

# アノテーションを用いて Web ドキュメントを分かりやすく提示する方法

東中 竜一郎  
慶應義塾大学  
政策・メディア研究科

長尾 確  
日本アイ・ビー・エム (株)  
東京基礎研究所

## 概要

アノテーションとは、コンテンツに対するメタコンテンツであり、XML 形式のデータとして表現される。本稿ではアノテーションを持つテキストと言語構造が付加された辞書をプロキシを介し利用することにより Web 上のテキストをより分かりやすくする方法について述べる。現在、Web 上で様々な文書を読むことができるが、アクセスするユーザにとって必ずしも理解しやすい文書ばかりとは言えない。この問題に対処する手段として、テキスト内の難解な言葉を、構造化辞書を使って分かりやすく加工し提示する手法を提案する。具体的には、辞書の定義文のポップアップ表示や原文への挿入、グロッサリー（用語集）、原文の言い換えの自動作成を実現した。

## 1 はじめに

Web 上にさまざまな文書が存在するがそれらは多様な内容と形式を持つために文書によっては背景知識を持たない人にとって分かりにくかったり、意味が取れなかったりすることがよく起こる。例えば専門用語であったり、一部でだけ通用するような単語を含む文書であると他の分野に携わっている人にとっては非常に難解なものになる。

これでは Web の発達によって様々な文書にアクセスできる状況になったとはいえ、本当に文書が共有されているとは言えない。誰にとってもある程度理解可能な文書を提供することが可能になれば Web の価値はより高まるのではないかと考える。難解な語を含む文書に出会った場合、通例辞書を引いたり、知人にその意味を尋ねたりして解決する。しかし、時間的制約などを考えると、分からない単語に出会う度にそのようなことを行うのは効率が悪い。そのため、可能な限り辞書引きを自動化し、その結果をユーザに分かりやすく提示する方法を今回提案する。

例えば、翻訳ソフトなどでよくマウス辞書が使用される。わからない単語の上にマウスポインタを置くとその単語の意味がポップアップして表示されるというものであるが、従来のマウス辞書では語義が解決できておらず、複数語義のある単語を調べた場合、すべての語義を表示してしまう。これではどの意味にその単語が使われているのかが明解でない。また使用する辞書はネットワーク上で共有され、ダイナミックに語彙

を増やすことのできるような仕組みを持つことが、辞書を充実させる上で必要である。

単語の語義を自動的に辞書引きできた場合、次のステップはその提示方法である。上に挙げた定義文の「ポップアップ」はその1つであるが、他に「挿入」と「グロッサリー（用語集）」、「言い換え」を考えた。挿入は定義文を文中に埋め込む方法で、言い換えは原文の表現を辞書に基づいて変換する方法、グロッサリーは文書に出現する難解な用語をその説明と共に、別ウィンドウに表示する方法である。言い換えは意味的な情報を必要とし一般に実現が困難である。さらに、原文のどの部分がユーザにとって難解であり、分かりやすく変換されるべきかを決定することは大きな問題である。

本稿ではこれらへの解決策として、アノテーションと呼ばれる付加情報を利用して上記問題点を解決するシステムを提案する。2章で Web におけるこのシステムの位置づけとアノテーションについて解説し、3章では構造化辞書を利用した定義文のポップアップや挿入、言い換え、グロッサリーについて説明し、Web 上の辞書に語彙を登録するシステムについても述べる。4章でまとめと今後の課題を述べる。

## 2 アノテーション

アノテーションは、コンテンツに対するメタコンテンツであり、XML (eXtensible Markup Language) 形式のデータとして表現される [14]。もちろん、アノテーションに対するアノテーションも考えられるが、これ

はまだ実現されていない。われわれは任意の HTML ドキュメントの任意のエレメントにアノテーションの XML データを関連付ける仕組みを開発した [1]。具体的には、XPath[15] と呼ばれる手法を用いて HTML エレメントを指定し、アノテーションデータのファイル名を関連付けるテーブルを用意する。

