情 報 学 基 礎 51－13
自 然 言 語 処 理 127－13
（1998. 9. 17）

Web文書に対する言語処理の問題点と言語処理を援助するタグセットについて

渡辺 日出雄
日本ＩＢＭ（株）東京基礎研究所
watanabe@trl.ibm.co.jp

近年ＷＥＢ文書に対する各種自然言語言語処理プログラム（機械翻訳、自動要約等）が使われるようになってきたが、ＷＥＢ文書を処理する際に遭遇する幾つかの問題点のため本来の処理能力を最大限に発揮しているとは言えない現状である。本論文ではＷＥＢ文書の自然言語処理における問題点について代表的なものを幾つか紹介し、これらの問題点を解消するのに役立つＸＭＬ規約に則った言語的情報を付加するためのタグセット（ＬＡＬ）を提案する。

Problems of NLP for Web Documents and Tag Set Assisting NLP

Hideo Watanabe
IBM Research, Tokyo Research Laboratory
watanabe@trl.ibm.co.jp

Recently some NLP programs are used for Web documents, but these programs do not perform their tasks in their full power due to some problems. In this paper, we introduce these problems and propose an XML-compliant tag set called Linguistic Annotation Language (or LAL) for assiting NLP programs.

1 はじめに

近年のインターネットの普及によりＷＥＢ文書[1]に代表されるインターネット上の文書に対する自然言語処理アプリケーション（例えば、機械翻訳やテキスト自動要約。広義には検索システムも含めることができるだろう）が使用される機会が増えてきている。しかし、自然言語処理システムの不完全さゆえ、一般ユーザーの満足度が高いとは言えない状態である。この満足度の低さのかなりの部分は自然言語処理技術の不完全さから来ているが、ＷＥＢ文書を処理対象とした場合には自然言語処理技術の不完全さ以前の各種問題も存在する。本論文では、まず、ＷＥＢ文書に対する自然言語処理システムが遭遇する問題点について報告する。

これらのＷＥＢ文書に対する自然言語処理の問題点を解消する一つの方策として、自然言語処理システムを援助するような情報を文書中に埋め込むということがある。このような観点から、言語リソースにタグ付けをして共有し、かつ、高度な言語処理を達成しようという動き [1, 2, 4, 7, 9] が多数出てきている。本論文では、基本的にこれらの動きに賛同しつつも、既存のＷＥＢ文書にシームレスにタグ付けすることを目的とし、なおかつ、それほど複雑でない[2]レベルのタグセットを提案する。

2 ＷＥＢ文書処理に関する問題点

ここでは、ＷＥＢ文書の翻訳と要約を例として、自然言語処理プログラムが遭遇する幾つかの問題点を紹介する。

2.1 視覚的効果のためのタグの誤用

ボールド書体で表示させたいなどの視覚的効果を得るために本来使うべきでないタグを使っている例がある。ＨＴＭＬ普及の初期の頃には、特にヘッダータグをそのような用途のために使っているケースが多かった。

HTML ソース

```
I used &lt;h3&gt; tag to emphasize
⟨h3⟩this part⟨/h3⟩
.
```

表示結果

I used ⟨h3⟩ tag to emphasize
this part
.

機械翻訳にとってヘッダータグ内のテキストはタイトルであるとして特別な規則で解析をするという場合があり、このような場合に問題となる。しかしながら、誤用の問題はＨＴＭＬの書き方に関するコンセンサスの広まりとＨＴＭＬ編集プログラムの普及により減少しつつある。更に、今後スタイルファイルの使用が広まれば更に減少することが期待できるであろう。

[1]本論文ではＨＴＭＬとＸＭＬによる記述されたテキストをＷＥＢ文書と呼ぶことにする。

[2]やる気になれば手作業でもタグ付けしたくなるくらいの複雑度を想定している

2.2 文スコープの認定

特に英語の場合、一般に、複数文からなる段落が与えられた場合、その中から文の単位を認定するのはそれほど簡単なタスクではない。例えば、以下の文の場合、

> I went to New Ark St. Paul lived in two years ago.

