知的インタフェースにおける自然言語処理

1986

東京工業大学工学部教授　田中　穂積

目次

1　知的インタフェース

知的インタフェースについてお話しします。インタフェースという言葉はコマーシャルなどにしばしば使われ、テレビでもよく出てきますし、新聞の広告などにもときどき出てくるようになりました。お分かりの方も多いと思いますが、機械と人間とが対峙したときに、今の機械が理解できる言葉とか図形は非常に限られています。それをなんとか人間並みにスムーズに、自分が思っていることを機械に伝えたり、また、機械が我々にあたかも人間であるかのごとく答えたり、語りかけてくれる、そういうシステムをつくりたい。それが知的インタフェースです。

いずれにしても、現在のコンピュータはインタフェースの面で行きますと、いわゆる計算機言語を理解していなくてはいけないとか、向こうが打ち出してくるメッセージも、非常につっけんどんなメッセージで、専門家でないとよくわからないという状況にあります。先ほど古川さんが要約されて面白くお話しされたと思いますが、第５世代コンピュータのなかでは、知的インタフェースは一つの大きな研究課題になっています。

これは自然言語によるロボットとの対話の例です。　今からおよそ15年ぐらい前にＭＩＴのウィノグラードという人がつくったシステムです。テレビの画面にこういう絵が出ているわけです。いうなれば、テレビの中に、言葉を理解して何か行動を起こしてくれるロボットがシミュレーションされています。Ｈと書いてあるのは人間の応答を示しますが、　たとえば人間がロボット（機械）に向かって"Will you please stack up both of the red blocks and either a green cube or a pyramid?" と打込みますと、機械が "OK!" と答えて、言われた通りの積み木の世界の操作をする。

指令を出した人間は、機械が自分の思いどおりのことをやってくれたとすると、機械は自分の言っていることを理解したということがわかる。そういう非常にデモンストレーション効果の高いシステムをつくったわけです。

このシステムですが、これは一つの文章を理解するだけではなく、さらに場面がこう変わり、"Which cube is sitting on the table?" などという質問を出しますと、システムは"The large green one." と答えたり、"Which supports the red pyramid?" と質問すると、赤いピラミッドをのせている積み木を答えてくれる。

こういう会話をするためには、今の場合、"the table" といったら画面にはこれ一つしかなく、これに決まっているからあんまり面白くないんですが、"the table" というようなことを言ったら、それがどれかということを計算機はちゃんと認識しなくてはいけない。

さらに "Is there a large block behind a pyramid?" などと言うと "Yes. three of them. "（三つあります）と答えてから、さらに "A large red one, a large green cube and a bule one." と答えてくれます。

これはまたこの状況を変えるためのものです。"Put a small one onto the green cube which supports a pyramid." とやると、"OK!" と言ってから、指定のものを持ち上げて、積み木の世界の状況を変えるというようなことをします。

自然言語による対話によって、積み木の世界がだんだん変わっていったり、積み木の世界の様子を聞くと、機械が答えてくれる、こういうことをするシステムを Winograd が作ったわけです。これがほぼ１５年前に大きな反響を呼んだプログラムの動作の概略です。

このレベルだと、You といえば相手だとすぐわかりますし、それほど高度な推論はいらないのですが、それでもいろいろなことを知っていて複雑なことをしなくてはならない。言葉を理解するというのは、その言葉の文字づらの意味だけではなく、もっといろいろなことを推論して理解しなくてはいけない。

たとえば機械翻訳のことを考えてみますと、"I take a bus." というと、バスに「乗る」というふうに翻訳する。"I take a bath." とすると、takeを「（風呂に）入る」と訳さなければいけない。そういうことをするためには背後に知識を持っていて、目的語にどういうものが来たかを調べてやらないと、うまく訳し分けができない。

そこまではいいのですか、たとえば "The taxi is comming over there." という文章があって、"Let's take it." という文章があったとします。この場合"it" が何を指すかを決めてやらないと、"take" をどう訳していいかわからない。

という具合で、いくつかの文章のつながりを理解することは、いろいろな場面で重要になってきます。ウィノグラードは、こういうことを含めて natural language understanding だと言ったわけです。以前には、 natural language processing という言葉はありましたけれども、 natural language understanding という言葉はあまり使われなかった。ウィノグラードが言いたかったのは、理解の背後にはいろいろな推論があるのだ、その推論が非常に重要だということを言いたくて、あえて natural language understanding という言葉を使ったのだと思います。

