

帰納的学習を用いた言語行動変換規則獲得手法における 指示命令文に対する適応能力の評価

青山 泰久[†] 荒木 健治[†] 栃内 香次[‡]

[†] 北海道大学大学院工学研究科

[‡] 北海学園大学大学院経営学研究科

本研究はロボットに言語獲得能力を付与することによりロボットがユーザの自然言語による命令文を理解し、行動できるようになることを目的としている。我々はロボットが命令文の意味を理解することとは命令文を外部情報に対しての具体的な行動に変換することと考えている。ユーザが発する命令文が同じでも外部情報および具体的な行動はロボットが用いられる環境により変化するため、このような変化に応じた命令文、外部情報、具体的な行動の関連付けをロボットが学習により自動的に獲得する必要がある。そこで、命令文と行動の関連付けにおいても、言語とそれに対する行動の組の単なる記憶ではなく、様々な表現の変化に対応する適応能力がロボットに求められる。本稿では、学習手法として帰納的学習を用いた場合の適応能力の評価をシミュレーション実験により行ったのでその結果について報告する。

1. まえがき

近年ヒューマノイドロボットの開発はハードウェアとしては製品化されるまでに進展した [1]。一方、自然言語を理解し行動するシステムもコンピュータ内に作られた人工的な世界については数多くの研究が行われてきた [2-4]。しかし、市販に向けた製品化を考えると、特定の環境においてのみ使えるのではなく、導入先においても学習により周囲の環境に動的に適応することができる必要がある。この基準からすれば、自然言語による指示が意味する行動を実世界の情報に適応して学習するシステムは研究されている [5] が実用化されたものは存在しない。したがってこのような研究テーマは今後重要なものであると考えられる。

また、解析的な手法 [4] には、自然言語による命令文を「形態素解析→構文解析→意味解析

→行動生成」と多段階に処理するため、各処理段階での解析の精度が高かったとしても、最終的な精度が下がってしまう問題点が存在する。

我々は、帰納的学習を用いた言語行動変換規則獲得手法 (Language-Action Translation Rule Acquisition Method Using Inductive Learning. 以下、IL-LAT と記述する) を提案してきた [6]。本稿では本手法の詳細について説明する。IL-LAT は前述の課題に対して実例からその中に内在する規則性を抽出する帰納的学習 [7] を適用したもので、人手である程度の正解例を与え続けければ、自然言語による指示命令文 (以下、命令文と記述する) が意味する行動を外部の情報に適応して実行するよう学習することができる。また、入力と出力の組である用例をもとに入力の形態素解析結果から直接行動を生成しているため、入力を出力にするまでに処理する段階数を減らすことができる。用例のみでは用例と同じ命令文にしか応答できない。本手法は用例を拡張することにより未知の入力にも応答することができる。用例を拡張する手法としては、他に言い換え [8,9] による手法も考えられるが、言い換えは他の命令文に言い換えてから命令文を行動に変換する必要がありそこで精度が低下する。その比較を図 1 に示す。また、類似度による方法も考えられるが、

“黄色い箱にキックして”

“黄色い箱にキックしてください”

のような似た命令文に対し同じ行動をすることができても、上記の 2 文に

“黄色い箱にヘディングして”

を加えた 3 種類の命令文を適切に区別して行動することは困難である。本手法ではその違

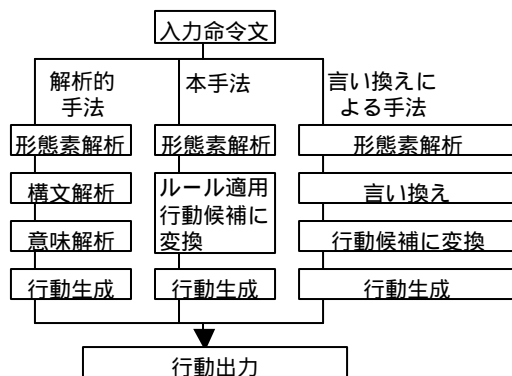


図 1: 他手法との比較

連絡先: 北海道大学大学院工学研究科

〒060-8628 札幌市北区北 13 条西 8 丁目

E-mail: ao@media.eng.hokudai.ac.jp

いをルール組み合わせによる入力命令文との完全一致を出力条件とすることにより明確に区別して学習することができる。

以下、2章では本手法の概要を、3章では本手法に基づくシステムの評価実験を、4章では実験結果についての考察を述べる。

2. IL-LAT の概要

IL-LAT の概要を図2に示す。

ユーザによる入力は形態素解析され、その結果に対しルールを適用し、「位置関係 - 行動列」の組の候補を生成する。その候補の中から、行動処理部が行動を選択し出力する。

