーションガイドラインのメタデータについて定義を行い、アノテーションインスタンス・アノテーションガイドライン・アノテーションガイドラインのメタデータの３つのデータ構造の定義を行う。

既存の手法と比べて、われわれのフレームワークは以下の４つの特徴を持っている:

- アノテーションガイドラインが、アノテーション作業の必須な要素として扱われる。
- アノテーションインスタンスが３つの要素で構成される (単語列、記述子、決定) (Figure 4)。
- 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がリンクで結びついている。
- アノテーションガイドラインがそれぞれキーワードが付けられていて、キーワードによるアノテーションガイドラインの検索が行える。

以下の章では、アノテーションインスタンス・アノテーションガイドライン・アノテーションガイドラインのメタデータについて詳しく説明する。

3.2.1 アノテーションインスタンス

Section 2 で、実際にはアノテーションされない単語列が、アノテーションガイドラインの管理には重要になることがあることを示した。対して、多くの既存のフレームワークではアノテーションインスタンスには “決定” は無く、２つの要素で構成されている (単語列、記述子)。しかし、既存のフレームワークでは実際にはアノテーションされてない単語列を扱うことができないため、このようなアノテーションインスタンスをアノテーションガイドラインの管理に用いることができない。我々の手法ではアノテーションインスタンスが３つの要素で構成される (単語列、記述子、決定)。“決定” を用いることで、我々は実際にアノテーションされた単語列だけでなく、実際にはアノテーションされなかった単語列もアノテーションインスタンスとして扱うことができる。

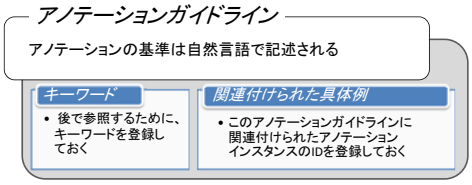

Figure 5: アノテーションガイドラインとメタデータのデータ構造

例えば、プロテインの固有名詞にアノテーションを行い、*“IκB”* と *“IL2R”* に対して記述子を与えなかった場合でも、*“IκB”* と *“IL2R”* は “PROTEIN_FAMILY_OR_GROUP” に関するアノテーションガイドラインにおいて良い具体例とすることができる。しかし、*“serum”* は良い具体例とは言えない。というのも、*“serum”* がプロテインではないと、容易に判断できるからである。我々のフレームワークでは、*“IκBα”* のような単語列に対するアノテーションインスタンスをポジティブアノテーションインスタンスと呼び、*“IκB”* と *“IL2R”* のような単語列に対するアノテーションインスタンスをネガティブアノテーションインスタンスと呼ぶ。我々は、このポジティブ・ネガティブアノテーションインスタンス両方をアノテーションガイドラインに関連づけ、よい具体例として用いることができる。

3.2.2 アノテーションガイドライン

アノテーションガイドラインとは、どのような単語列に対してアノテーションを行うかの基準を記述したものである (Figure 5)。我々のフレームワークでは、アノテーションガイドラインの基準は自然言語で記述することができる。

3.2.3 メタデータ

アノテーションガイドラインは、複数のアノテーションインスタンスと関連付けられ、関連付けられたアノテーションインスタンスによってアノテーションガイドラインの理解を深めることができる。例えば、アノテーションインスタンス *“IκB”* は、キーワード “PROTEIN” や “PROTEIN_FAMILY_OR_GROUP” などが付けられているアノテーションガイドラインと関連付けられるべきだろう。そのため、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は多対多に関連づけられる。我々のフレームワークで

は、この関連づけをアノテーションガイドラインのメタデータとして表現している。

すべてのアノテーションインスタンスには固有の ID が割り振られている。アノテーションガイドラインはメタデータを持ち、メタデータにはアノテーションガイドラインと関連付けられたアノテーションインスタンスの ID が保存されている (Figure 5)。また、後でアノテーションガイドラインを参照するために、必要なアノテーションガイドラインをすばやく検索する手法が必要になる。我々はその手法として、アノテーションガイドラインのメタデータに対してキーワードを割り振り、そのキーワードを管理することで、素早く検索できるようにしている。

4 検証

Section 3 では、アノテーション作業のプロセスとガイドライン管理のプロセスを統合するためのフレームワークを提案した。ここでは、実際にアノテーションガイドラインを作りながらアノテーション作業を行う例を述べ、本手法を用いた場合と用いなかった場合の比較を行う。