アノテーションには数種類あり、今回はテキストの言語構造に関するアノテーション(言語的アノテーション)を主に使用する。言語的アノテーションの利用によって、テキスト内のある単語に対し名詞や動詞といった品詞情報を付加することができ、日本語テキストの単語の境界を機械的に認識するなどの様々なことが実現可能となる。さらに属性として語義を付与すれば、辞書引きに利用できる。本稿ではこの点に着目している。アノテーションによるアプローチは文書の情報をよりリッチにすることによって、言語の高度な機械的処理を早期に実現する。

## 2.1 アノテーションの作成と管理

アノテーションの作成と管理のために、われわれはアノテーションエディターというオーサリングツールと、アノテーションサーバーを開発した。

アノテーション環境は図 1 のようになっている。

たとえば、HTML ファイルの場合、次のようなプロセスでアノテーションが作成され、管理される。

1. アノテーターと呼ばれるユーザーはアノテーションエディターを起動して、対象となるドキュメントの URL を入力する。
2. アノテーションサーバーはエディターから URL を受け取ると、Web サーバーに問い合わせる。
3. アノテーションサーバーは Web サーバーからドキュメントを受け取る。
4. アノテーションサーバーはドキュメントの DOM ハッシュ値を計算すると、その値と URL をデータベースに登録する。
5. サーバーは、ドキュメントをエディターに送る。
6. ユーザーはエディターを使ってアノテーションを作成すると、それを自分のプロファイル情報(名前と専門分野など)と共にアノテーションサーバーに送信する。

7. サーバーは受け取ったアノテーションデータを URL と関連付けてデータベースに登録する。
8. サーバーは同時にアノテーターのプロファイル情報を更新する。

## 2.2 アノテーションエディター

アノテーションエディターは Java アプリケーションとして実装されており、アノテーションサーバーと通信できるようになっている。

アノテーションエディターは以下の機能を持っている。

1. URL を用いてアノテーションの対象となるドキュメントをサーバーに登録する。
2. Web ブラウザーと連動して、ドキュメントの任意のエレメントを選択できる。
3. XML 形式のアノテーションデータを生成し、サーバーに伝達する。
4. コンテンツが更新されたときに、以前に作成したアノテーションを再利用できる。

エディター(図 2)の左側のウィンドウは、HTML ファイルの内部構造を示している。Web ブラウザー上で任意の HTML エレメントを選択すると、その部分のテキストがエディターに渡され、エディターの右側のウィンドウに表示される。選択された部分には自動的に XPath と呼ばれるエレメントの ID が付与される。

このエディターを用いて、ユーザーは言語構造(構文や意味に関する構造)をテキストに関連付けたり、ドキュメント内の任意のエレメントにコメントを付けたりすることができる。

テキストの言語構造に関するアノテーションを言語的アノテーションと呼ぶ。言語構造は、まず Juman[7]、KNP[8]を用いて自動的に生成されるが、その構造に誤りが含まれる場合は、それをインタラクティブに解消することができる。言語構造を修正するために、自動的に解析された構造を分かりやすく表示するための工夫を行っている。言語的アノテーションは図 2 の右に表示されている画面上の操作によって容易に修正できる。テキストは依存関係のツリーグラフとして表されており、依存関係を修正する場合はドラッグアンドドロップで行うことができる。