たとえば「あるものをこのグリーンの積み木の上に載せろ」ということを言ったとします。そうすると、載せろというのは確かに載せろだけれども、たぶんこれを載せるためにはグリーンの積み木の上に載っているものがちょっと邪魔になる。だからロボットは、邪魔なものを下ろしてから、指令されたものを載せるということまでしなくてはならない。それも一つの推論である。彼のシステムはそこまでやったんです。ですから彼のシステムは、推論の塊のようなシステムであるとみなすことができると思います。

しかしながら、彼の作った対話システムの世界は積み木の世界であって、非

常に限られています。ですから今から１５年程前でもうまくできた、そういうことだろうと思います。

それでは、こういうシステムの中味の構成はどうなっているかを、ちょっとお話ししてみたいと思います。

２　知的インタフェースの構成

古川さんの話のなかに出てきましたけれども、こういうことをするための知的インタフェースの構成は、おそらくこんな形になるだろう。ＩＣＯＴの中でも音声を取上げようという機運にあるわけですが、とりあえずＩＣＯＴは自然言語理解システムを作ろうしているわけです。

自然言語というのは普通の言葉です。コンピュータで言語というと、人はすぐにプログラム言語のことだと思いますから、それではないと言うためにあえて「自然」をつけています。

実は一般的に言って自然言語理解システムの実現は非常に難しい。文章を入力しますと、最初に言語解析をするようなシステムで処理されます。ちょっと注意してほしいのは、その言語解析が終わったあと推論問題解決を行うわけですが、言語解析のなかでも推論問題解決が実は必要になることがあるということです。

2-1 言語解析

自然言語の文章は、言語解析されます。言語を解析するシステムの中身をよく調べてみると、あえて分けるならば、一つは構文解析のレベルがある。それから意味解析のレベル、もう一つ文脈解析のレベルがある。文脈解析は談話解析と言われることもあります。

2-1-1 構文解析

構文解析は文法的な知識を使って、この文章が文法に合っているかどうか、日本語なら日本語の文法に合っているかどうかを調べることです。ですから構文解析が行う主たる仕事は、日本語の文法規則が与えられると、それを用いて、

文を構成する単語の品詞の並びが日本語なら日本語として妥当な品詞の並びになっているかどうかを調べることです。

たとえば日本語では、名詞のあとに助詞が来る。たとえば「テニスをした」という文章は、名詞「テニス」の後に「を」という助詞が、それから動詞がくる。日本語の文法規則では、この並びは宜しいと書いてあるので日本語として妥当な文であるということになる。

ところがそれだけの解析ですと、「チョークをした」という文章があったとすると、これは意味的にちょっと変な文章ですが、「チョーク」は名詞で「を」は助詞、そして動詞がきますから、構文解析のレベルではオーケーになってしまう。それでは困るので意味的にちょっとおかしい文を検出するのが、意味解析の役割です。

普通、構文解析をしますと、構文解析の結果は思ったよりもいっぱい出ます。たとえば「ＡのＢのＣのＤ」というような名詞句があったとすると、「ＡのＢ」でくくるべきなのか、「Ａの」とやって「ＢのＣ」とくくるべきなのか、全部がバラになるのか、くくり具合がいろいろあります。ですから普通、日本語の文章だと、30語から40語ぐらいの単語が並びますが、そういう文章を文法規則を使って構文解析しますと、ものすごい数の構文解析結果が出るんです。それでは具合が悪い。なぜなら意味的に異常な解釈を強制する構文処理結果がそこに多数含まれるからです。意味解析によりそれらを排除する必要があります。

2-1-2 意味解析

ここで「と」についてお話しします。「日本の動物と英国の植物」、こんな文章があったとしますと、我々は「動物と英国」というようなくくりかたはしない。「日本の動物」と「英国の植物」とが並立するようなくくりかたをします。それは、動物と植物とは意味的にちょっと似ている、動物と英国とはちょっと違うといったことから分かるのだと思います。コンピュータでそういうことをするのが意味解析です。それにより意味的に異常な構文解析結果を排除して構文解析結果の数をうんと減らすことができます。

しかし一つの文章を見ていただけではうまくいかないこともあります。先程の代名詞の it などはいい例です。it がなにを指すのかは、前の文章を見なくては決りません。つまりこれは、一つの文章の中での話では解決できず、文の前後をみる文脈解析によってはじめて可能になります。

2-1-3 文脈解析

「動物をかった」という文があったとしましょう。この「かった」というのは、「刈る」とか、「勝つ」とか、「飼う」とか、「買う」とか、いろいろありますが、意味解析の観点からは、前二者はすこしおかしい。「動物」が羊みたいなものでないとすると、「刈る」はおかしい。そういうことで意味解析のレベルで、「かった」は「飼う」か「買う」であると絞ることができる。しかし、そのどちらかということは、前の文を見てみないとわかりません。