例として、生物医学論文に対してプロテインの固有名詞にアノテーションを行ってみよう。この検証では、２つの異なった形式 (AIMed、GENIA) でのアノテーションを行う。この２つの形式を生物医学論文に対して行い得られたコーパスは、広く自然言語処理の世界で用いられている。この二つの形式では用いられたアノテーションガイドラインが互いに異なるため、この二つの形式によって行われた固有名詞アノテーションは、同じ文章に対して異なったアノテーションになる。我々は、この二つの形式によるアノテーションを用いて、我々の手法を用いなかった場合と用いた場合による違いについて検証する。

4.1 AIMed style でのアノテーション

まずは、AIMed 形式でのアノテーション作業を、ステップごとに説明しよう。本検証では初期状態として、１つの記述子 “PROTEIN、” と２つの単語列 *“IL2R,”* と *“IκB”* を持つコーパスが存在するとする (Figure 6 の AIMed の (a))。

Step (1) 単語列を選択する

- ここでは *“IL2R,”* を選択したとする。

Step (2) 適切なアノテーションガイドラインを探す → 存在せず

- *“IL2R”* はプロテインの集合を示す単語列である。
- アノテーターはプロテインの集合を示す単語列へのアノテーションに関するガイドラインを探す。ここでは、適切なガイドラインが存在しないとする。

Step (3) アノテーションガイドラインの更新

- アノテーターは、“プロテインの集合を示す単語列に対してプロテインの固有名詞である記述子をつけない” という基準を作成し、新たなアノテーションガイドラインを作成する。
- この例では、アノテーターは新たな記述子を作成する必要はない。

Step (4) アノテーションを行う

- 新たなアノテーションガイドラインに従い、アノテーターは単語列 *“IL2R”* に対して記述子を割り振らないという決定をする。

Step (5) 具体例とアノテーションガイドラインとを関連付ける

- アノテーターは、このアノテーションインスタンスがアノテーションガイドラインにとって良い具体例だと考えた場合、アノテーションインスタンスに関する情報をアノテーションガイドラインに登録する。
- 後でこのアノテーションガイドラインを参照するために、このアノテーションガイドラインを整理しておく必要がある。このとき、整理の仕方はアノテーターに依存する。

Step (1) 単語列を選択する

- アノテーターがプロテインの集合を示す別の単語列 *“IκB,”* を選択したとする。

Step (2) 適切なアノテーションガイドラインを選択する → 存在する

- 単語列 *“IL2R”* をアノテーションした時に得られたガイドラインが、適切なアノテーションガイドラインとなる。

Step (4) アノテーションを行う

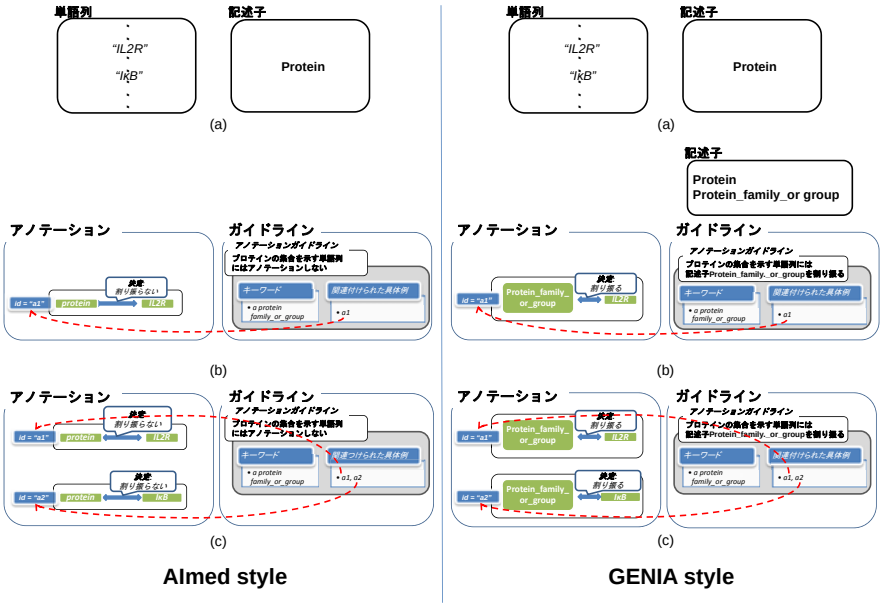

Figure 6: AIMed style と GENIA style によるアノテーションの検証

- 適切なアノテーションガイドラインが存在しているので、アノテーターはガイドラインに従って、記述子 "PROTEIN" を単語列 "*IκB*" に割り振らないことを決める。

