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

東京大学　辻井研究室　D3　大内田賢太

はじめに

テキストアノテーション

定義

・テキストデータに対して、人間の言語知識を用いたラベルをつけていく作業

目的

・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、テキストアノテーションされたコーパスから言語知識を得る手法が一般的に行われている

テキストアノテーションにおける問題点

人手によるアノテーションにおける問題点

▪ 時間がかかる・多くの人数が必要であり、巨大なテキストデータを、同じ基準でアノテーションするのは困難

一貫性の無いアノテーションになってしまう問題点

・複数のアノテーターによる、一貫性の喪失　(*inter-annotator discrepancy*)

・同一のアノテーターによる、一貫性の喪失　(*intra-annotator discrepancy*)

アノテーションガイドライン

定義

・判断が難しい場合に手助けとなるガイドライン。一般的には、いくつかの例を列挙したリストになっている

目的

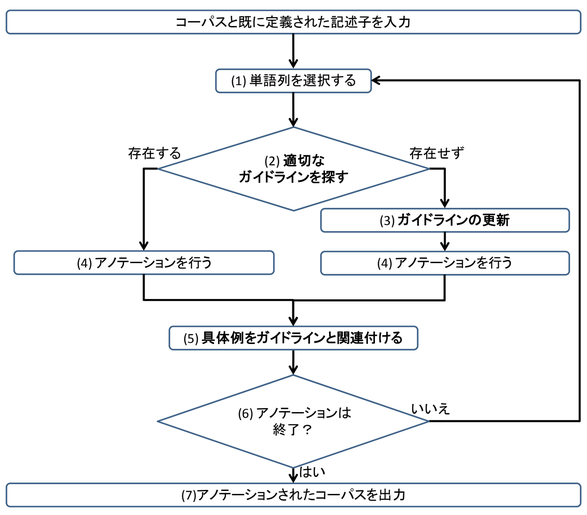
・個々のアノテーターに偏ったアノテーションを防ぎ、一貫性の高いアノテーションを行えるようになる

問題点

・アノテーションを行う前から、アノテーション上のすべての問題を想定することは困難

アノテーション作業を行いながら、同時に
アノテーション・ガイドラインの管理する手法を提案する

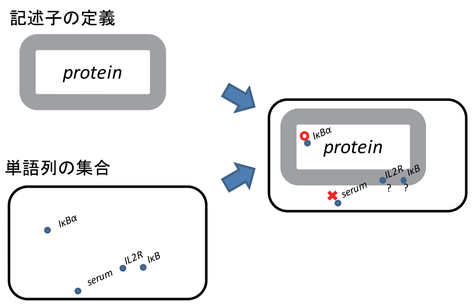
人手によるアノテーションの流れ



主なアノテーションの流れは左の図のようになる。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。詳しいアノテーションの流れと具体例は、右の検証で説明することにしよう。

実際のアノテーション作業にかかわるステップは(1)・(4)・(6)であり、アノテーションのガイドラインに関する部分は(2)・(3)・(5)である。

記述子の定義



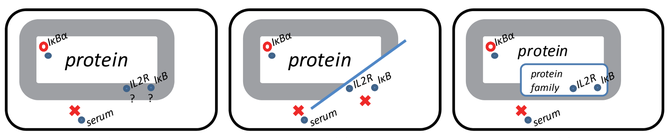
ここでは、実際のアノテーション作業を図で表現してみる。例として、入力されたコーパスには、以下の４つの単語列が含まれているとしよう：*“IκBα”* *“IL2R”* *“IκB”* *“serum”*。

この例では、選択した単語列がプロテインの固有名詞である場合、記述子“Protein”を単語列に割り振る作業とする。左の図は、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。定義から

・*“IκBα”* はプロテインの一種

・*“serum”* はプロテインの一種ではない

ということは明らかにわかる。しかし *“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。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。



・(a)は、*“IL2R”* や *“IκB”* のような単語列が“Protein”の定義のボーダーライン上に存在することを表している。
・(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
・(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

いずれの場合でも、決定するまでの基準を作り、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。
図中では、新しいアノテーションガイドラインを青い線であらわしている。

また、新しいアノテーションガイドラインにおいて *“IL2R”* はとてもよい具体例である。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ため、既存のアノテーションツールでガイドラインの具体例として残すことは難しい。我々はアノテーションされてない文字列を具体例として残す手法について提案する。

アノテーションフレームワークの提案

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。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は、２つの要素で構成されている(*単語列*、*記述子*)。
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*)。
- ・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。

アノテーションインスタンス

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”* は無い。そのため、既存の多く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ので、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。

