文書構造の帰納学習に基づく構造化記述の自動生成*

1Y－4

水野 貴史　武田 正之†
東京理科大学大学院 理工学研究科 情報科学専攻‡

1 はじめに

近年、従来からの紙に印刷された文書から計算機上で利用されることを目的とした電子化された文書まで様々な形態の文書が存在する。これらの文書を統一的に扱えることができれば、文書形式に応じて閲覧ソフトウェアを変更する必要がないなど、文書利用・交換の観点から有効であることは明らかである。

本研究では、帰納学習システム Progol を利用して、文書中に現れる単語の位置関係などを用いて電子化文書から構造化記述された文書を自動生成することを試みる。タグ付けの手段として、文章形式を自由に扱うことのできる XML(eXtensible Markup Language)[1]の文書形式を利用する。XML にはリンク機構などがあるが、本研究においてはそのような機能は利用せず、XML の表現法のみを構造化記述を行うための手段とした。

2 構造化記述の自動生成

2.1 帰納学習

構造化記述を生成する際に予め与えておいた訓練例から一般的な規則を発見するために帰納推論を行う。この帰納推論を行うために、S.Muggleton の開発した帰納学習システム Progol[2] を用いる。Progol は正事例・負事例・モード宣言・タイプ宣言・背景知識を入力として、モード宣言に従って正事例と背景知識から負事例を被覆しない仮説を生成する。

本研究においては文書種別を判断する規則と構造化記述を生成する際のタグ付け判定規則を学習するときに Progol を用いる。文書種別判断部においては、文書の種別を特徴付けるキーワードと文書全体の行数、キーワードの出現した行番号をリスト形式にして正事例として与える。文書の種類の間には排他的な関係を仮定したので、ある文書種別に対して正事例となるものは他の文書種別では負事例となる。構造化記述生成部では、訓練例においてタグが付加される前後の行や文書全体の構造等の情報をリスト構造にして正事例として与える。この場合は、タグに関して絶対的な排他関係が存在しないが、負事例登録に文書種別判断部と同じ処理を行い、Progol のパラメータ noise(負事例を被覆する割合) で調整を行う。また、背景知識はリスト処理を行う述語を用意しておく。

2.2 システムの概要

本研究のシステムの概要を図 1 に示す。

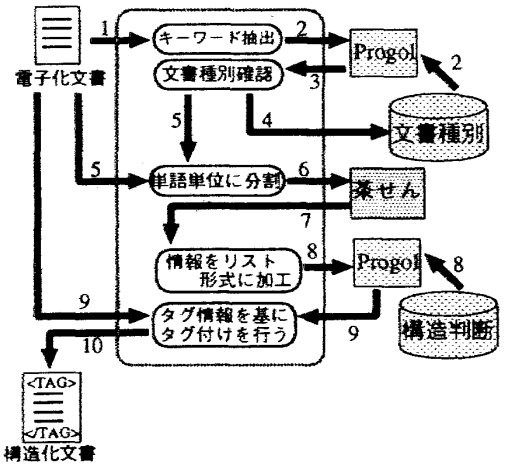

図 1: システム構成

システムは、大きく 2 つの部分からなる。まず、入力された文書からキーワードを抽出して文書種別を判断する (文書種別判断部)。更に、入力された文書を単語単位に分割し、行番号などの必要な情報と共に行毎に構造フレームを形成する。構造フレームからタグ付けに関する情報を得て、構造化文書を生成する (文書構造記述部)。

まず、文書種別判断部 (1〜4) について説明する。文書から文書種別を判断するために必要と思われるキーワードを抽出する。その抽出されたキーワードとそれが現れた行番号、全行数をリスト形式にする。複数のキーワード情報を更にリスト形式にして Progol に渡す。Progol では、キーワードそのものや位置情報から

*Automatic Generation of Structured Description based on Inductive Lerning of Document Structure
†Takafumi Mizuno and Masayuki Takeda
‡Dept. of Information Sciences,Science University of Tokyo