その出力結果に対し、ユーザが正解判定を行う。その結果はルールにフィードバックされる。ユーザが不正解と判定した場合または未適用の場合、システムはユーザに正解行動列を入力することを要求する。その後、システムは用例からルールを抽出する。抽出したルールは今後の入力に対して適用される。

2 - 1 ルール抽出部

以下、ルールとその抽出方法を説明する。ルールは以下のような構造をもつ用例から抽出される。

- ・ 命令文
- ・ ロボットから見た目標物の相対的な位置関係（以下、位置関係と記述する）
- ・ 正解行動列

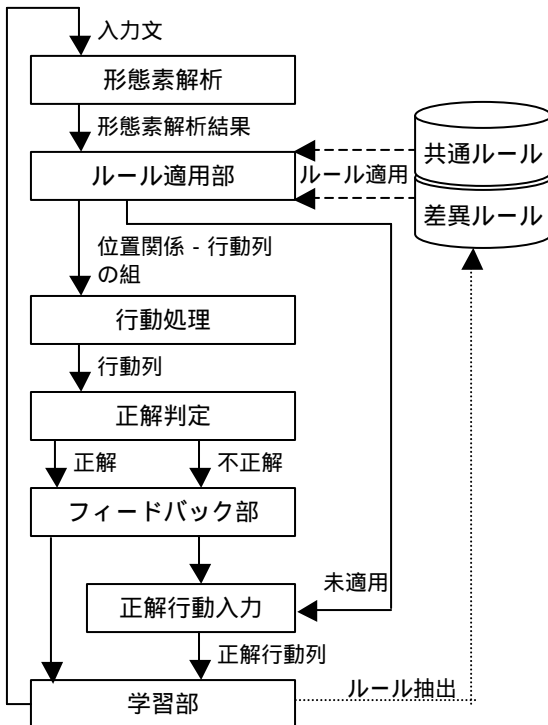


図2：本手法の概要

表1：プリミティブと行動の対応表

プリミティブ	意味
α	前進
δ	右 90 度回転
ι	前にヘディング
μ	左キック
ν	左前キック

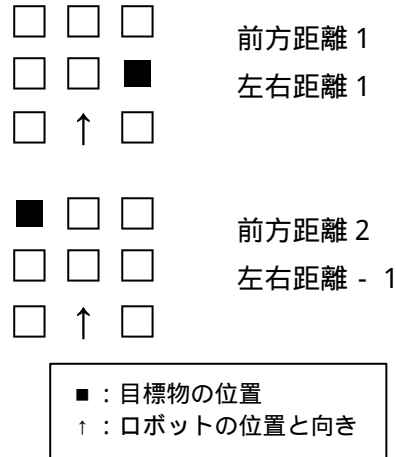


図3：前進距離と左右距離

命令文は茶釜[10]で形態素解析された結果（以下、形態素列と記述する）を用い、位置関係は絶対的な座標とロボットの向きから「前方距離、左右距離」の2種類に変換される。位置情報の前方距離、左右距離は、

- ・ ロボットの位置を原点
- ・ ロボットの進行方向と垂直な方向を x 軸（ロボットから見て右方向を正、左方向を負とする）
- ・ ロボットの進行方向を y 軸（ロボットの進行方向を正とする）

とした座標系においての以下の値である。

前方距離：目標物の y 座標

左右距離：目標物の x 座標

その例を図3に示す。正解行動列はプリミティブ列として人手により与えられる。プリミティブとは表1のようなロボットに取らせる行動を表す $\alpha \sim \omega$ までの記号で、これを「 $\alpha \delta \nu$ 」というようにプリミティブ列とすることで連続した行動を表現する。

