論文全文データベース作成のためのSGMLタグ付けの自動化*

成田 えりか　　松本 裕治
奈良先端科学技術大学院大学 情報科学研究科

1 はじめに

本研究では、電子図書館などにおいて効果的な検索をもたらすと考えられる全文データベース作成のための一方法について述べる。特に科学技術論文をとりあげ、それらのプレーンテキストにSGMLタグを自動的に付けていく方法について考える。SGMLとは、Standard Generalized Markup Languageのことであり、1986年に国際標準化機構(ISO)によって発行された[1]。SGMLを用いることの利点としては、テキスト交換の容易さ、テキストの意味的な構造を意識したタグの使用[2]などがあげられる。現在、日本でもSGMLを用いた全文データベースの作成が積極的に行なわれている[3],[4],[5]。SGMLを用いた論文データは、例えば章タイトルだけでの検索、導入部や結論など特定の節内のみの検索など、多様な検索方法を可能にする。ここでは、OCRで読みとった科学技術論文のプレーンテキストを対象に、文書構造パターンというものを用いた自動タグ付けを提案する。

2 文書構造パターンを用いたタグ付けの処理の流れ

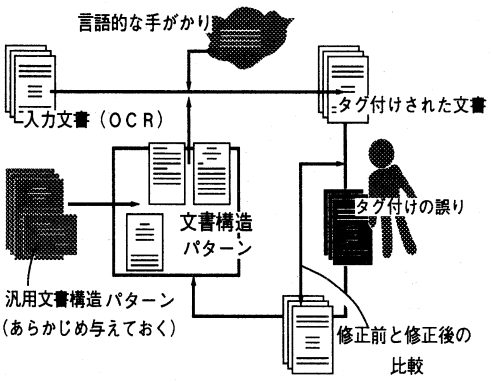


図1: 文書構造パターンを用いたタグ付けのプロセス

*Automatic SGML Tagging for Full-text Databases of Technical Papers

図1は、文書構造パターンを用いたタグ付けの流れを示したものである。

タグ付けの過程は、次のようなサブタスクからなる。

1. OCRによる論文の読み取り
2. 部分的形態素解析
3. 文書構造パターンを用いての論文のタグ付け
4. タグ付けされた論文のタグの誤りの指摘（タグの修正）
5. タグ付き論文の修正前と修正後の比較
6. 新たなフォーマットの文書構造パターン獲得

文書構造パターンは、タグを付ける対象がプレーンテキスト内のどこにどのように存在するかを示す情報で構成されている。

SGML文書の構成の一つにDTD(Document Type Definition: 文書型定義)があり、SGML文書作成の際にはこのDTDを参照する。DTDは、文書中のすべての要素の名前と内容、要素の出現回数、出現順番、タグが省略可能かどうかなどを定義している[1]。

タグ付けの自動化を行なう時にDTDの情報は重要であるが、プレーンテキストのタグ付けを行なうには、まだ十分ではない。そこでタグを付けるための情報として、文書の内容に対する制約を付加したものが必要であり、それを文書構造パターンと呼ぶ。

文書構造パターンの制約の中には、言語的な手がかりが必要なものも含まれているので、プレーンテキストに対して部分的な形態素解析を行なう。

文書構造パターンを用いてプレーンテキストにタグを付けた後、既存のパターンではタグ付けがうまく完了しなかった部分をユーザーが修正し、システムがその差を見ることにより、文書構造パターンの修正を行なう。これにより文書構造パターン内の制約の種類と数は詳細化され、最終的にはほぼ自動的にプレーンテキストのタグ付けが行なえるようになる。

3 技術論文の構造と文書構造パターン

本研究で対象とするタグ付けにおいて、重要な鍵を握っているのは文書構造パターンである。どのようなパターンが必要かを知る手がかりとして、日本語論文の構造を例としてあげる。