適切な文書種別を判断し、その結果を新たに文書種別を判断するために知識ベースに登録する。始めは、知識ベースには事実のみが蓄えられているが、ある程度、知識を習得した時点で知識の一般化 (学習) を行う。学習によって、その文書種別を特徴付ける節形式の述語が得られる。例えば、メールを判定する節として以下のものが得られた。

```
sort_2(A) :- have_p(recived,top,A).
```

これは、recived と言う語を文書の上部に持つことを意味している。

次に、文書構造記述部 (5～10) について見てみる。与えられた文書を茶筌に入力する。ここで、茶筌とは奈良先端技術大学院大学自然言語学処理講座で開発された日本語形態素解析器である。茶筌では入力されたものを単語単位に分割し、同時に意味情報も返す[1]。そこで、処理が行われている行 (対象行)、及び前後の行に対して単語に分割された情報を得る。文書全体のインデントなどの段落構造、最初に判断した文書種別などと共にリスト形式の構造フレームを作成する。構造フレームの形態は、以下の形式を取る。

```
[[文書全体構造],[対象行の直前行],
   [対象行],[対象行の直後行],[行番号],
                [全体の行数],[文書種別]].
```

以上が構造化記述を生成する流れであるが、予め、訓練例を利用して規則を学習しておかねばならない。

3 評価

ニュース文書 30 件、メール文書 30 件、マニュアル 24 件をサンプルとして実験を行った。ニュース、メール、マニュアルそれぞれに対して簡単な DTD[2] を定義し、その DTD に従ってタグ付けを行った構造化文書を作成して、学習を行わせた。3 種類の文書に対して合計 17 種類のタグを用意した。メールやニュースの文書に対しては、ヘッダや引用を表すタグを設定し、マニュアルに対しては、セクションや URL を表すタグを設定した。以下に学習に成功した節を記しておく。

```
tag_s6([A,B,C,D,E,F,G]) :-
                  listtop('Subject',C).
tag_s15([A,B,C,D,E,F,G]) :- numtop(C).
```

tag_s6 は現在処理している行の先頭が Subject という語であれば 6 番のタグを付加し、tab_s15 は先頭が数字で始まるなら 15 番のタグを付加するという意味である。

また、学習に成功した節数は以下のようになった。

訓練数	9	18	27	36	…
学習節数	28	30	31	31	…

訓練数が 9 とはニュース、メール、マニュアルの各文書から 3 つずつ取り出してシステムに訓練させた文書数を表す。ここで、訓練数が 9 のとき現れた開始・終了タグは合計 32 個で、他のものは 34 個である。訓練数が 9,18 のときに学習出来なかったのは、訓練数が足りないためであると考えられる。訓練数が 27 以降の場合は全 34 個中 31 個学習できた。学習できなかった 3 つのものは、文書種別を表す終了タグ (</MAIL>等) であったが、これは、この種のタグが全ての文書において最終行に現れるために、負事例が多過ぎてしまい一般化出来なかったためであることが分かった。このままでは、知識ベースに事実としてのみ登録されており一般性に欠ける。この対処法としては、負事例を減らしてやり noise を増加させることで学習が行える。

更に、noise による学習の変化を調べてみたが、noise をある程度 (30%) まで変動させても学習が成功する節数に変化が見られなかった。学習は比較的 noise が少ない状態でも行われるが、割合を上げることで学習に対する精度が上昇した。例えば、セクションを表す節 (前述した tag_s15) を学習したとき、noise が少ない状態では、listtop(3,C) のように具体的な値で学習されていたものが、noise を増やすことで numtop(C) と一般化が進むことが確認できた。

4 おわりに

文書構造から帰納学習を用いて構造化文書の自動生成を行った。本研究のシステムにおいて訓練数が多くなくとも文書種別や文書構造の学習が行えることが分かった。また、負事例を被覆する割合をある程度大きくすることによって、学習結果の精度が上がることが判明した。今後の課題として、DTD そのものの学習などの一段階上の構造の学習が挙げられる。

参考文献

[1] XML：http://www.w3.org/XML/

[2] Progol : http://www.comlab.ox.ac.uk/oucl/groups/machlearn/progol.html

[1] 但し、この意味情報は得ているが、処理効率の観点から利用はしていない。

[2] Document Type Definition：文書型定義