構文解析をすることなしには一文であると認定するのは難しい。

これに加えて、ＨＴＭＬでは、筆者が ⟨br⟩ タグで文の終わりの代用をしてしまっているケースがある。これは、特にテーブル内でよく見かける現象である。

HTML ソース

```
⟨table⟩
⟨tr⟩
⟨td⟩
Internet Shops⟨br⟩
Cool Sites⟨br⟩
What's New!
⟨/td⟩
⟨/tr⟩
⟨/table⟩
```

自然言語処理プログラムにとってテーブルのセル内が一行一文として訳すべきか、普通に句読点までを一文として訳すべきか知ることは出来ない。

2.3 翻訳結果へのタグの挿入

機械翻訳プログラムでは、翻訳結果にソースと同じようにタグを挿入するという処理をする必要がある。デフォールトの動作としては、ソースの対応する単語に付いているタグをターゲット側にも付ければよいのであるが、そうでないケースもある。以下の例では、

> ⟨p⟩ 海の中に沢山の ⟨a href="..."⟩ 魚 ⟨/a⟩ がいます。⟨br⟩

デフォールトの動作だけだと次のような出力になってしまうが、

> There are many ⟨a href="..."⟩fishes⟨/a⟩ in the ⟨p⟩sea.⟨br⟩

本来は以下のようにならなければならない。

> ⟨p⟩There are many ⟨a href="..."⟩fishes⟨/a⟩ in the sea.⟨br⟩

これは、⟨p⟩ タグが文の先頭（性格には段落の先頭）に位置すべきものであるという位置に関する制約が分かって初めて可能となる。ＨＴＭＬを処理するプログラムではこれらの位置制約があらかじめプログラムの中に埋め込まれているので正しい結果が得られるが、今後増えるであろうＸＭＬの翻訳を考えると、任意のタグを作成可能なので、何らかのタグの位置制約を記述する仕組みが必要となる。

2.4 文書内要素の認定

WEB文書は一般に様々な要素から構成されている。典型的な例としては、関連リンク、タイトル、本文、著者名、著作権表示などから構成されている。現在のHTMLでは意味を理解しない限りこれらの要素を正確に認定するのは困難である。

ここで、問題となるのは、テキストの自動要約処理である。現在実用化されている自動要約処理プログラムのほとんどのものは、重要な文を取り出す文抽出型のもの[3, 10, 13]である。この文抽出型要約処理では、それぞれの文に関して、含まれる重要なキーワードの数、文のタイプ（事実、意見など）、文の文章中の位置（始めと終わりにある文は重要であることが多い）等の情報を基にして重要度を決定しているものが多い。この様な処理の場合、HTMLテキスト中の要約処理に関連する本文とタイトル部分だけを計算対象に含めないと、おかしな要約結果が出てしまうことになる。現実的には幾つかのパターンを用意して著作権表示部分を対象外にしたりしているが、全ての場合を尽くすパターンを用意できないので、ページの先頭や末尾付近の文が選ばれることがある。

2.5 言語の認定

多言語が混在した文書に対する自然言語処理を考えた場合、小さな単位（例えば段落等）での言語認定が必要である。統計的に言語を認定するプログラムは色々あるが、小さな段落では間違うことも多い。基本的にはlang属性をきちんとつけてもらうしかないであろう。

3 Linguistic Annotation Language

ここでは、前節で挙げた各種の問題点を解消することを目指したタグセット(Linguistic Annotation Language or LAL)に関して説明する。

3.1 設計方針

LALは以下のような方針に基づいてデザインした。

- 既存の文書へのタグ付け ... 新たな文書タイプを導入するのではなく、既存の文書にタグ付け出来ることを目指す。具体的には、任意のXML文書へタグ付けできることを目標とする。
- 簡単さ/効率の良さ ... 基本的にはツールを使ってタグ付けするのであろうが、人手であっても可能な程度の簡単さを目指す。これは、ツールを使った場合でも、ユーザーへのフィードバックを考えた場合に必要であると思われる。また、効率の面からも、自然言語処理プログラムの問題点の全てを解決するタグセットではなく、なるべく少ない労力でかなり大きな改善率が選られるようなレベルのタグセットが必要である。
- 自然言語処理プログラムの援助 ... 自然言語処理プログラムのタイプにより固有のアルゴリズムなどがあるので、それら固有の処理を援助するようなタグも導入する。
- 既存のタグセットとの連続性 ... 従来から提案されているタグセットの中で、上記の基準に合うものは積極的に取りいれていくことにより、一種のサブセットの提案を目指す。

最初の設計方針である「任意のXML文書へのタグ付け」を考えた場合、XML[11] Namespace[12][3]を用いて言語処理用のタグセットを定義するというのをまず最初に思いつくことであろう。しかし、以下のような例[4]を考えて見ると分かるように、

> She ⟨a href="..."⟩⟨seg⟩went to Paris⟨/a⟩ last month⟨/seg⟩.