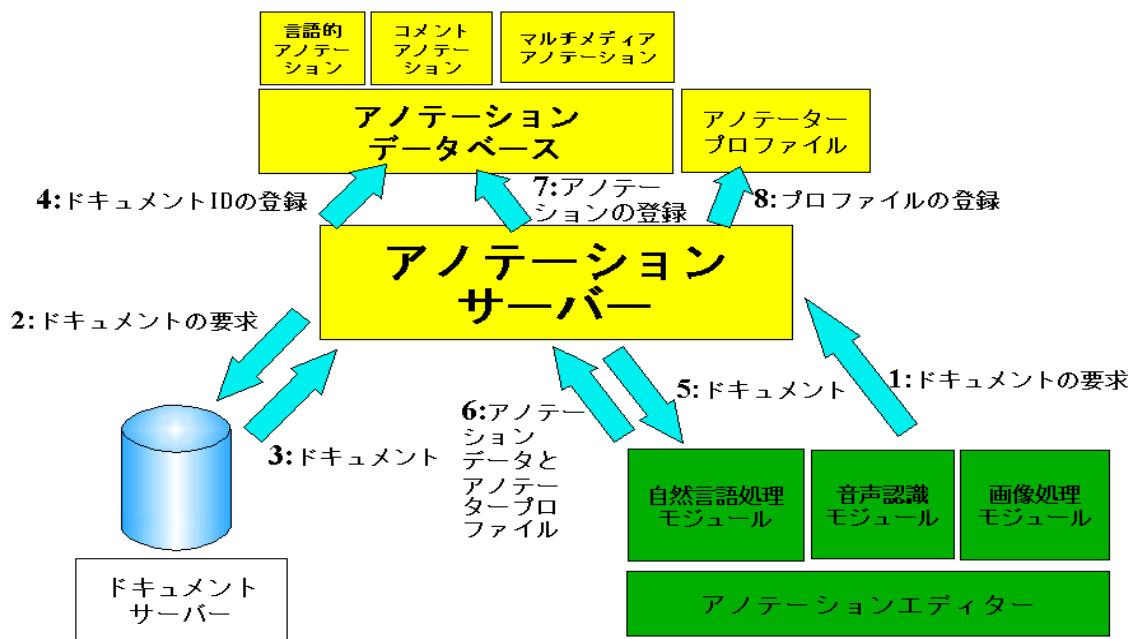


図 1: アノテーション環境

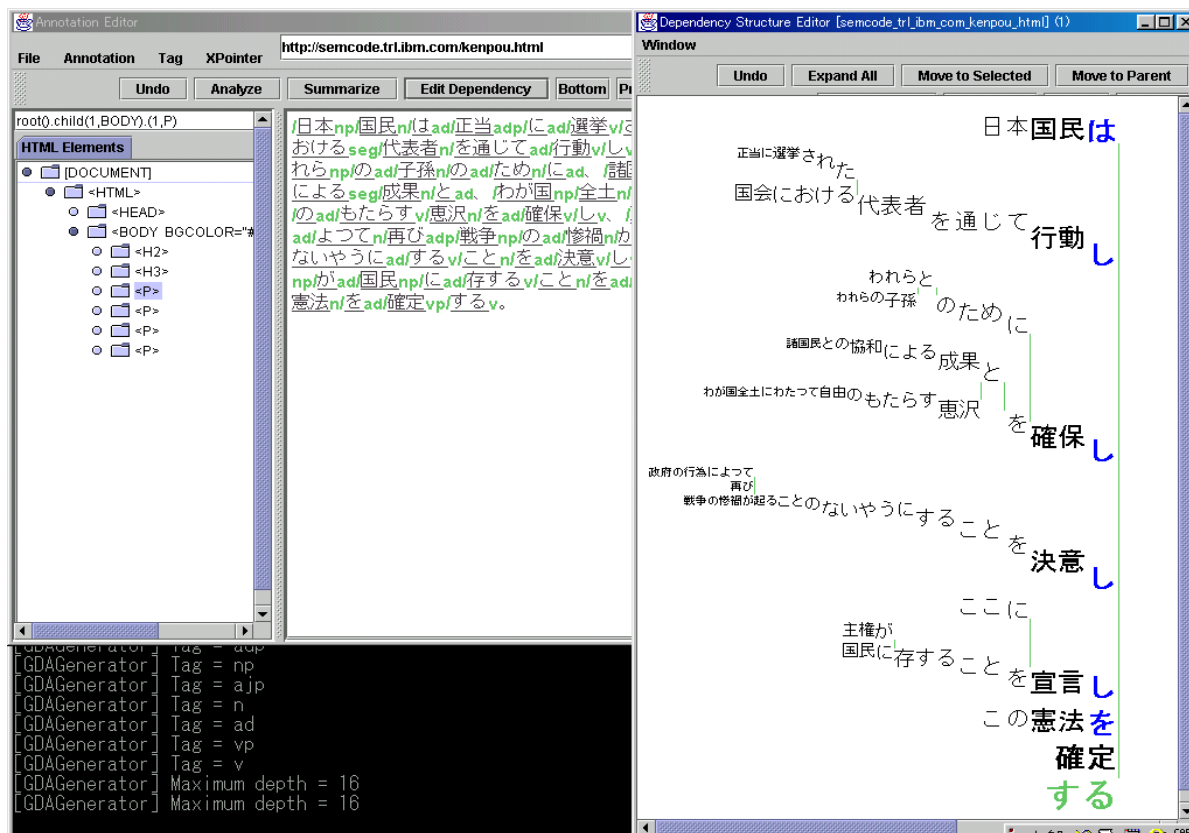


図 2: アノテーションエディターの画面

## 2.3 言語的アノテーション

言語的アノテーションは、ドキュメント内テキストエレメント (<H\*>, <P>, <OL>, <UL>, <DL>など) の文章の意味構造に関するアノテーションである。それは、語間の係り受け、代名詞の指示対象、多義語の意味など、かなり細かい情報を含む。このタイプのアノテーションは、ドキュメントの内容理解に大きく貢献し、文書変換以外にも、たとえば、内容検索や知識発見などに利用される。

言語的アノテーションは、具体的には XML 形式のタグファイルである。タグセットには、電総研の橋田らの提唱する GDA (Global Document Annotation) [11] のものを用いている。GDA は多言語間に共通な意味的・語用論的タグをドキュメントに付与することにより、その機械的な内容理解を可能にし、ドキュメントの検索・要約・翻訳を実用的なレベルで実現するとともに、ドキュメントの作成・公開(共有化)・再利用を考慮した統合的なプラットフォームを構築して、世界的に普及させようという、壮大なプロジェクトである。われわれのプロジェクトは GDA を現在の Web のアーキテクチャ上で利用可能にし、さまざまなサービスと連動させることによって、GDA の思想をより具体的な形で浸透させようとする試みの一つと位置付けられる。

一般に、GDA ドキュメントはネットワーク構造を成しており、そのリンクには、タグの入れ子構造によって定義される関係と参照関係の 2 種類がある。また、GDA のタグ集合は 10 項目以上からなるが、さしあたり、そのうちで自動タグ付け作業が比較的大変だと思われる、統語構造、文法・意味関係、語義、照応、修辞関係という 5 項目だけを扱っている。GDA タグセットの詳細については、<http://www.etl.go.jp/etl/nl/gda/> を参照のこと。