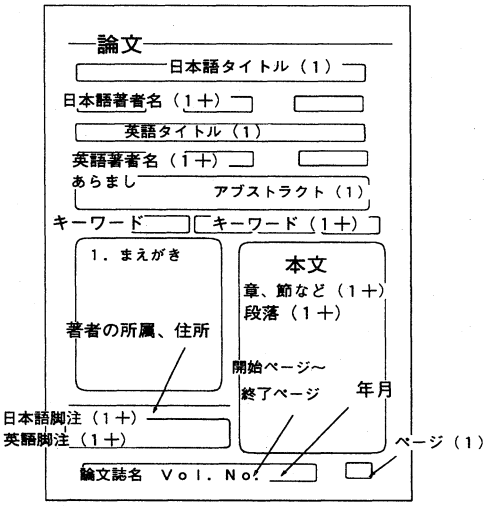


図 2: 日本語論文の構造例 (1)

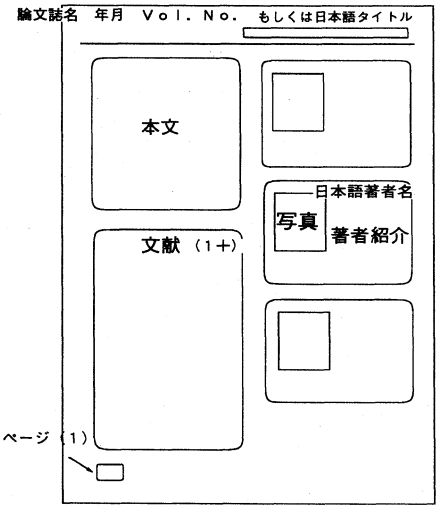
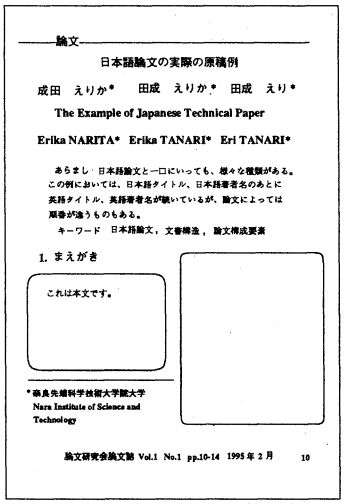


図 3: 日本語論文の構造例 (2)


論文

日本語論文の実際の原稿例

成田 えりか* 田成 えりか* 田成 えり*

The Example of Japanese Technical Paper

Erika NARITA* Erika TANARI* Eri TANARI*

あらまし 日本語論文と一口にいっても、様々な種類がある。この例においては、日本語タイトル、日本語著者名のあとに英語タイトル、英語著者名が続いているが、論文によっては順番が違うものもある。

キーワード 日本語論文，文書構造，論文構成要素

1. まえがき

これは本文です。

* 奈良先端科学技術大学院大学
Nara Institute of Science and Technology

論文研究会論文誌 Vol.1 No.1 pp.10-14 1995年2月 10

図 4: 実際の原稿例

論文
日本語論文の実際の原稿例
成田えりか*田成えりか*田成えり*
The Example of Japanese Technical Paper
Erika NARITA* Erika TANARI* Eri TANARI*
あらまし日本語論文と一口にいっても、様々な種類がある。
この例においては、日本語タイトル、日本語著者名のあとに
英語タイトル、英語著者名が続いているが、論文によっては
順番が違うものもある。
キーワード日本語論文, 文書構造, 論文構成要素
1. まえがき
これは本文です。

図 5: OCR 読み取り例

図2と図3は、それぞれ論文の最初のページと最後のページの構造例である。括弧でくくられた数字は、タイトルなどの論文の構成要素がいくつ存在するのかを表している。

図2のような構造は、論文誌上では実際には図4のようなページとしてあらわれている。このページをOCRで読み取ると、図5のようなテキストが得られる。

タグ付けを行なう対象であるプレーンテキストは、ほぼ図5の形をとっている。プレーンテキストの形は、実際には読み取りに用いるOCRの精度に左右されることが多い。ここでは、OCRは読み取ったページの改行を忠実に再現すると仮定する。また、テキストの脚注、図、表に関しては取り扱わないとする。

