2言語の文法対応学習系の学習能力

4Q−13

山口 昌也　瀧口 伸雄　小谷 善行　西村 恕彦
(東京農工大学 工学部 電子情報工学科)

1. はじめに

二つの言語の例文の組を与えることによって、二つの言語間の対応関係を学習する文法推論の枠組みが提案されている。この枠組みは二つの言語の文法を同時に学習し、両方向の言語からの変換を可能にする規則(翻訳文法)を生成する。そして，この枠組みを用いて，二つの自然言語間の翻訳規則を学習しようという試みが，すでになされている[1]。

本論文では、この枠組みの中で、学習の結果得られる文法規則と学習アルゴリズムの評価を目的とする。そのためには，学習対象を自然言語ではなく，より限定された言語を対象とする事が望ましいと考える。そこで，評価は，次のように行うものとする。

- 評価用の2言語間の翻訳文法を定義する。
- 定義した翻訳文法から例文組の集まりを生成する。
- これを作成した学習システムに入力する。
- 学習された翻訳文法とはじめに定義した翻訳文法を比較する。

この過程をいくつかの評価用の翻訳文法に対して，行うことによって，評価を行う。

2. 学習の条件

学習の際の条件を次のように定める。

・評価用の文法の例

学習の対象としては，2言語間の厳密な対応関係を示すことができなければならない。本論文では，通常の四則演算式とそれに対応する逆ポーランド記法とを用いることにする。

・例文組を与える方法

システムに与える例文組は正の例だけを与えることにする。また，学習は，すべてに例文組が与えられたあと，始まるものとし，逐次的には行わない。

3. 翻訳文法

翻訳文法，$G=(V, \Sigma_L, \Sigma_R, S, P)$ は次のように定義されている[1]。

- V ：非終端記号の集合
- Σ_L：言語Lの終端記号の集合
- Σ_R：言語Rの終端記号の集合
- S：開始記号　$S \in V$
- P：規則の集合

$$\alpha_L \leftarrow A \rightarrow \beta_R \qquad A \in V$$

$\alpha_L \in (V \cup \Sigma_L)^+$, $\alpha_R \in (V \cup \Sigma_R)^+$

なお，Aを規則の左辺，α_Lを規則の言語L側の右辺，α_Rを規則の言語R側の右辺と呼ぶ。

ただし，α_L，α_Rに対して，次に示す条件を付加する。

$$N(\alpha_L) = N(\alpha_R) \text{ かつ}$$
$$M(\alpha_L) = M(\alpha_R) = \phi$$

ここで，記号列βに対して，

$N(\beta)$：βの中に含まれる非終端記号の集合

$M(\beta)$：集合$N(\beta)$の要素の中で，重複する非終端記号の集合

と定める。なぜなら，言語Lと言語Rは同じ非終端記号を含まなければならないからである。

さらに，規則の右辺に同じ非終端記号が二つ以上含まれる場合は，2言語間の非終端記号の対応関係が不明確になるからである。なお，本論文では，このような場合，表記上，非終端記号に添字を付けて区別することにする。

4. 学習アルゴリズム

学習は，次の手順で行う。これは[1]をもとにし，主として再帰的文法を学習することを目的として，新たに設計した。

【step1】

開始記号を左辺に持つ，二つの規則 r1, r2 を比較し，

$$diff(r1, r2) = (d1, d2, d1', d2')$$

を求める。これをすべての規則の組み合わせに対して求める。ただし，各記号の意味は次の通りである。

$$\alpha\ d1\ \beta \leftarrow S \rightarrow \gamma\ d1'\ \delta \qquad \text{(規則 r1)}$$
$$\alpha\ d2\ \beta \leftarrow S \rightarrow \gamma\ d2'\ \delta \qquad \text{(規則 r2)}$$
$$\alpha, \beta \in (V \cup \Sigma_L)^+ \cup \lambda,$$
$$\gamma, \delta \in (V \cup \Sigma_R)^+ \cup \lambda$$

【step2】

・もし，$diff(r1, r2) = (\lambda, \lambda, \lambda, \lambda)$ を満たす規則があれば，規則r1を消去する。そして，step1 へ戻る。

・もし，$diff(r1, r2) = (A, d, A, d')$

$$A \in V,\ d \in \Sigma_L,\ d' \in \Sigma_R,$$

という条件を満たす規則があれば，

規則　$d \leftarrow A \rightarrow d'$

を追加し，規則r2を消去する。(r1, r2 は入れ換えてもよい)

さらに，

$$\alpha\ d\ \beta \leftarrow S \rightarrow \gamma\ d'\ \delta$$
$$\alpha, \beta \in (V \cup \Sigma_L)^+ \cup \lambda,$$
$$\gamma, \delta \in (V \cup \Sigma_R)^+ \cup \lambda$$

という規則をすべて，次のように一般化する。

$$\alpha\ A\ \beta \leftarrow S \rightarrow \gamma\ A\ \delta$$

step1 へ戻る。

・以上の二つの条件に当てはまらなければ，step3 へ行く。

Learnability of A Correspondence Grammar Acquisition System
Masaya YAMAGUCHI, Nobuo TAKIGUCHI, Yoshiyuki KOTANI, Hirohiko NISIMURA
Tokyo Universitiy of Agriculture and Technology