Step (5) 具体例とアノテーションガイドラインとを関連付ける

- アノテーターがアノテーションガイドラインに対して、単語列 "*IκB*" をよい具体例だと考えた場合、この単語列に関する情報をアノテーションガイドラインに登録する。

Table 1 は、本手法を用いなかった場合と用いた場合の違いを示している。

Figure 6 の AIMed style の (a) は Figure 1 の (1) を、Figure 6 の AIMed style の (b) は Figure 1(2)-(5) の右側のルートを表している。アノテーターが記述子の概念のボーダーライン上にある単語列 (例えば "*IL2R*") にアノテーションを行わなければいけないとき、アノテーターは単語列をプロテインとしてアノテーションすべきかどうか決定を行わなければいけない。AIMed style の場合、アノテーターは "プロテインの集合を示す単語列に対してアノテーションの固有名詞である記述子をつけない" という基準をもとに決定を行う。Figure 6 の AIMed style の (c) はは Figure 1(1)-(5) の左側のルートを表している。このアノテーションガイドラインに従うことによって、プロテインの集合を表す固有名詞に対して一貫性のあるアノテーションを行うことができるようになる。

4.2 GENIA style でのアノテーション

Figure 6 の右側は、 GENIA style による同様のアノテーション作業を表している。

Step (1) 単語列を選択する

- ここでは "*IL2R*," を選択したとする。

Step (2) 適切なアノテーションガイドラインを探す → 存在せず

- "*IL2R*" はプロテインの集合を示す単語列である。
- アノテーターはプロテインの集合を示す単語列へのアノテーションに関するガイドラインを探す。ここでは、適切なガイドラインが存在しないとする。

Step (3) アノテーションガイドラインの更新

- アノテーターは "プロテインの集合を示す単語列には、記述子 "PROTEIN_FAMILY_OR_GROUP" を割り振る" という基準を作成し新たなアノテーションガイドラインを作成する。

Table 1: 本手法を用いなかった場合と用いた場合の比較

Step in Figure 1	本手法を用いなかった場合	本手法を用いた場合
(2) **適切なアノテーションガイドラインを探す**	一般的に。アノテーションガイドラインはワードプロセッサーや Wiki などで管理される。そのため、適切なアノテーションガイドラインを探す場合、文字列検索などを用いる必要がある。	本手法を用いる場合、ガイドラインは**キーワード**:“PROTEIN_FAMILY_OR_GROUP” を用いて検索することができる。
(3) **アノテーションガイドラインの更新**	アノテーションガイドラインの形式は、アノテーターに依存する。一般的に。アノテーションガイドラインはワードプロセッサーや Wiki などで管理される。	アノテーターは、新しいアノテーションガイドラインを作成し、自然言語で基準を記述し登録する (Section 5.1.2)。
(5) **具体例とアノテーションガイドラインを関連付ける**	アノテーションインスタンスがアノテーションガイドラインにとって良い具体例と考えられる場合、一般的には、アノテーションガイドラインにアノテーションインスタンスの字面・前後の単語・文脈などを書き留めておく。	単語列はアノテーションインスタンスとして扱われ、既存のアノテーションツールによって**固有の ID が割り振られる**。**アノテーションガイドラインは、メタデータによってアノテーションインスタンスの ID を登録する** (Section 5.1.3)。これにより、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は関連づけられる ((b) of AIMed style in Figure 6)。

- このガイドラインに従い、新たな記述子 “PROTEIN_FAMILY_OR_GROUP” を作成する。

Step (4) アノテーションを行う

- 新たなアノテーションガイドラインに従い、アノテーターは *“IL2R”* に対して記述子 “PROTEIN_FAMILY_OR_GROUP” を割り振るという決定をする。