図6では構造例とOCR読み取り例の対応関係を見て、必要なタグを取り出している。

文書の種類（1個）：論文

日本語タイトル（1個）：日本語論文の実際の原稿例

日本語著者名：成田えりか*田成えりか*田成えり*
（1個以上、*によって個々の名前が区切られている）

英語タイトル（1個）：
The Example of Japanese Technical Paper

英語著者名：
Erika NARITA* Erika TANARI* Eri TANARI*
（1個以上、*によって個々の名前が区切られている）

「あらまし」（1個）：
あらまし日本語論文と一口にいっても、様々な種類がある。
この例においては、日本語タイトル、日本語著者名のあとに
英語タイトル、英語著者名が続いているが、論文によっては
順番が違うものもある。

「キーワード」：
キーワード日本語論文, 文書構造, 論文構成要素
（1個以上）

図 6: OCR 読み取り例と構造例の対応

```
文書種類：
<documentstyle>‘‘論文’’,(1)</documentstyle>
日本語タイトル：
<jtitle>(日本語 and 名詞句),(1)</jtitle>
日本語著者名：
<jauthor>(日本語 and 固有名詞)‘‘*’’,(1+)=auth</jauthor>
英語タイトル：
<etitle>(英語 and 名詞句),(1)</etitle>
英語著者名：
<eauthor>(英語 and 固有名詞)‘‘*’’,(1+)=auth</eauthor>
ラベル：
<label>‘‘あらまし’’(1)</label>
日本語要約：
<jabstract>(日本語 and 文),(1+)</jabstract>
ラベル：
<label>‘‘キーワード’’(1)</label>
日本語キーワード：
<jkeyword>(日本語 and 名詞句),(1+)</jkeyword>
本文：
<body>(日本語 and 文),(1+)</body>
ラベル：
<label>‘‘参考文献’’,(1)</label>
参考文献内容：
<bibliography>‘‘[’’(数字)‘‘]’’( ),(1+)</bibliography>
```

図 7: 文書構造パターンの例

文書構造パターンはすでに述べた通り、文書の内容に対する制約の情報を持つ。

例えば図5の中に含まれる制約は、図6の丸括弧内に書かれている事項である。また、その他「あらまし」や「キーワード」といった言葉も、後に続くテキストの内容を表していることから制約に加える。

論文を構成する要素それぞれの個数も、重要な制約の一つである。論文を構成する要素には、1個しかないもの、1個以上あるもの、0個以上あるもの（ない場合もあるもの）、ないかまたはあるなら1個しかないものの大きくわけて4種類のものを考える。これらは図の中で、それぞれ (1)、(1+)、(0+)、(0,1) と表現される。

図7は文書構造パターンの例であり、そこにはタイトルから本文までと参考文献の制約がかかれている。これらは、プレーンテキストの表面的な情報で比較的満たされやすい制約であるといえる。参考文献は著者名、書名など複数の情報のセットだが、それらが参考文献であるということは表面的に調べられると考え、制約に加えた。図7の文書構造パターンは、図2の構造例に従って記述している。

文書構造パターンの制約の記述は、以下の形に従っている。

- **文書構造パターンの制約の順序**は、対象とする文書の構成要素の出現順序に対応している（例えばタイトルの制約が、参考文献の制約の後ろにかかれたりはしない）
- 各制約は、<>で始まり</>で終るタグセットでくくられている
- 文書中に現れる**言葉**、**記号**は、“ ” で囲む
- **言語の種類**、**品詞の種類**などの文法上の制約は丸括弧 ( ) で囲む
- 文法上の制約の括弧内では、**and** と **or** でそれぞれの制約間の関係を表す
- 文書の構成要素の個数の制約は **(1)** = 1個、**(1+)** = 1個以上、**(0+)** = 0個以上（0個を含む）、**(0,1)** = 0個あるいは1個（2個以上を受け付けない）で表現する
- 例えば日本語著者名と英語著者名のように、要素の個数の制約が一致しなければならないものは **(個数)=変数** の形で変数による束縛を行なう（例えば著者名の変数は auth）

これらの記述方法をまとめると、文書構造パターンの各構成要素は、

< 開始タグ >文字列と制約による正規表現,(個数)
</ 修了タグ >

と記述される。

4 汎用文書構造パターン

文書構造パターンにあてはまらないフォーマットのプレーンテキストが与えられた場合、初期的な文書構造の解析を支援するために、汎用的な文書構造が用意されている。これを汎用文書構造パターンと呼ぶ。汎用文書構造パターンは、論文構造の共通要素、例えばタイトルや著者名についての情報を持っている。