XML（あるいはHTML）タグのスコープと言語処理タグのスコープが交差するOverlapping Hierarchies[1]と呼ばれる現象が割と頻繁に起こることがわかる。これは、XMLのタグの制約に違反してしまうので、XMLの通常のエレメントとなるタグを用いることは出来ない。そこで、LALでは、そのような制約がなく任意の位置に書くことが出来る処理命令(Processing Instructin or PI)を用いて実現することとした。[5]ただ、PIを用いることで、XMLパーザーなどが持つであろうタグの入れ子構造のチェックなどの機能を用いることは出来なくなるという欠点はある。[6]

次に「簡単さ/効率の良さ」を考えると、自然言語処理プログラムの誤りのかなりの部分は、前置詞句の係り受け、従属節の係り受けや並列句の範囲認定などの構文的な範囲認定（スコーピング）が正しく分かれば解消できると思われる。よって、LALでは、文のレベルを含めた構文的範囲指定を主眼に置いて、使用するタグを限定することにした。これによりかなり少ない種類のタグで割と大きな改善効果を見込むことが出来、効率の良いタグセットと言えることになる。

3.2 LALタグの構造

LALタグは、XMLの処理命令を用いて実現する。XMLの処理命令は以下のような形式をしている。

> ⟨?処理命令ターゲット 処理内容?⟩

処理内容の部分に関しては特に決められた形式はないので、LALタグは処理ターゲットとしてlalを用い、処理内容の部分は以下のような形式とする。

> 開始タグ ::= ⟨?lal タグ名 付加情報?⟩
> 終了タグ ::= ⟨?lal _タグ名 付加情報?⟩
> 空要素タグ ::= ⟨?lal タグ名 付加情報_?⟩

[3] これはまだW3Cより正式な勧告とはなっていない。

[4] この例の中で使われている⟨seg⟩は任意の句を指定するものとする。

[5] XMLでのタグ付けと言語処理のタグ付けの目的は異なるので、交差は避けられないはずであり、本来これらのタグは各種の制約が独立に適用されるような別空間で記述できるようにすべきであろう。

[6] しかし、以下に述べるように、LALで用いる処理命令の構文形式をXMLと同様のものにすることにより、LAL対応のパーザーをXMLパーザーを変更して作成するのは容易であると思われる。

上記の形式に加えて、ＬＡＬタグには以下のような制約がある。

- 開始タグと終了タグは同一のタグ名を持ち、かつ終了タグのタグ名の前には'_' を付ける。(よって、タグ名の先頭文字として、'_' を使うことは出来ない。)
- 空要素タグは、付加情報の最後が'_' で終わっている。(よって、空要素でないタグの処理内容の最後が'_' であってはならない。)
- ＬＡＬタグに関して、開始タグと終了タグで構成されるスコープが互いに交差してはならない。

3.3 構文タグ

構文タグとしては以下のものを使用することにした。

文 (s) ... 文の単位を示す。

⟨?lal s?⟩ これは文です。⟨?lal _s?⟩

単語 (w)... 単語を指定する。

⟨?lal w?⟩New York⟨?lal _w?⟩

句 (seg) ... 任意の句のレベルを指定する。また、句のカテゴリーを指定するオプショナル属性 cat を指定できる。

She saw ⟨?lal seg cat="np"?⟩a man with telescope⟨?lal _seg?⟩.

3.4 意味タグ

意味的な情報指定タグとしては以下のものに限定する。これらは、基本的に句の境界を指定すると同時に意味的な属性も指定していると考える。

固有名詞 (proper) サブタイプ (type 属性) として人名・場所・機関・国等の指定もできる。

⟨?lal proper type="country"?⟩U.S.A.⟨?lal _proper?⟩

日付 (date)

⟨?lal date?⟩Dec. 25, 1999⟨?lal _date?⟩

時間 (time)

⟨?lal time?⟩6:00 PM EST⟨?lal _time?⟩

数値表現 (num)

⟨?lal num?⟩three hundred and twenty-one⟨?lal _num?⟩

ＨＴＭＬでは abbr と acronymn のタグが使用できる。ＬＡＬではＨＴＭＬ文書においてこれらのタグの使用を推奨する。また、ＸＭＬでは、これらと同様の以下のＬＡＬタグを用意する。

⟨?lal abbr title="doctor"?⟩Dr.⟨?lal _abbr?⟩
⟨?lal acronymn title="International Business Machines"?⟩IBM⟨?lal _acronymn?⟩