このようなタグ付けは多くの労力を要すると思われるが、アノテーションエディターにいくつかの自然言語処理モジュール(統語・意味解析、照応解析など)を統合することによって、極力人間の負担を減らせるように工夫している。人間がインタラクティブに解析した部分は、事例として次の機会に再利用されるので、それによって解析の精度が少しずつ上がっていくことになる。解析の精度が上がれば、それだけ人間の負担が減ると思われるので、将来的にはタグ付けのコストは十分に少なくなることが予想される。

## 3 自動辞書引き

### 3.1 システム

前章で述べたアノテーションの枠組みの中にわれわれは辞書を参照し Web ページのテキストを分かりやすく変換する機構を盛り込んだ。システムは図 3 のようになる。

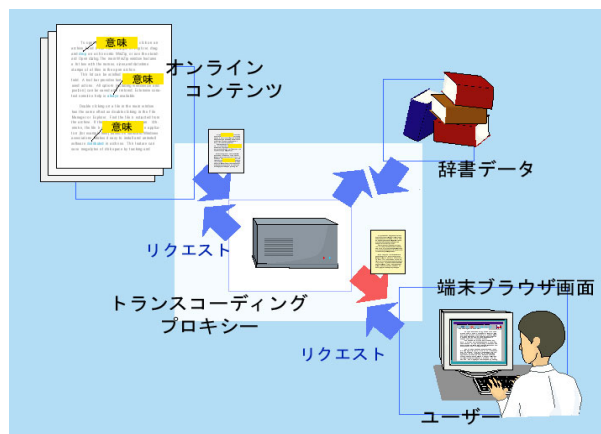


図 3: システム図

このシステムは以下のように動作する。

1. ユーザはドキュメントの取得をプロキシに要求する。
2. プロキシは語義アノテーションを持つドキュメントを取得する。
3. プロキシは辞書サーバにアクセスし、アノテーションとユーザの専門分野などの個人情報に従って辞書引きを行う。
4. プロキシは辞書の定義文を用いて文書を変更する。
5. プロキシは新たに生成された文書をユーザに提示する。

### 3.2 辞書と語義アノテーション

プロキシはアノテーションに従い単語の意味を解決する。そのために必要となる辞書と語義アノテーションに付いて述べる。

一般的に国語辞典は以下のようなフォーマットで作られている。

見出し語	
定義文	用例
定義文	用例
定義文	用例...
etc...	