【step3】
左辺が開始記号で，右辺の長さの最小値Ｎを求める。ただし，右辺が，１文字の非終端記号の場合は除く。
【step4】
右辺の長さがＮの規則を，

$\alpha_L \leftarrow S \rightarrow \alpha_R$　　（規則Ｒ）

としたとき，次の処理を行う。なお，右辺の長さＮの規則が複数個，得られた場合は，それぞれの規則に対して，step4,5,6 を行う。

(1)　$\alpha\ \alpha_L\ \beta \leftarrow S \rightarrow \gamma\ \alpha_R\ \delta$

$\alpha, \beta \in (V \cup \Sigma_L)^+ \cup \lambda,$

$\gamma, \delta \in (V \cup \Sigma_R)^+ \cup \lambda$

という規則をすべて，次のように一般化する。

$\alpha\ C\ \beta \leftarrow S \rightarrow \gamma\ C\ \delta$

(2) 規則　$\alpha_L \leftarrow C \rightarrow \alpha_R$　を追加する。

(1)を行ったとき，置き換える規則と，置き換えたあとの規則を比較して，異なる部分が置き換えた場所だけの場合は，再帰的な規則を生成する。例を挙げて，説明する。ただし，非終端記号の添字は，右辺に同じ非終端記号が二つ以上あることに対処したものであり，それ以外の意味はない。

$N_1 * N_2 \leftarrow S \rightarrow N_1 N_2 *$　　（規則ａ）

という規則がいちばん短いとき，

$N_1 * N_2 * N_3 \leftarrow S \rightarrow N_1 N_2 * N_3 *$

があったとする。この場合は，次の規則を生成する。そして，規則(a)で一般化するものがなくなったとき，規則(b)を使って一般化を行う。

$C_1 * N_1 \leftarrow C \rightarrow C_1 N_1 *$

$N \leftarrow C \rightarrow N$

$C_1 * N_1 \leftarrow S \rightarrow C_1 N_1 *$　　（規則ｂ）

【step5】
一般化できなくなるまで，step4 を繰り返す。一般化できなくなった場合，step1 の処理を行い，

diff(r1, R)=(A, B, A, B)　かつ，

B　←A→　B，A∈V，B∈V

を満たす規則r1があれば，規則Ｒを消去する。
【step6】
step1 へ戻る。ただし，長さＮの規則に対して，step4 で一度も一般化が行われなかった場合は，次回，step3 を実行する際にはＮよりも大きいという条件をもうける。
【step7】
step3 で最小値が得られないときは，終了する。

5．学習例および考察

３で示した学習アルゴリズムを用いて，図4.1 に示した翻訳文法規則Ｇから生成される例文を入力として，学習を行った。図4.1 の文法は再帰的な規則を含んでいるので，無限個の例文が生成される。今回は，再帰の回数を各規則ごと１回という制限のもと，生成されるすべての例文を入力とした。つまり，次のような例文組を入力した。

a　←S→　a
b　←S→　b
a＋a　←S→　a a＋
a＋b　←S→　a b＋
：
b＊b＊b＊a　←S→　b b＊b＊a＊
b＊b＊b＊b　←S→　b b＊b＊b＊

学習された結果の翻訳文法Ｇ′を図4.2 に示す。

S＋T　←S→　S T＋
T　←S→　T
T＊F　←T→　T F＊
F　←T→　F
a　←F→　a
b　←F→　b

図4.1 基準となる文法規則Ｇ

U＋T　←S→　UT＋
T_1＋T_2　←S→　T_1T_2＋
T＊F　←S→　T F＊
V＋F　←S→　VF＋
W＋F　←S→　WF＋
F　←S→　F
U＋F　←U→　U F＋
F　←U→　F
T＊F　←T→　T F＊
F　←T→　F
V＋F　←V→　V F＋
T　←V→　T
F＋T　←W→　F T＋
a　←F→　a
b　←F→　b

図4.2 学習された文法規則Ｇ′

次に，本学習方式で得られた文法で定義される言語について述べる。入力例文組の集まりを言語L_1，基準となる文法および学習された文法規則で定義される言語をそれぞれＬ（Ｇ），Ｌ（Ｇ′）とすると，次の関係が成り立つ。

$L_1 \subset L(G') \subset L(G)$

$L_1 \subset L(G')$ は，Ｇが入力例文をすべて受理し，さらに無限個の文を生成することからわかる。

一方，$L(G') \subset L(G)$ は次のことからわかる。まず，文法規則Ｇ′の斜字体の非終端記号は，文法規則Ｇの非終端記号Ｓによって，置き換えることができる。さらに，例えば，Ｇ′からは

U＋T＋T　←S→　U T＋T＋

という規則を導出することはできない。しかし，Ｇからは導出できる。したがって，$L(G') \subset L(G)$ が成り立つ。

6．おわりに

通常の演算式とその逆ポーランド記法との対応関係を学習の対象として，その対応関係を示す文法の学習方法について示した。現在，このアルゴリズムに基づいた実験システムを実現中である。

7．参考文献

［１］ 佐藤理史，長尾真：文法推論に基づいた翻訳文法の学習方式，知識工学と人工知能　46-11，pp. 81-89，1986

［２］ Jerome A.Feldman, Geoge Lakoff, Andreas Stolcke and Susan Hollbach Weber, Miniature Language Acuisition : A touchstone for cognitive science, Technical Report of ICSI 90-009, 1990