

知識獲得機能を有する 日本語理解システムに関する考察

5K-1

林 隆史 田中 英彦 元岡 達

(東京大学・工学部)

1. はじめに

現在、世界的に見ると、広義の自然言語処理に多少なりとも関係する研究が、種々の分野でおこなわれている。そして、多くの成果が得られ始めている。分野によっては、ほぼ実用化の目途のついた研究もいくつかある。しかし、自然言語を自由自在に操る人間の精神活動の本質に迫ろうとする本当の意味での自然言語処理に関する研究は、まだ始まったばかりである。

ここで考察を行うシステムは自然言語に関する問題(自然言語文の解析・生成、言語能力の発達、発話のプランニング等)を論理学の助けを得て統一的に処理して、学習及び知識獲得によって詳細な自然言語理解能力を得ることを目的としている。

2. 知識の統合化

自然言語理解システムを作成する場合、システムをモジュール化して、できるだけ独立したプログラムを作成しようとするのが一般的である。しかし、人間の自然言語理解のプロセスにおいて、それほど明確なモジュール化がなされているとは考えにくい。このような立場から、ここでは、モジュール化を避けて、データとプログラムの区別もおこなわないことにする。実際には、プログラミング言語としてPrologを想定しているが、Prologには述語論理とプログラム言語という2面性が存在するため、手続き記述、データベース記述、論理式の記述など多面的に利用できる。この多面性を活用することにより、本稿で述べるシステムはその内部で扱うすべての対象を述語論理に基づいたものとして統合的に取り扱うことが可能となる。しかし、モジュール化することなしに、高水準の自然言語理解システムを作成することは非常に困難である。そこで、始めは低水準のシステムであるが、学習及び知識獲得により、言語能力の向上をはかることとする。

3. 述語論理と意味表現

自然言語理解システムを構築する場合、自然言語の意味を記述するための意味表現法が必要となる。意味表現及び知識表現法として一般的に用いられているも

のとしては、意味ネットワーク、フレーム、概念依存文法等があり、最近ではBarwiseらによる状況意味論の研究〔1〕がある。ここでは述語論理による意味表現を用いることとした。

述語論理を用いて意味表現を行うことの利点としては、以下の点が挙げられる。

1. 述語論理には形式的な意味論が与えられ、自然言語の意味を扱う際に見通しがよいと思われる。
2. 述語論理の式はプログラムとみなすことができ、ホーン節に限ればPrologによって実際に計算機上で実行できる。
3. 1., 2. より、対話者(ユーザ)のモデルが容易に構築できる。

論理式によって意味を扱うものとしてはモンテギュ文法があるが、ここでは文の解析・生成プログラムと意味表現に述語論理を用いるという点で、文からラムダ算術で直接論理式を導出するモンテギュ文法とは異なっている。

すでに述べたように、述語論理を扱えるプログラミング言語としてPrologを用いている。Prologは、ホーン節しか扱えないため自然言語を記述するにはあまりにも力不足である。そこでPrologの記述能力を自然言語の意味表現に適するように、いくつかの拡張をおこなうこととする。例として「太郎は東京から大阪へ自動車で行く。」という文について考えてみよう。この文を一般的な述語に翻訳すると、

行く(太郎, 東京, 大阪, 自動車) ①

となる。これと同じ「行く」という動詞を用いた「次郎は沖縄へ行く。」という文は、先程と同じ4項述語「行く」を用いて書くと、

$\exists x \exists y$ 行く(次郎, x , 沖縄, y) ②

となる。ここで現われる変数 x , y は元の文には現われておらず、自然な考え方によると必要ないものである。さらに、このような $\exists x$, $\exists y$ はPrologでは非常に扱いにくいものである。

そこで、述語の記法を拡張してこの問題を解決することとした。それは、まず述語の引数の個数を不定とし、各引数に名前を付け、その名前によって引数の対応をとることとする。この記法では、①、②の述語は各々③、④に書換えられる。

行く (は (太郎), から (東京), へ (大阪),
で (自動車)) ③

行く (は (次郎), から (沖縄)) ④

そして、「太郎はどこへ行くのですか。」という疑問文は、

←行く (は (太郎), へ (X)) ⑤

となり、③、⑤が統合化されて

X = 東京.

が返ってくる。これは、自然言語における漠然性を自然なかたちで取り扱うための一つの方法となると考えている。

4. 学習機能

本システムに於ける学習機能としては、以下に述べる機能を有するものを想定している。

1. 学習によってシステムに欠陥があると判明した場合には、システムの如何なる部分であろうとも修正が可能であること。
2. 学習によって得られた事実が、たとえ誤りであってもシステム全体に重大な影響を与えないこと。さらに、のちにその誤りが検出できること。
3. 学習によって得られた事実が、システム内部の知識と矛盾するものであっても、その事実の受け入れが可能であること。

学習機能の実現方法としては、システム自体が述語論理で記述されているので、Prologプログラムのデバッグ、もしくは自動合成と考えると思っている。Prologプログラムのデバッグ・自動合成に関する研究については参考文献〔2〕を参照されたい。

ここでは、この学習機能の具体的な例として自然言語の解析・生成能力の学習について考える。既に述べたように、文の解析・生成は格文法を基本としたもの

である。そこで、意味素性、及び格フレームの詳細化を学習によって実現する場合について以下に示す。

今、「行く」という動詞の格フレームが次のようなものであったとする。

「行く」 agent : living-object
from : location
to : location

ここで「行く」という動作の主体が人間であるときには、交通手段を用いることができる。このことを示すためにはliving-object という意味素性を持つものなかで、特に人間に関してはあらたな意味素性を与える必要がある。この意味素性をhuman とすると、次の節が生成できればよい。

living-object (X) ← human (X) .

さらに、「行く」の格フレームとしてあらたに次のものが生成される。

「行く」 agent : human
from : location
to : location
by : transporter

このようにして、学習によって言語の解析・生成能力の向上が実現されるものと思われる。

5. おわりに

計算機によって自然言語理解システムを構成する方法の一例を示した。しかし、まだ残された多くの課題を解決していかななくてはならない。今後、今回の考察をもととしたシステムを、Prologを用いて実装する予定である。

《参考文献》

- 〔1〕 Barwise, J. and Perry, J. : Situations and Attitudes, MIT Press, Cambridge (1983) .
- 〔2〕 Shapiro, E.Y. : Algorithmic Program Debugging, MIT Press, Cambridge (1983) .