以上に示すように1つの見出し語に対して定義文が複数個あることは多く、実際の例<sup>1</sup>を以下に示す。

アイス	1 氷
	2 アイスコーヒー
	3 アイスクリーム

この例では「アイス」という単語に3つの語義が与えられている。このように機械的に辞書を引く場合、見出し語だけでは、どの意味で使われているのかという問題が解消しない。解決策として単語に対しその語義をアノテーションとして付加することにする。具体的には語義に対し一意の識別子を付与し、識別子と定義文の対を辞書とする。実装としてEDR 日本語単語辞書[9]の概念識別子、定義文としては同辞書の定義文を用いた。

語義アノテーションはアノテーションエディターを用いて行われる。ユーザが、語義を明確にしたい単語を選択すると、語義の候補がリストで提示される。「アイス」の例では候補として3つ挙がることになり、ユーザはどれかを選択する。図4は語義アノテーションと辞書の例である。

語義アノテーションの例
<np sense="3c2c24"> 読み="アイス" 品詞="名詞-一般"> アイス</np> (3c2c24 が概念識別子)
辞書 <sup>2</sup> の例
概念識別子 定義文 ; 辞書見出し [読み]; ... 3c2c24 コーヒーを冷やした飲み物; アイス [あいす]; アイスコーヒー [あいすこーひー];

図4: 語義アノテーションと辞書の例 (この場合「アイス」は「コーヒーを冷やした飲み物」という意味になる。)

<sup>1</sup> 三省堂小学国語辞典 [10] による。

<sup>2</sup> ただし、辞書を用いた言い換えを行うときは定義文に言語的アノテーションを付ける。

### 3.2.1 辞書の動的作成

現在、EDR 日本語単語辞書を利用しているが専門用語や、人名などの固有名詞は含まれていない。さらに、そういった単語は時間とともに量を増して行く。そのための対処としてオンラインに存在する様々な辞書の利用に加え、アノテーションエディター上でユーザによる辞書項目の追加を考案した。この仕組みにより、新しい言葉にも対応することが可能になり、辞書データを、同じプロキシを利用するすべての人で共有することができる。

処理の流れは以下のようになる。

1. ユーザが語義を付けたい単語を選択する。
2. ユーザが定義文を入力する。(言語的アノテーションも付加する)
3. 単語の属性に newsense という属性が入り、値としてユーザの入力した語義がセットされる。
4. アノテーションサーバにアノテーションが送られる。
5. アノテーションサーバは newsense 属性を見つけると、新しい語義に関して、その新規の概念識別子、定義文、アノテーター名を辞書に登録する。

## 3.3 語義の提示

ある単語に対して語義が決定された場合、その定義文をユーザにどのように提示するかということは大きな問題であり、その手法によって最終的な理解は大きく左右される。本稿ではポップアップ、挿入、グロッサリー、言い換えの手法を提案し、実装した。

### 3.3.1 ポップアップ

最も単純な定義文の提示の手法としてポップアップがある。これは語義を知りたい単語をマウスでポイントすると(語義がアノテートされていれば)、その定義文がポップアップウインドウに表示されるものである。従来のマウス辞書では、ある単語を調べるとその単語の持つすべての語義を提示してしまうが、語義アノテーションによって、その文脈で適切な語義のみが表示されるため、理解が促進される。

原文テキストは変化しないため、元のレイアウトを保持できる。定義文を参照できる単語に関してはテキ

ストの背景色をわずかに変え、それとわかるようにした。この手法では、ユーザが自然に知らない単語の意味にアクセスできる。図5は画面例である。

### <東証>テレコムがもみ合い。

(9時25分)もみ合い。買気配で始まり、前日比10万円高の483万円まで上昇したが、前日まで5日続伸しただけに、さすがに利益確定売りが出ているようだ。市場では「J-フォンの黒字転換など明るい材料も多い。資金が一部NTTやNTTドコモから流れてきている可能性がある」(中堅証券の情報担当者)との声も聞かれた。