Step (5) 具体例とアノテーションガイドラインとを関連付ける

- アノテーターは、このアノテーションインスタンスがアノテーションガイドラインにとって良い具体例だと考えた場合、アノテーションインスタンスに関する情報をアノテーションガイドラインに登録する。
- 後でこのアノテーションガイドラインを参照するために、このアノテーションガイドラインを整理しておく必要がある。このとき、整理の仕方はアノテーターに依存する。

Step (1) 単語列を選択する

- アノテーターがプロテインの集合を示す別の単語列 *“IκB,”* を選択したとする。

Step (2) 適切なアノテーションガイドラインを選択する → 存在する

- 単語列 *“IL2R”* をアノテーションした時に得られたガイドラインが、適切なアノテーションガイドラインとなる。

Step (4) アノテーションを行う

- 適切なアノテーションガイドラインが存在しているので、アノテーターはガイドラインに従って、記述子 “PROTEIN_FAMILY_OR_GROUP” を単語列 *“IκB”* に割り振ることを決める。

Step (5) 具体例とアノテーションガイドラインとを関連付ける

- アノテーターがアノテーションガイドラインに対して、単語列 *“IκB”* をよい具体例だと考えた場合、この単語列に関する情報をアノテーションガイドラインに登録する。

Figure 6 の GENIA style が、GENIA style によるアノテーションの流れを表している。GENIA style の場合、アノテーターは “プロテインの集合を示す単語列には、記述子 “PROTEIN_FAMILY_OR_GROUP” を割り振る” という基準をもとに決定を行う。GENIA style と AIMed style とでは得られるアノテーションされたコーパスは異なるが、基本的なアノテーション作業の流れは同じである。そのため、アノテーションガイドラインを扱う Step が同じように存在し、既存の手法との違いは Table 1 で示している通りになる。

5 実装

我々のアノテーションシステムを基に、アノテーションインスタンス・アノテーションガイドライン・アノテーションガイドラインのメタデータを扱うためのツール “Annotation Guideline Editor (AGE)” の実装を行った。AGE は既存のアノテーションツールの機能の拡張を行い、既存のツールでは行えなかったアノテーション作業のステップのサポートを行う。

Section 5.1 では、AGE の持つ機能がどのように各ステップのサポートを行うかについて説明する。ここで、一般的なアノテーションの流れを再掲しよう。

(1) 単語列を選択する

(2) 適切なアノテーションガイドラインを探す

(3) ガイドラインの更新

(4) アノテーションを行う

(5) 具体例をガイドラインと関連付ける

(1) と (4) は既存のツールで行われるため、AGE ではサポートしない。AGE では、(2)・(3)・(5) のサポートを行う。(2) については Section 5.1.1 で、(3) については Section 5.1.2 で、(5) については Section 5.1.3 で説明する。

また、(4) については既存のアノテーションツールに大きく依存し、(4) で得られたアノテーションインスタンスを AGE で受け取る API などが必要になる。Section 5.2 では、既存のアノテーションツールとのプラグインとして用いるために必要な要素について説明する。

5.1 AGE の機能

5.1.1 適切なアノテーションガイドラインを探す

Figure 7 は、キーワードを用いてアノテーションガイドラインを検索しているスナップショットである。アノテーションガイドラインを検索するためには、2 つの機能が必要になる。1 つ目はアノテーションガイドラインを特徴づけるようなキーワードを用いた検索手法である AGE では、木構造を用いてキーワードのセットを管理している。木構造の各ノードが 1 つのキーワードを持ち、あるキーワードを持つノードの下のノードには、そのキーワードのサブクラスのキーワードが格納されている。例えば、あるノードにはキーワード “PROTEIN” が格納され、そのノードの下のノードにはキーワード “PROTEIN_FAMILY_OR_GROUP” が格納されている。もし、アノテーターがプロテインに関するアノテーションガイドラインをすべて探し出してほしいなら、キーワード “PROTEIN” や、キーワード “PROTEIN_FAMILY_OR_GROUP” を含むサブクラスのキーワードを用いて検索することになるだろう。

アノテーションガイドラインを探し出すもう一つの方法としては、具体例となったアノテーションインスタンスから辿る方法がある。例えば、記述子 “PROTEIN” に関するアノテーションガイドラインを探したいなら、記述子 “PROTEIN” によってアノテーションされたアノテーションインスタンスを探しだし、その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見つける方法がある。これについては、Section 5.1.3 で説明する。