3.5 処理依存タグ

ここでは、特定の自然言語処理アプリケーション用のタグということで、機械翻訳と自動要約用のタグについて説明する。

3.5.1 翻訳スコープ

翻訳プログラムに対して明示的に翻訳のスコープを指定するタグとして以下のものを用いる。

⟨?lal tranStop_?⟩ ... 翻訳処理の停止
⟨?lal tranStart_?⟩ ... 翻訳処理の開始

初期状態は翻訳開始状態であるとする。

3.5.2 翻訳タグ情報

前述したように、翻訳結果へのタグ付けはタグの使われ方に関する情報がないと正しく行えない場合がある。そこで、以下のようなタグ情報 (taginfo) タグを用いて直後の XML(又は HTML) タグの使われたかに関する情報を付加する。

⟨?lal taginfo TagAttr*_?⟩
TagAttr ::= tagType | tagLoc
TagType ::= 'type=' ('open' | 'close' | 'single')
TagLoc :== 'loc=' ('bos' | 'eos').

ここで、type の open は開始タグ、close は終了タグ、single は空要素タグであることを示し、loc の bos はそれが文頭に置かれること、eos は文末に置かれることを示す。

例えば、以下の HTML 行は、

⟨p⟩ ⟨a href="..."⟩ IBM ⟨/a⟩

以下のように注釈付けすることが出来る。

⟨?lal taginfo type="single" loc="bos"_⟩ ⟨p⟩
⟨?lal taginfo type="open"_⟩ ⟨a href="..."⟩
IBM ⟨?lal taginfo type="close" _⟩ ⟨/a⟩

3.5.3 要約スコープ

これも、前述したように、要約の対象となる部分を正確に判定するのは難しいので、以下のような要約対象を指定するタグを導入する。

```
⟨html⟩
⟨title⟩ThinkPad 770 Notebooks⟨/title⟩
⟨body⟩
⟨h1⟩ThinkPad 770 Notebooks⟨/h1⟩
⟨p⟩
⟨a href="http://www.ibm.com/"⟩IBM⟨/a⟩ announced the new ThinkPad 770. The ideal combination of
mobile computing technology and features into one lean, powerful package.
⟨/body⟩
⟨/html⟩
```

(a) HTML ソース

```
⟨?xml version="1.0" encoding="us-ascii" ?⟩
⟨!DOCTYPE html PUBLIC "-//W3C/DTD HTML 4.0//EN" ...⟩
⟨?lal tranStop_?⟩
⟨html⟩
⟨title⟩⟨?lal tranStart_?⟩ ⟨?lal s type="hdr"?⟩ThinkPad 770 Notebooks ⟨?lal _s?⟩⟨?lal tranStop_?⟩ ⟨/title⟩
⟨body⟩
⟨h1⟩⟨?lal tranStart_?⟩ ⟨?lal s type="hdr"?⟩ ThinkPad 770 Notebooks⟨?lal _s?⟩ ⟨?lal tranStop_?⟩ ⟨/h1⟩
⟨?lal tranStart_?⟩
⟨?lal taginfo type="single" loc="bos"_?⟩ ⟨p⟩
⟨a href="http://www.ibm.com/" ⟩ ⟨?lal proper type="organization"?⟩IBM ⟨?lal _proper?⟩ ⟨/a⟩ announced
the new ⟨?lal proper?⟩ ThinkPad 770⟨?lal _proper?⟩. The ideal combination of ⟨?lal seg?⟩mobile computing
technology and features⟨?lal _seg?⟩ is assembled into one lean, powerful package.
⟨?lal tranStop_?⟩
⟨/body⟩
⟨/html⟩
```