売りと買いが両方あり、小幅の値動きを繰り返す状態。

図5: 語義のポップアップ

### 3.3.2 挿入

システムは語義の付与された単語を見つけるとその直後に括弧付きでその定義文を挿入することもできる。括弧の中の定義文は薄い色で表示され、元の文と区別できるようになっている。

この手法は文中に埋め込むため一般に文が長くなり、文章の理解を妨げる恐れがある。図6は画面例である。

### <東証(東京証券取引所)>テレコム(テレコミュニケーション株)がもみ合い(売りと買いが両方あり、小幅の値動きを繰り返す状態)。

(9時25分)もみ合い(売りと買いが両方あり、小幅の値動きを繰り返す状態)。買気配(好材料などで買い人気が一時的に強く、商い不成立で値がつかない状態)で始まり、前日比10万円高の483万円まで上昇したが、前日まで5日続伸(株や商品取引などの相場が引

図6: 語義の挿入

### 3.3.3 グロッサリー(用語集)

グロッサリーとは一般に単語とその語義の対をページの末尾などに付加するものである。今回は実装として、ページを表示しているウインドウの他に新しいウインドウをもう一つ作成し、その中にグロッサリーを表示した。

グロッサリーに説明が含まれる単語は下線が引かれ、背景色が変化しているためそれとわかるようになっており、クリックされるとグロッサリーウインドウ中の説明の部分がフォーカスされる。また、グロッサリーウインドウ中の見出し語を選択すると、本文における

対応する単語がハイライトされる。

ポップアップとは違い文章中のすべての用語およびその説明を概観することが可能であり、全体的なイメージを把握しやすい。挿入との違いは、説明が文中に埋め込まれないため、文章の流れを理解しやすい点である。図7は画面例である。

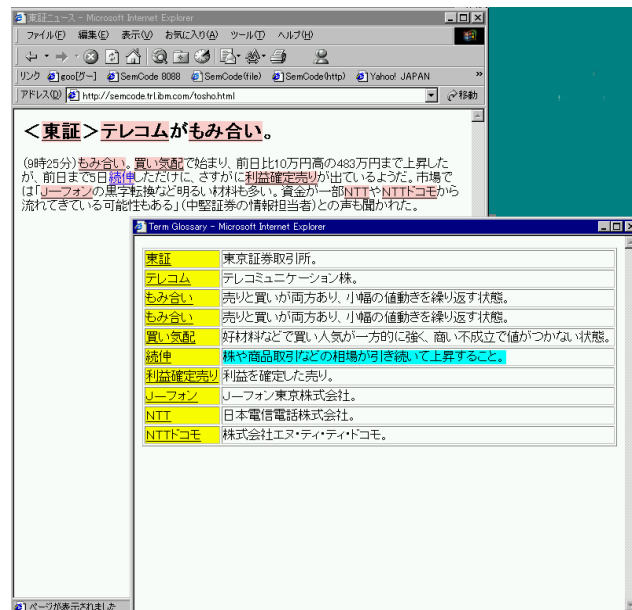


図7: グロッサリー

### 3.3.4 言い換え

語義の付与された単語を定義文を利用し言い換えることもできる。言い換えを行うことによってポップアップに必要なマウスの移動や、挿入に必要な視線の煩雑な移動が要らなくなり難解な語の説明を最も認知的に負担の少ない形で提示することができる。

重要な点としては、文章を音声化した場合、単語の説明が文章の流れを止めることがないため、内容の理解を妨げにくく、最も自然な形でユーザに内容を伝えることが可能である。

言い換えられた部分は背景色を少し変化させて表示されるため、それと分かるようになっており、マウスでクリックすることにより言い換え前の語を知ることが可能である。この機能によってユーザは新たな語を学習することができる。さらに、次回以降、同じ語義の単語を含む文書を提示する場合に言い換えを行うかどうかも設定できる(図8)。

ところで、単語を定義文でそのまま置き換えたのは文との親和性が良くない。理由として以下の2点が

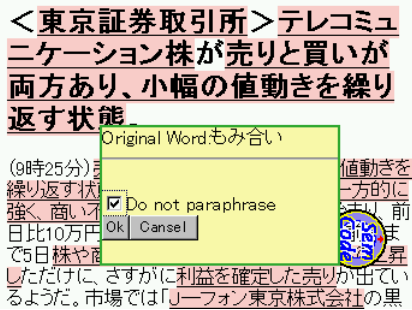


図 8: 言い換え前の語の参照と以後言い換えをするかを決定するポップアップウィンドウ

ある。

- 辞書引きをする単語とその定義文が直接言い換え可能な形式になっていない(例えば、単語とフレーズの置き換えの場合)。
- 辞書の定義文は文脈から独立して意味を持つため、文中にそのまま埋め込もうとすると文脈上のミスマッチが起こる。