5.1.2 アノテーションガイドラインの更新

Figure 8 は、AGE を用いてアノテーションガイドラインを更新しているスナップショットである。Figure 8 中の Guideline Metadata Viewer は、

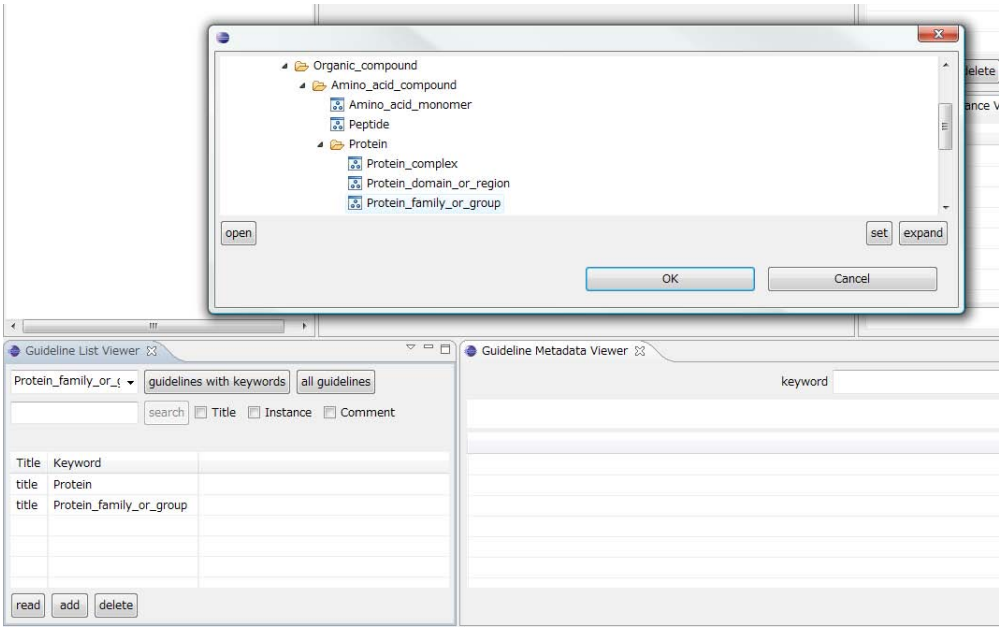

Figure 7: AGE のスナップショット：キーワードによるアノテーションガイドラインの検索

アノテーションガイドラインやメタデータに関する情報を表示する機能を持っている。我々は、Guideline Metadata Viewer を用いて、アノテーションガイドラインやメタデータの追加・編集を行うことができる。

この例では、キーワード “PROTEIN_FAMILY_OR_GROUP” を持つアノテーションガイドラインの編集を行っている。アノテーションガイドラインは、記述子 *“IκBα”* によってアノテーションされたアノテーションインスタンスと関連付けられている。AGE は、アノテーターがアノテーションガイドラインを用いてアノテーションを行うとき、アノテーションガイドラインのメタデータにアノテーションインスタンスとのリンクを保持する機能を持っている。

5.1.3 具体例をアノテーションガイドラインと関連付ける

アノテーションガイドラインは、単語列をアノテーションするときに用いられる。もし、アノテーションガイドラインを用いて単語列に記述子を割り振るか否か、具体例を保存することができれば、この具体例はアノテーションガイドラインの説明をより明確なものにするだろう。我々は、アノテーションガイドラインのメタデータ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する。

アノテーションインスタンスへのリンクは、以下の手法で作成される。まず、既存のアノテーションツールによって、単語列に記述子が割り振られ、アノテーションインスタンスが作られる。このとき、アノテーションインスタンスには固有の ID が割り振られる。AGE はこの ID をアノテーションガイドラインのメタデータに格納することができる。

記述子が割り振られた単語列は、ポジティブアノテーションインスタンスとして扱われる。また、記述子が割り振られなかった単語列も、アノテーションガイドラインにとっては良い具体例になる場合がある。良い具体例となる、記述子が割り振られなかった単語列は、ネガティブアノテーションインスタンスとして扱われる。AGE では両方のアノテーションインスタンスの ID をアノテーションガイドラインのメタデータに格納する。もし、あるアノテーションインスタンスに関連するアノテーションガイドラインが必要なときは、AGE はすべてのアノテーションガイドラインを一度読み込み、その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探し出すことができる。

