28 構文解析結果とその意味表現からの帰納的学習を用いた意味解析規則の獲得

峨家正樹* 荒木健治 栃内香次
(北大工)†

1 はじめに

自然言語処理において文の意味表現を得ることができればさまざまなことに利用することができる。このような表現を得る方法として従来は解析的な手法が多く行われてきた。しかし、解析的な手法では、辞書や文法の構築は人手で行わなければならない。大規模になればそれらを矛盾しないように作成したり、また拡張することは非常に困難なものになる [1]。

このような問題を解決する方法として、大量データから意味表現に変換する規則を自動的に獲得する研究というものが行われている。我々はこれまで帰納的学習によって表層文をさまざまな表現へ変換する規則の獲得を行ってきた [2]。その中には意味表現への変換を行うものもある [3]。しかし表層文から直接意味表現を得るということは、単純な構造の文ではさほど問題はないが、ある程度長く複雑なものに対してはうまく処理できなかった。これは文全体の構造を捉えることが困難になるためである。このような問題を解決する手段として、表層文以外のより高度な入力からの帰納的学習というものが考えられる。本稿では構文解析結果からの意味表現への変換規則の獲得手法を提案する。本手法により、より複雑な文に対しての意味解析が可能になる。以下では本システムの処理過程を示すとともにそれを用いた実験の結果から本システムの有効性について考察する。

2 意味表現と構文表現

2.1 EDR 日本語コーパス

意味表現や構文表現はさまざまなものがある。今回は実験用のデータとして EDR 日本語コーパス [4] における構文解析結果と意味表現の対を使用することにした。よってそれぞれの表現形式もそれに準じたものにしている。以下に EDR 日本語コーパスの構文表現と意味表現について説明する。

2.2 構文表現

EDR 日本語コーパスの構文情報は構文木をリスト表示したもので表されている。Table 1 に例を示す。この中で S, t, W などは節点を表す標識である。形態素ごとに W という標識が付き、それらの形態素同士が構文として結びついているときはそれがどのような関係で結ばれているかを示す合成関係子が付属する。S はこの合成関係子である。また形態素や部分的な構文が主節点であれば t を付加する。

Table 1 構文表現の例

あちこちから拍手が起こる。
(S(t(M(S(t(W 1 あちこち))(W 2 から)) (t(M(S(t(W 3 拍手))(W 4 が)) (t(S(t(W 5 起こ))(W 6 る)))))))(W 7 。))

2.3 意味表現

意味情報は格フレームで表現される。フレームはスロット名とそのスロット値で構成されている。スロット名には main, 概念関係子, attribute, S-attribute, which が入る。スロット値にはフレーム全体も入り、階層的に表現される。

Table 2 意味表現の例

あの男の性格から、真相を突き止めるまではあきらめないだろう
[[main あきらめ] [S-attribute seem] [attribute not] [manner [[main 突き止め] [object 真相]]] [source [[main 性格] [possessor [[main 男] [modifier あの]]]]]]

Table 2 において S-attribute は文全体に対しての話者の視点を表すものである。attribute は文の要素に対しての話者の視点を表すもので、この例では動詞の否定のために not が入る。

3 処理

3.1 変換規則

獲得される規則は Table 3 のようにその規則の中心となる語、構文表現、意味表現で構成される。中心となる語とは、意味表現における main スロットの値のことである。また規則獲得の過程で一般化することができればそれを@1 などのように変数にする。

*gake@media.eng.hokudai.ac.jp
†札幌市北区北 13 条西 8 丁目北海道大学工学部

Table 3 獲得される規則の例

WORD	あ
構文	(t(M(S(t(@1))(W に))(t(M(S (t(@2))(W が))(S(t(W あ)(W る)))))))
意味	[[main あ][place @1][object @2]]

3.2 ルール獲得の処理過程

ルール獲得の処理は次のようになる。

まず既存のルールと入力文の意味表現をそれぞれ分割し、共通部分を見つける。このときの以下の条件が成り立つときに共通とした。

1. スロット名が全て等しい
2. main のスロット値が等しい

このようなときに共通部分を獲得する。差異部分があればそれは変数とすることにより一般化を行なう。

次に構文表現の共通部分を獲得する。既存のルールと入力文の構文表現をそれぞれ構文情報を用いて分割する。分割された構文表現同士を比べ、同じ構造の構文表現に対して共通語が見つかればそれを共通な構文表現として共通部分を獲得する。

こうして得られた意味表現と構文表現それぞれの共通部分の中で対応するものがあればそれを新たなルールとして獲得する。

4 実験

4.1 実験方法

EDR 日本語コーパスの"あ"で始まるものから順に構文解析結果とそれらの意味表現の組 300 を取り出し、それらを用いてルールの獲得実験を行った。本稿でははルールの獲得状況を調べることを目的としている。ゆえに獲得されたルールを実際に適用しての変換処理についてはまだ行っていない。結果 Table 4,5 に実際に獲得されたルールの例を示す。また入力に対して獲得されたルール数を Fig 1 に示す。