この2点を考慮して、われわれは辞書引きする語の属性と、その定義文の属性に基づく言い換えルールを作成し実装した [2]。以下は NikkeiNet [16] 平成 12 年 6 月 1 日付けの経済記事を言い換えた例である。

#### 言い換え前

(9時25分)もみ合い。買い気配で始まり、前日比10万円高の483万円まで上昇したが、前日まで5日続伸ただけに、さすがに利益確定売りが出ているようだ。市場では「J」フォンの黒字転換など明るい材料も多い。資金が一部NTTやNTTドコモから流れてきている可能性もある(中堅証券の情報担当者)との声も聞かれた。

#### 言い換え後

(9時25分)売りと買いが両方あり、小幅の値動きを繰り返す状態。好材料などで買い人気が一方向的に強く、商い不成立で値がつかない状態で始まり、前日比10万円高の483万円まで上昇したが、前日まで5日株や商品取引などの相場が引き続いて上昇ただけに、さすがに利益を確定した売りが出ているようだ。市場では「J」フォン東京株式会社の黒字転換など明るい材料も多い。資金が一部日本電信電話株式会社や株式会社エヌ・ティ・ティ・ドコモから流れてきている可能性もある(中堅証券の情報担当者)との声も聞かれた。

また、言い換えに付いては過去にいくつか研究がな

されている。佐藤らは、複合名詞の言い換え [4] や、サ変名詞の言い換え [3]、格変換による言い換え [5] を提案し実装している。彼らは言い換えを以下の3つのクラスに分類している。

#### 1. 構文的言い換え

言葉に関する知識によって実現可能な言い換え。単語を同義語や類義語に置き換える言い換えや、構造のマッピングに基づく言い換えをこのクラスに分類する。

#### 2. 意味的言い換え

参照表現などを、それが指す内容で置き換える言い換えがこのクラスに含まれる。また、省略されているものを意味的に補う言い換えもこのクラスに含める。

#### 3. 語用論的言い換え

(1), (2) 以外のより複雑な言い換え。ある状況において同じ効果を持つような文に言い換えるものがこれに含まれる。

本稿での言い換えはアノテーションを利用することにより以上のクラスの (1) と (2) を実現する。(2) はアノテーションによって指示詞が何を指すかというようなものを記述できることによる。

われわれはアノテーションを持つテキストと構造化辞書を利用することによって、より一般的な言い換えを実現するルールの作成を目指している。今回試作したルールは日本語の文法依存であるが、適応するルールを変更することにより、特定の文法理論に依存しない言い換換のシステムが実現できる。

## 4 まとめと今後の課題

ポップアップ、挿入、グロッサリーに関しては比較的分かりやすい文書に変換されたように感じた。言い換えに関しては、まだまだルールや辞書が完全に整備されていない為、不自然な言い換えになったり、若干読みにくくなった面も見られた。

動的な辞書の追加ができるようになったため、ある時点で辞書にない言葉でも人手ではあるが意味を付けることによって、将来の辞書引きが可能である点も非常に有益であった。

本稿では Web 文書をアノテーションによって分かりやすく変換する手法について述べたが、これからの課題として以下の点が挙げられる。

- ユーザプリファレンスを考慮した辞書引き
- 提示された文書の分かりやすさ・難しさの定量的評価
- Web を利用した固有名詞等の自動辞書登録
- 単語の自動語義付け
- 著作権に関する対処
- アノテーション作成の自動化

現在は語義アノテーションが付与されたエレメントすべてについて辞書引きを行っている。しかし、それではユーザにとって既知の単語にまで辞書を引いてしまうなどオーバーヘッドが多く、処理にも時間がかかってしまう。