Figure 9 は、AGE 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関連づけ格納しているスナップショットである。Guideline Metadata Viewer はアノテーションガイドラインに関連付けられているアノテーションインスタンスが表示されている。同様に、An-

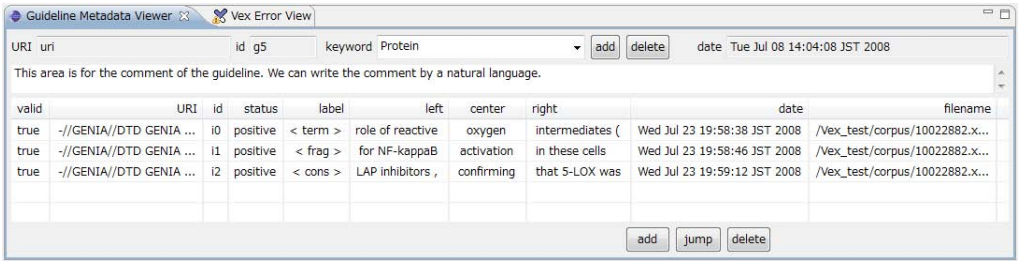

Figure 8: AGE のスナップショット：アノテーションガイドラインの管理

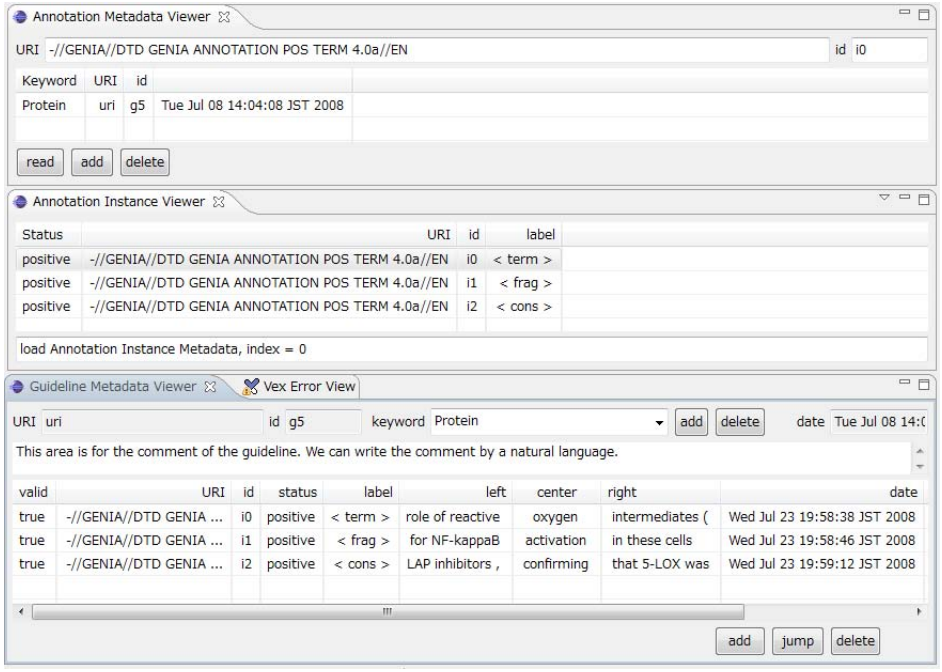

Figure 9: AGE のスナップショット：アノテーションインスタンスとアノテーションガイドラインの関連付け

notation Metadata Viewer はアノテーションインスタンスに関連付けられているアノテーションガイドラインが表示されている。

5.2 拡張性

本論文では、AGE を既存のツールの一つとして Eclipse[1] 上で動くアノテーションツール Vex[2] のプラグインとして実装を行った。Figure 10 は実際に、AGE を Vex のプラグインとして実行したときのスナップショットである。既存のアノテーションツールが AGE の必要としている API を持っている場合、AGE はアノテーションガイドラインの管理を行うことができる。純粋な Vex は AGE が必要としている API を持っていない。今回の実験では、Vex を拡張し、必要とする API を追加して、AGE と接続することにした。

AGE が必要としている API は以下のとおりである:

- 単語列に固有の ID を割り振り、アノテーションインスタンスを作ることができる API
- アノテーションインスタンスを追加・編集・削除した時に、AGE にイベントを伝えるための API

6 既存手法との比較

Section 4 では、我々のアノテーションシステムの検証を行い、Section 5 では、我々のアノテーションシステムの実装を行った。本章では、具体的に、どのような点でわれわれのツールにより改善された点について議論する。

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

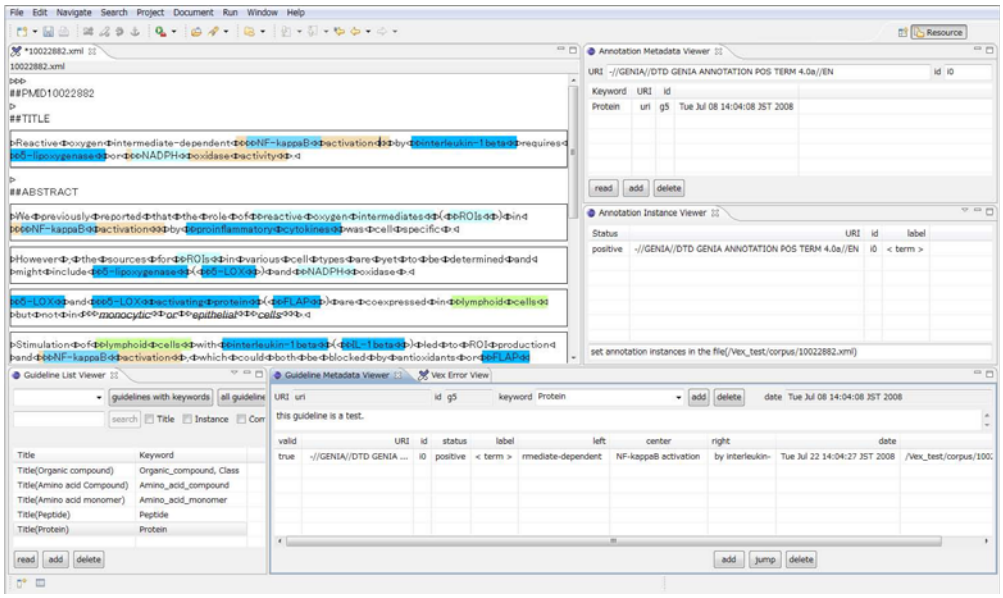

Figure 10: AGE のスナップショット：Vex との接続

われわれのツールは、アノテーションガイドラインの管理を行いながらアノテーション作業を行うことを実現することができる。アノテーションガイドラインは、実際にアノテーションガイドラインを用いてアノテーションされた instance により、より詳しい情報を持つことができる。これは、Word Processor や Wiki などによって管理されたアノテーションガイドラインよりも、より一貫性を保つために適したガイドラインになっている。これにより、より一貫性がとれたアノテーションが行えると考えられる。

また、アノテーションされた単語列が、どのようなアノテーションガイドラインによってアノテーション作業されたかも、簡単にわかるようになる。今までは、GENIA style や AIMed style のように、同じ目的のアノテーションが、異なるガイドラインによってアノテーション作業が行われた場合、得られたコーパスを比較することにより、どのようなアノテーションガイドラインの違いが存在するかを推測するしかなかった。本手法では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がが関連付けられた形で管理されているので、アノテーションガイドラインの比較が容易に行うことができる。

7 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の提案を行い、実際にアノテーションシステムの実装を行った。我々のアノテーションのフレームワークの説明の前に、まず一般的なアノテーションの流れを説明を行い、アノテーションガイドラインの管理の重要性を説明した。アノテーションガイドラインの管理のために、我々は２つのフレームワークを提案した。１つ目は、アノテーションインスタンス・アノテーションガイドライン・アノテーションガイドラインのメタデータのデータ構造について。２つ目は、それらのデータ構造に対する管理手法についてである。これらのフレームワークを基に、我々の手法を用いた場合と用いなかった場合のアノテーションの流れの検証を行い、その差を比較した。また、我々のアノテーションシステムをどのように実装できるか説明を行い、実際に実装を行った。

References

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

2001. *MMAX: A tool for the annotation of multi-modal corpora*.

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

2003. *WordFreak: An Open Tool for Linguistic Annotation*.