(b) LAL による注釈

図 1: LAL による注釈例

⟨?lal smrycalcStart_?⟩ ... 要約処理の開始
⟨?lal smrycalcStop_?⟩ ... 要約処理の停止

初期状態は要約処理開始状態である。

4 例

ここでは、ＬＡＬによる注釈の一例を示す。図 1(a) がオリジナルの HTML 文書であり、図 1(b) がそれのＬＡＬによる注釈付けの例である。

5 議論

従来から、言語リソースをタグ付けして共有しようという試みとして各種のもの [1, 2, 4, 7, 9] が知られている。これらは主に自然言語処理研究の発展のために考えられたものであり、意味レベルまでのかなり詳細な記述が可能となっている。そのため、一般的に普及するに至っていない。最近のＨＴＭＬの普及により一般ユーザーのある程度のタグ付けを期待した試みとして GDA[4, 5] がある。しかし、これもかなり詳細な記述を目指しており、一般ユーザーが気軽にタグ付けしたくなるレベルとは言えないであろう。ＬＡＬで目指したのは、実際にはツールが必要になるにせよ、ユーザーがなんとか人手で言語的タグ付けが出来るレベルのタグセットの提案であり、先行研究のある意味でのサブセットレベルを提示することにもなっている。このように、人手でのタグ付けが可能であることを目指してはいるが、ＨＴＭＬ文書の作成には最近ではツールを使う場合が多いことを考えると、言語的注釈付けもこのようなツールに統合される必要があるだろう。このようなツールでは注釈付けの結果をユーザーにフィードバックする必要があり、このことを考えてもある程度簡潔なレベルのタグセットにしておく必要があると考える。

更に、ＬＡＬの特徴としては、新たな文書型を導入するのではなく、既存の文書型（ＸＭＬ文書）中に言語的注釈をシームレスに挿入できることにある。このため、ＸＭＬの処理命令というメカニズムを使って実現した。新たな文書型の提案は、それを普及させるという観点からするとかなりの労力が必要である。それよりは、既存の文書型に変更を加えずに言語的情報を付加出来る方が一般には受け入れやすいと思われる。既存の文書型をそのまま使うという方式としては、ＬＡＬで提案したシームレス挿入方式の他に別文書として管理するという方式もあり、ＣＥＳ [1] ではこの後者の方式を採用している。別文書方式ではまったく元文書を変更する必要がないという点で優れているが、元文書と注釈データを一つとして管理する手間がかかるという問題がある。[7]

6 おわりに

本論文では、ＷＥＢ文書に対して機械翻訳や自動要約などの自然言語処理を行う場合の問題点について説明し、ユーザーからの補助があれば現状の技術レベルでももう少し良い処理結果を期待できることを示した。

また、ユーザーからの補助の手段として、言語的情報をタグ付けするためのタグセットであるＬＡＬを提案した。これは、任意のＸＭＬ文書にシームレスに言語的情報を挿入できるものであり、主要なタグは単語や句などの様々なレベルでの境界を指定するものである。更に、

[7] 例えば、元文書に変更に応じて注釈データを同期的に更新する仕組みが必要となる。

機械翻訳や自動要約などの言語処理に依存した情報付加のためのタグも用意した。

今後、ＬＡＬによるタグ付けツールとＬＡＬタグを認識する言語処理プログラムの開発などにより、言語的なタグ付けがされたＷＥＢ文書が増え、ユーザーが現在の言語処理プログラムを最大限に利用できるようになることを目指していきたい。

参考文献

[1] Corpus Encoding Standard (CES) (http://www.cs.vassar.edu/CES/)

[2] Expert Advisory Group on Language Engineering Standards (http://www.ilc.pi.cnr.it/EAGLES/home.html)

[3] H. P. Edmundson, "New Methods in Automatic Extracting," Journal of Association for Computing Machinery, Vol. 16, No. 2, pp. 264–285, 1969.

[4] Global Document Annotation (http://www.etl.go.jp/etl/nl/gda/)

[5] Koichi Hashida, Katashi Nagao, et. al, "Progress and Prospect of Global Document Annotation," (in Japanese) Proc. of 4th Annual Meeting of the Association of Natural Language Processing, pp. 618–621, 1998.

[6] Data elements and interchange formats – Information interchange – Representation of dates and times, ISO 8601:1988.

[7] A Standard Extraction/Abstraction Text Format for Translation and NLP Tools (http://www.opentag.org/)

[8] ISO/IEC 8879-1986 (E). Information processing – Text and Office Systems – Standard Generalized Markup Language (SGML). First Edition – 1986-10-15.International Organization for Standardization, 1986.

[9] Text Encoding Initiative (TEI) (http://www.uic.edu:80/orgs/tei/)

[10] Hideo Watanabe, "A Method for Abstracting Newspaper Articles by Using Surface Clues," Proc. of 16th International Conference of Computational Linguistics, pp. 974–979, Aug. 4-9, 1996.

[11] Extensible Markup Language (XML) (http://www.w3.org/TR/PR-xml-971208), World Wide Web Consortium, Dec. 8, 1997.

[12] Namespaces in XML (http://www.w3.org/TR/1998/WD-xml-names-19980327), World Wide Web Consortium, March 27, 1998.

[13] 山本和英、増山繁、内藤昭三、「文章内構造を複合的に利用した論説文要約システム GREEN」、自然言語処理、Vol.2, No. 1, pp. 39–55, 1995.