単語親密度<sup>3</sup>をアノテーションとして利用すればその度合いに応じて辞書引きをするかどうかを決定できるが、手に入る親密度のデータではすべての単語をカバーするのは不可能である。それに、すべての人にとって単語親密度は一意に決まらない。ある人にとって分かりやすい単語も、他の人にとっては分かりづらいという可能性は高い。何らかの指標を用いてユーザが持つ語彙を推定し、その人に合った辞書引きをする必要がある。

Web を利用して用語の説明を自動生成する研究も始められている [13]。人手では辞書登録に限界があるため、なるべく自動で行えるようにすべきだろう。

今回提案した4つの提示方法によってどの程度 Web ドキュメントが分かりやすくなったかということについて、今回は定量的な指標を用いていない。さまざまな要因がわかりやすさ・難しさを構成しており、一般的な指標を求めることは非常に困難である。

トランスコーディングプロキシーによって、Web ドキュメントは加工されて、ユーザに提示される。つまり、著作者の意図と異なるようにユーザに提示される可能性がある。しかし、著作者は自らのページに対するアノテーションを拒否することができるし、ユーザはプロキシーを介さなければオリジナルコンテンツにも同じ URL でアクセスできることから、著作権上の問題は少ないと考える。

最後に、語義アノテーションを付けることは現在比較的時間のかかる作業である。文脈情報を利用して、ある

<sup>3</sup> 単語親密度 (word familiarity) とは、ある単語がどの程度なじみがあると感じられるかを表した指標で、認知実験の結果により、そのなじみの程度を 1 から 7 までの数字 (1 : なじみがない ~ 7 : なじみがある) で表している。

単語が辞書の中のどの語義として使われているか推測する機構をエディターに組み込むことを検討している。

## 謝辞

本研究に関して、有益な助言を頂いた慶應義塾大学の石崎俊教授に感謝します。

## 参考文献

- [1] Katashi Nagao, Shingo Hosoya, Kevin Squire, Yoshinari Shirai. Semantic Transcoding: Making the World Wide Web more understandable and usable with external annotations. COLING-2000 Workshop on Semantic Annotation and Intelligent Content 2000.
- [2] 東中竜一郎, 長尾確. アノテーションに基づく知的文書変換. 情報処理学会研究報告 2000-ICS-120 pp.33-40 2000.
- [3] 近藤恵子, 佐藤理史, 奥村学. 「サ変名詞+する」から動詞相当句への言い換え. 情報処理学会論文誌 vol.40 No.11, pp.4064-4074, 1999.
- [4] 佐藤理史, 論文表題を言い換える. 情報処理学会論文誌 vol.40 No.7, pp. 2937-2945, 1998.
- [5] 近藤恵子, 佐藤理史, 奥村学. 格変換による単文の言い換え. 情報処理学会研究報告 00-NL-135 pp.119-126, 2000.
- [6] 益岡隆志, 田窪行則: 基礎日本語文法 (改訂版), くらしお出版 1992.
- [7] 松本裕治, 黒橋禎夫, 山地 治, 妙木 裕, 長尾 真. 日本語形態素解析システム JUMAN 使用説明書 version 3.3 1997. <http://www-nagao.kuee.kyoto-u.ac.jp/nl-resource/juman.html>.
- [8] 日本語構文解析システム KNP <http://www-lab25.kuee.kyoto-u.ac.jp/nl-resource/>
- [9] 日本電子化辞書研究所. EDR 日本語単語辞書マニュアル 1996. [http://www.ijnet.or.jp/edr/J\\_index.html](http://www.ijnet.or.jp/edr/J_index.html).
- [10] 三省堂, 小学国語辞典 (第八版) 1989.
- [11] Koiti Hasida. Global Document Annotation. <http://www.etl.go.jp/etl/nl/gda/>.
- [12] 三省堂 日本語語彙特性データベース 1999. <http://www.brl.ntt.co.jp/cs/human/goi/index.html>
- [13] 桜井裕, 佐藤理史. ワールドワイドウェブを利用した用語検索の実現. 情報処理学会研究報告 2000-NL-137 pp.23-29 2000.
- [14] Extensible Markup Language (XML) <http://www.w3.org/XML>.
- [15] XML Path Language (XPath). <http://www.w3.org/TR/xpath.html>.
- [16] Nikkei Net. <http://www.nikkei.co.jp/>.