アノテーションガイドライン

アノテーションの基準は自然言語で記述される

キーワード
- 後で参照するために、キーワードを登録しておく

関連付けられた具体例
- この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のIDを登録しておく

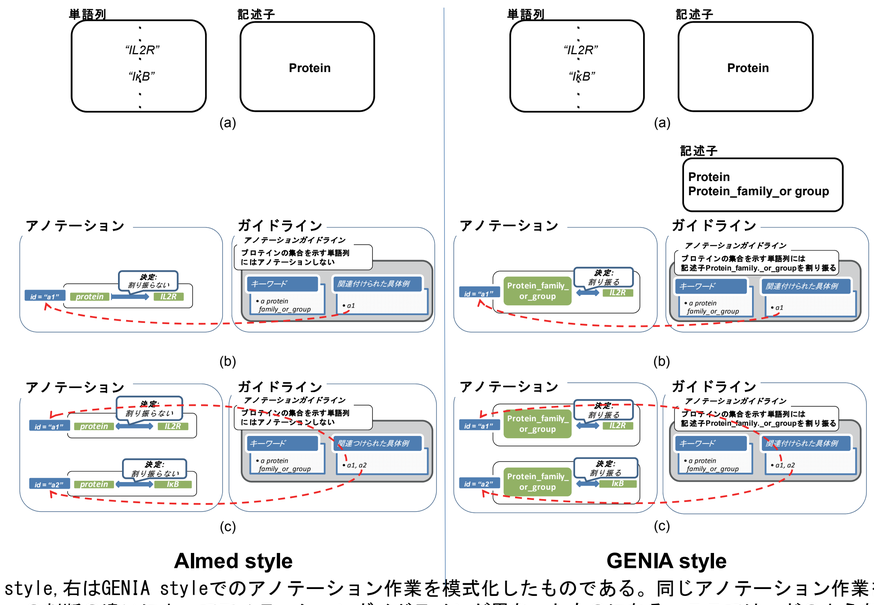
我々のフレームワークでは、アノテーションガイドラインの基準は自然言語で記述することができる。アノテーションガイドラインは、複数のアノテーションインスタンスと関連付けられ、関連付けられたアノテーションインスタンスによってアノテーションガイドラインの理解を深めることができる。そのため、アノテーションインスタンスとアノテーションガイドラインは多対多に関連づけられる。

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。

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

検証

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として、それぞれ別々のアノテーションガイドラインが得られるアノテーション作業を行う。これにより、さまざまなアノテーションガイドラインが得られるアノテーション作業に本手法を用いることができる検証を行う。



左はAIMed style,右はGENIA styleでのアノテーション作業を模式化したものである。同じアノテーション作業を行っているが、アノテーターの判断の違いによってアノテーションガイドラインが異なったものになる。ここでは、どのようなアノテーターの判断が行われたとしても、本手法によってアノテーション作業が正しく行えることを検証する。「人手によるアノテーションの流れ」で説明した手順で、アノテーション作業を行ってみよう。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

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) 適切なガイドラインが存在せず
“IL2R” はプロテインの集合を示す単語列であるのでプロテインの集合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、存在しないとする。

Step (3) ガイドラインの更新
<u>アノテーターは、“プロテインの集合を示す単語列に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。</u>

Step (4) アノテーションを行う
<u>新たなガイドラインに従い、アノテータは単語列 *“IL2R”* に対して記述子を割り振らないという決　定をする。</u>

Step (5) 具体例とガイドラインと関連付ける
アノテーターは、このインスタンスがガイドラインにとって良い具体例だと考えた場合、インスタンスに関する情報をガイドラインに登録する。

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) 適切なガイドラインが存在する
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、適切なガイドラインとなる。

Step (4) アノテーションを行う
<u>適切な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。</u>

Step (5) 具体例とガイドラインを関連付ける
アノテーターが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ガイドラインに登録する。

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) 適切なガイドラインが存在せず
“IL2R” はプロテインの集合を示す単語列であるのでプロテインの集合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、存在しないとする。

Step (3) ガイドラインの更新
<u>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”という基準を作り新たなガイドラインを作成する。このガイドラインに従い、記述子“PROTEIN FAMILY OR GROUP”を作成する。</u>

Step (4) アノテーションを行う
<u>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。</u>

Step (5) 具体例とガイドラインを関連付ける
アノテーターは、このインスタンスがガイドラインにとって良い具体例だと考えた場合、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。

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) 適切なガイドラインが存在する
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、適切なガイドラインとなる。

Step (4) アノテーションを行う
<u>適切な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。</u>

Step (5) 具体例とガイドラインを関連付ける
アノテーターが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ガイドラインに登録する。

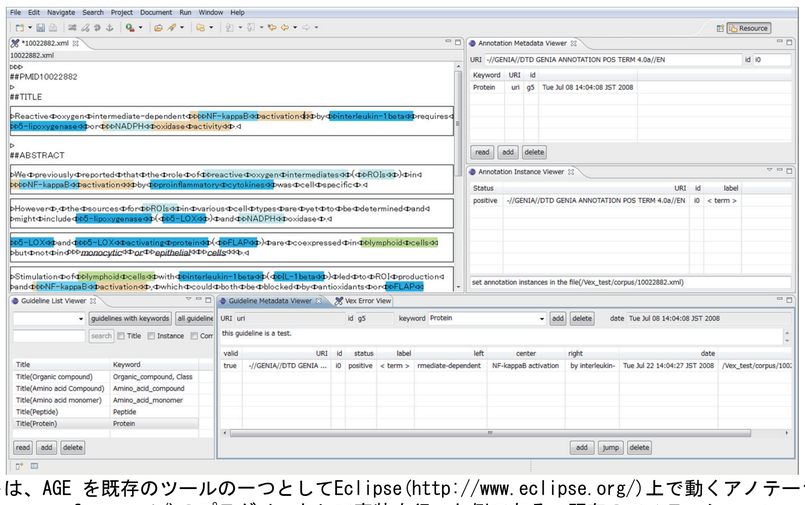
比較

上記の検証を基に、Aimed styleとGENIA styleでのアノテーションが、本手法を用いなかった場合と用いた場合で、どのような違いが出るかを比較する。

	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
(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。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。

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・ガイドライン・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。



上のスナップショットは、AGE を既存のツールの一つとしてEclipse(http://www.eclipse.org/)上で動くアノテーションツールVex(http://vex.sourceforge.net/)のプラグインとして実装を行った例である。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