システムは現在と過去の用例について各情報の比較を行い、各情報ともある一定の閾値を超えて一致する場合、差異部と共通部から「形態素列 - 位置関係 - 行動列」を持つ差異ルール、共通ルールを抽出する。

表2にその例を示す。また、用例は共通ルールに加える。

表 2：ルール抽出例

	形態素列	位置 関係	行動列
現在	目標/物/に/ヘディング /し/て	1,1	$\alpha \delta \iota$
過去	目標/物/に/キック /し/て	1,1	$\alpha \delta \mu$
共通	目標/物/に/ @0 /し/て	1,1	$\alpha \delta @0$
差異	@0=ヘディング	@,@	@0= ι
差異	@0=キック	@,@	@0= μ

@,@0 は変数を表す

表 3：ルール適用例

	形態素列	位置 関係	行動列
入力	目標/物/に/頭/突き /し/て	-	-
共通	目標/物/に/ @0 /し/て	1,1	$\alpha \delta @0$
差異	@0=頭/ 突き	@,@	@0= ι
生成	-	1,1	$\alpha \delta \iota$

@,@0 は変数を表す

2 - 2 ルール適用部

入力命令文の形態素解析結果に対し、「形態素列 - 位置関係 - 行動列」を持つルールを適用する。以下の条件を満たす場合、位置関係と行動列の変数を代入し「位置関係 - 行動列」の候補を生成する。

- ・差異ルール 1 つと共通ルール 1 つで入力命令文と同じ文を再現可能
 - ・かつ、各ルールにおける行動列部分の変数の個数が等しい
 - ・かつ、一方のルールにのみ位置関係を含む
- 表 3 にその例を示す。

2 - 3 行動処理部

ルール適用部で作成された候補から、以下の条件を満たす候補を選択し行動列を出力する。

- ・入力された位置関係と候補の位置関係が一致
 - ・かつ、使用ルールの平均正解率が最大
- 平均正解率は次のように定義する。

基礎となる文字列 @ j r @ x を @ t r @ x に @ t c @ x に @ t w して下さい @ x に @ t w をしよう @ x に向かって @ j c @ x へ @ j r @ x へ @ t w しろ @ x を @ t c @ x を @ t r @ x を @ t r 下さい @ x を @ t w @ x を @ t w して @ x を @ t w して下さい @ x を見たらすぐ @ t r	@ j c に挿入する文字列 右向け右 進め うしろを向け 右を向け 左を向け	@ t c に挿入する文字列 蹴れ 進め いけ
@ x に挿入する文字列 目標物 黄色い箱 黄色い目標物 目標 黄色いブロック 黄色のブロック 四角い黄色い箱 黄色の箱 黄色の 黄色いの	@ j r に挿入する文字列 まわって まっすぐに行って 後ろにあるいて うしろにふりむいて 左向いて 右向いて 右に回って 左に回って 前にあるいて 前に進んで	@ t r に挿入する文字列 蹴って 頭でたたいて
	@ j w に挿入する文字列 前進 移動 左旋回 右旋回 90 度右に回転 90 度左に回転 180 度右に回転 1 ブロック前進	@ t w に挿入する文字列 キック ヘディング 左前キック 右前キック 前ヘディング 頭突き 右キック 左キック

図 4：実験用入力データの命令文作成に用いた文の生成規則

$$\text{平均正解率} = \frac{\text{各ルールの正解数の合計}}{\text{各ルールの使用回数}^{\ast 1} \text{の合計}}$$

ここで、正解とはユーザがロボットに求める行動列である。また、正解数とはルールを使用したときにその行動列が達成された回数を意味し、各ルールがその値を保持している。詳しくは次のフィードバック部で述べる。

2 - 4 フィードバック部

ユーザの判定に基づいて、使用されたルールを次のように更新する。

- ・正解：正解数を1追加
- ・不正解：不正解数を1追加
- ルール抽出時の初期値は以下のように定義した。
- ・ルール...正解数1，不正解数1^{*2}
- ・用例...正解数1，不正解数0