Table 4 は日という名詞についての名詞句の規則である。これは"あと 3 日"というようなものから数字の部分が一般化されて作られた規則である。また Table 5 は"する"という動詞についての規則である。

Table 4 獲得された変換規則 1

日
(t(M(W 1 あと)(t(S@1(t(W 日))))))
[[main 日][number @1][modifier あと]]

Table 5 獲得された変換規則 2

する
(t(M(S(t@1)(W を))(t(W する))))
[[main する][object @1]]

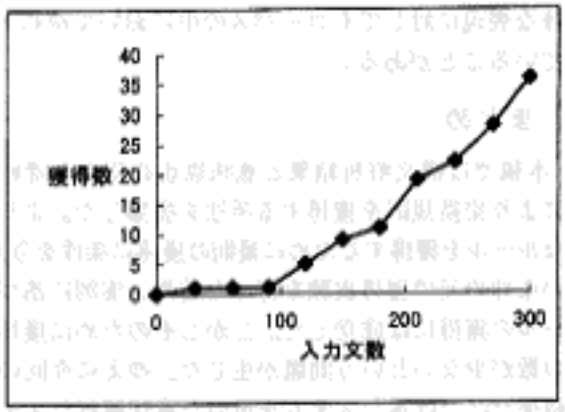

Fig 1 獲得した変換規則の数

4.2 考察

300 文という入力に対して得られたルール数は 36 という結果であった。また、得られたルールは確かに実例に即したルールではあるものの、その全てが構造が単純なものや、長さの短いものであった。また得られた規則は名詞句に関しての規則が大部分を占めていた。この結果について考察する。

このような結果の一番の原因として考えられることは、意味の共通部分の取得の条件の厳しさだと考えられる。main が等しいものしかルールとならないため、名詞句に関してはその名詞の数だけルールが必要であるということである。しかし名詞の数は膨大であり簡単に同じ名詞が出てくるとは限らない。このためそれらのスロットの値が全て一致するような表現はなかなか見つからないのである。

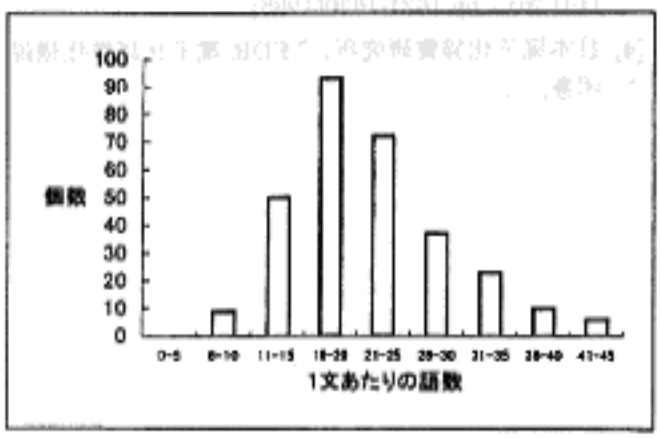

Fig 2 コーパス 1 文あたりの語数

また今回入力として用いたEDR日本語コーパスに関しての問題も考えられる。EDR日本語コーパスのデータにはFig 2で示されているように長く複雑なものが多い。それだけでなく出現する語も多彩である。名詞に関しても莫大な数があり固有名詞も存在する。それが前述した名詞句のルール獲得の障害となっている。このため同等な表現に対してもコーパスの中においてゆれが存在していることがある。

5 まとめ

本稿では構文解析結果と意味表現の組から帰納的学習により変換規則を獲得する手法を提案した。より実用的なルールを獲得するために規則の獲得に条件を与えた。その条件の元で獲得実験を行った結果、実例に基づいたルールの獲得には成功した。しかしそのために獲得ルールの数が少ないという問題が生じた。ゆえに今回の実験の結果だけでは本システムを用いた意味解析システムの有効性については判断できず、その可能性を示すにとどまった。

今後はまず大量のデータを用いての実験を行うことによりルール数の不足を解消して行く予定である。またそれとともに、ルールの獲得に条件およびその方法に更なる検討が必要であろう。最終的には獲得されたルールを用いての変換処理も行う。

参考文献

[1] 石崎俊, "自然言語処理", 昭晃堂 (1995)

[2] 荒木健治, 高橋祐治, 桃内佳雄, 栃内香次, "帰納的学習を用いたべた書き文のかな漢字変換", 信学論 (D-II),vol.J79-D-II,No.3,pp.391-402(1996)

[3] 森英悟, 荒木健治, 宮永喜一, 栃内香次, "帰納的学習による表層文から意味表現への変換規則の自動獲得と適用", 電子情報通信学会論文誌 D-II,vol.J81-D-II,No.7,pp.1621-1630(1998)

[4] 日本電子化辞書研究所, "EDR 電子化辞書仕様説明書", .