5 自動タグ付けの処理

自動タグ付けの流れを以下の図に示す。

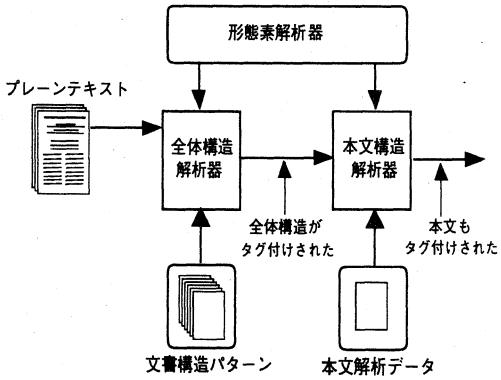


図 8: 自動タグ付けのプロセス

タグ付けの自動化を行なうに際して、文書構造パターンの他に文書の本文の部分を解析するためのデータも用意しておく。これによって、文書の全体の構造によらず、本文の構造解析の処理が行なわれる。なぜなら全体の構造と本文の構造は独立性が高いからである。

自動タグ付けの過程は、以下の通りである。

1. プレーンテキストから1行を読み込み、文書構造パターンの1行目と合致するか否かをみる
2. 制約違反を起こした場合、別の文書構造パターンを用いて1行目からやり直す
3. 制約を満たしている場合、タグを付けてから2行目以降をみる
4. すべての制約が満たされた時、その文書構造パターンを用いてタグ付けされたテキストを結果とする

この段階では、本文にはまだタグが付いていない状態である。

次に、ここでは本文のタグ付け方法の一例として、数字・記号項目のタグ付け方法について述べる。

1. まず、(a)(1)(ア)[a][1]1 …を探す機能があることを前提とし、本文のプレーンテキストの最後から読み込む
2. 最初にであった一番始めの数字・記号（(a)(1)(ア)[a][1]1 など）にマークを付け、戻りながら二番目、三番目…と数字・記号の順番が途切れるところまでタグを付けていく
3. 違うマークを用いて、上記の動作を繰り返す
4. マークの位置に応じてタグ付けを行なう

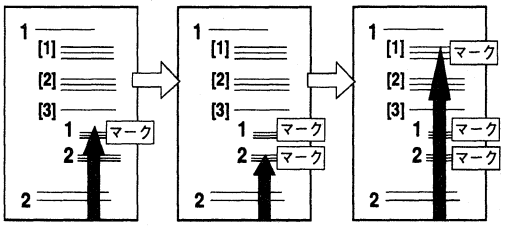


図 9: 本文の数字・記号項目のタグ付け方法例

6 おわりに

ここでは科学技術論文を対象に、文書構造パターンを用いたSGMLの自動タグ付け方法を提案した。まず、文書の構造の表現として文書構造パターンを提案し、次に文書構造パターンを用いたタグ付けの処理の流れについて述べた。

今回は、文書構造パターンを用いてタグを自動的に付ける手法を中心に述べたが、今後の課題として、文書構造パターンの自動学習に取り組む予定である。

参考文献

[1] van Herwijnen, Eric : 実践SGML, 日本規格協会 (1992).

[2] 根岸正光, 石塚英弘 : SGML の活用, オーム社 (1994).

[3] 石塚, 伊藤, 榎, 千原, 中西, 田中 : 日本化学会欧文誌のSGML 形式全文データベースの構築・印刷そして検索, 情報処理学会情報学基礎研究会資料, 29-1 (1993).

[4] 石塚, 伊藤, 榎, 千原, 中西, 田中 : 全文検索システムのリーソースとしてのSGML 方式データベース, 情報処理学会情報学基礎研究会資料, 33-6 (1994).

[5] 石塚, 伊藤, 竹内, 千原, 中野, 真野, 吉村, 中西, 田中 : 電子投稿によるSGML 方式全文データベースの作成 – 日本化学会の実験 –, 情報処理学会情報学基礎研究会資料, 35-1 (1994).