3. 評価実験

本手法に基づき作成したシステムを用い、シミュレーションによる評価実験を行った。今回シミュレーション実験とした理由は、ロボットの移動誤差によって正解が不正解になるなど実験結果が変化することを防ぎ、変換自体の有効性に実験の焦点を絞るためである。目標物の置かれた場所に近づき{キック、ヘディング、停止}のいずれかの目的に限定した821組の人工的な入力データを作成して用いた。命令文は図4の規則を用いて作成した。この規則は8名の大学院生にロボットの目標物に対するタスクを達成するような命令文を作成してもらい、その命令文から目標物の呼び方、行動を表す言葉を変数、残りを基礎となる文字列としたものである。これらの文を図4に示す“基礎となる文字列”の変数部分@xに“@xに挿入する文字列”のいずれかを挿入する。これをすべての“基礎となる文字列”を“@に挿入する文字列”と組み合わせることにより命令文を生成した。このような方法をとった理由は一定の傾向を持ったデータを大量に生成するためである。相対的な位置関係は図5のような4パターンをランダムに設定した。正解行動列は第一執筆者が付与した。行動列を付与不可能なデータや命令文が明らかに不自然なデータ、他のデータと命令文が重複するデータは作成時に削除した。図2に示すように本システムでは一つの命令文を入力後行動処理を行い、その後学習を行うので入力デ

*1 ただし、初期値を用例については1，ルールについては2と設定した。

*2 現在は仮にこの数字にしている。

*3 何らかの知識を初期状態で与えると入力データと初期データの類似性により性能が変化するので、システムの初期状態を一定にして評価するため空にしている。

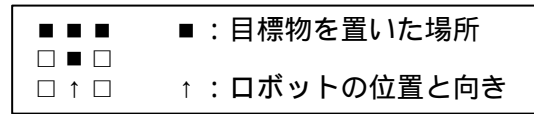


図5：ロボットと目標物の位置関係

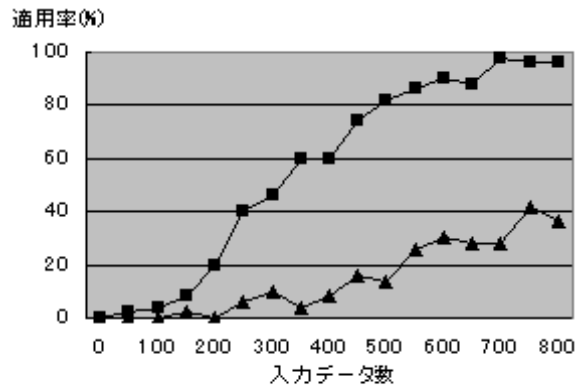


図6：適用率の比較

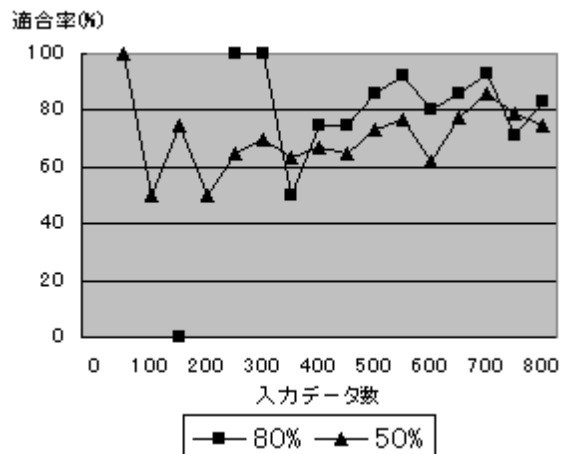


図7：適合率の比較

ータはすべて未知である。

システムは空の状態からルールを獲得し^{*3}、獲得したルールを適用して行動列を出力する。正解はあらかじめ与えた正解と同じ行動列と定義した。ルール抽出条件は以下のように設定して比較を行った。

- ・形態素列：50%以上一致・80%以上一致（実験により変更）
- ・位置関係：両方一致
- ・行動列：50%以上一致^{*2}

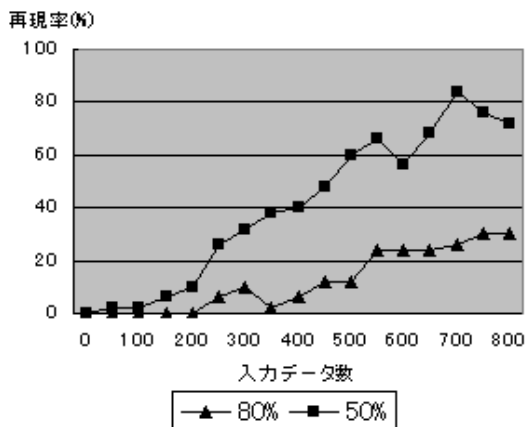


図 8：再現率の比較

4. 考察

抽出時の形態素列の一致条件を 50%, 80% にしたそれぞれの場合の 50 データごと 800 データまでの結果を示す。適用率を図 6 に、適合率を図 7 に、再現率を図 8 に示す。適用率、適合率、再現率の定義は以下の通りである。

$$\text{適用率(\%)} = \frac{\text{適用数}}{\text{入力データ数}} \times 100.0$$

$$\text{適合率(\%)} = \frac{\text{正解数}}{\text{適用数}} \times 100.0$$

$$\text{再現率(\%)} = \frac{\text{正解数}}{\text{入力データ数}} \times 100.0$$

80%, 50% いずれの場合も適用数が増加しているため再現率が増加しているが、ルール抽出条件がゆるやかな 50% の方が適用率・再現率が高いことがわかる。しかし、ルール抽出条件がゆるやかであまいなルールが獲得される分、適合率が低くなっている。より類似性の高い用例からルールを抽出する 80% の場合の方は適用数が増加するに従い適合率が高く安定することがわかる。

5. むすび

本稿では、言語と行動の用例からルールを獲得しそのルールを適用することで行動を行う、帰納的学習を用いた言語行動規則獲得手法 (IL-LAT) の詳細について説明した。また、IL-LAT の評価実験をすべての入力命令文が未知となる実験データを用いて行った。最終的な 750 データ目から 800 データ目までの再現率は 30.0%, 72.0%, 適合率は 83.3%, 75.0% となった。実験の難易度からすれば高い値とはいえない。

しかし、再現率が安定して上昇していることと、すべての命令文がシステムにとっては未知となる入力データに対しての結果であることを踏まえれば、本手法は有効であるといえる。

今後の課題は、再帰的にルールを抽出するより高度な学習を導入し、複数の命令文に対する行動などより複雑な命令文について学習させること、また他者により作成されたデータを用いた有効性の再確認及び、他手法との比較実験である。より広い空間で使用するには位置情報のパターン数を増加させた場合の実験も課題となる。

参考文献

- [1] <http://www.honda.co.jp/ASIMO/>
- [2] 錦見美貴子, 中島秀之, 松原仁, “一般学習機構を用いた言語獲得の計算機モデル”, 認知科学の発展, Vol.5, pp.143-185, 1992.
- [3] 須賀哲夫, 久野雅樹, “ヴァーチャルインファント 言語獲得の謎を解く”, 北大路書房, 2000.
- [4] 新山祐介, 徳永健伸, 田中穂積, “自然言語を理解するソフトウェアロボット: 傀儡”, 情報処理学会論文誌, Vol.42, Number 6, pp.1359-1367, 2001.
- [5] Naoto IWAHASHI, “Language Acquisition by Robots”, TECHNICAL REPORT OF IEICE, NLC2001-61, pp.75-80, 2001.
- [6] 青山泰久, 荒木健治, 柊内香次, “言語とその行動の組からの帰納的学習を用いた言語獲得手法の提案”, 2002 年度人工知能学会全国大会 (第 16 回) 論文集, 3F2-03, 2002.
- [7] 荒木健治, 柊内香次, “帰納的学習を用いた自然言語処理の有効性について”, 信学技報 TL99-41, pp.33-40, 2000.
- [8] 近藤恵子, 佐藤理史, 奥村学, “格変換による単文の言い換え”, 情処学自然言語処理研報, 2000-NL-135, pp.112000.
- [9] 高橋哲郎, 岩倉友哉, 飯田龍, 乾健太郎, “Kura: 統一的かつ宣 9-126, 言的記述法に基づく言い換え知識の開発環境”, 信学技報, TL2001-12, pp.23-30, 2001.
- [10] 松本裕治, 北内啓, 山下達雄, 平野善隆, “日本語形態素解析システム「茶釜」 version 2.0 使用説明書”, NAIST Technical Report, NAIST-IS-TR99008, 1999.