たとえばその前に「ペット屋に行った」という文があって、「ペット屋に行って動物をかった」といったら、「買う」の可能性が強い。前にもっと情報があると、どちらか一方の可能性がもっと強くなる。こういうことをするためには文脈を見て意味解析をしなければいけない。それが文脈解析の役割なのです。

古川さんが言った例をあげます。"You are wrong"と言ったら、相手がまた"You are wrong"と言い返したという例です。「お前が悪い」と言われて、言われたほうが「お前が悪い」と言い返しているわけです。もうお分かりのことと思いますが、同じ You が、状況しだいで相手になったり自分になる。つまり You の外延が異なるわけです。この様に状況を考慮した意味解析も文脈解析の範ちゅうに入ります。

2-2 推論問題解決

意味解析した結果を受けて、次に問題解決システムを働かせることになります。知識ベースはそのとき使うわけです。推論し問題解決を行うためには常識とかいろんな知識を使わなくてはならない。構文解析のときは文法と辞書という知識を、意味解析のときには、辞書に書かれた意味的な知識や常識を使います。そして、意味解析結果を使って推論し問題解決を行うために知識を使います。この様にして答が作り出されます。

2-2-1 知識ベース

知識ベースに蓄えられている知識のうち、文法的な知識についてはかなりはっきりしてきています。辞書に書かれている意味的な知識をどの様な形式にすべきかは、少し分かってきましたが、まだよくわからないことも多いのです。文脈解析を行うための知識についてはほとんど分かっていないといってよいで

しょう。解決すべきことがいっぱいある。

2-2-2 図形・画像理解

知的インタフェースのアーキテクチャはたぶんこうなるわけです。これは言語に関する知的インタフェースのアーキテクチャですが、その他に図形とか画像の理解があります。第五世代コンピュータの初期にはこんな計画も考えられていました。

たとえばＣＡＤがあります。ＣＡＤでは回路図を画面に出し、図形を通して図形を編集したり検索したりするシステムが必要になります。その他に書面に書かれた文章を読み込んだり、Ｘ線写真を読み込んだり、回路図を読み込んだりして、解析し、理解するシステムが考えられます。たとえば電球とはどういうものかを一生懸命に言葉で定義しても、それで十分とは言えません。ところがそれを絵で見せると非常にわかりやすいということがありますから、最近は絵入りの辞書も多くなっています。図形・画像理解を我々は何の苦もなく行なっています。ところがそれをコンピュータにさせようと思うと至難の業になります。分からないことがまだまだ一杯あるのです。

普通、画像というと濃淡図形のことを意味します。図形というと、線図形のことを意味します。知的インタフェースには、図形・画像と自然言語を理解する機能がなければなりませんが、これは長期を要する研究テーマです。しかし、日本は、図形・画像理解についての研究は非常に進んでいます。世界のトップであると言ってよいでしょう。

私は古川さんと同じで、前に通産省の工業技術院の電子技術総合研究所というところにいましたが、そこで図形・画像の研究をしている人たちが、外国へ行って帰ってきては、外国の研究レベルはそれほどでもなかったと言っていました。

しかしながら、濃淡図形のデータ量は膨大ですから、処理時間が非常にかかるので、コンピュータも相当高速なものを使わなければなりません。

３　プロログの利用

知的インタフェースの構成について２で述べましたが、自然言語に関する知的インタフェースのアーキテクチャはこの様になっています。こういうものを

つくるのに、私はプロログというプログラム言語を使っています。プロログという言語を使うと、どんな点が便利か、この研究会で東工大の池田君はこのあたりの話を既にしたかと思うんですが、こんなことが言えます。

以前は、構文解析をしたら次に意味解析をして、その次は文脈解析をするという順序で自然言語の解析をすすめたのですが、理想をいえば、人間はこの様な順序で自然言語の解析をしているとは思えません。これら三つの解析を融合していると思われます。そこでもしプロログを利用して自然言語の解析をしますと、三者の融合は驚くほど簡単にできます。これは認知心理学の観点からも好ましいと言えるでしょう。その他にも以下に述べるような利点があります。

3-1 構文解析

これまでは、構文解析をするためには、構文解析用の特別なプログラムをつくらなくてはいけないと考えられていました。入力した文章を、文法と辞書を見ながら解析するプログラムが必要でした。これをパーザと言いますが、そういう自然言語解析用の特別なインタプリタが必要だったんです。

ところがプロログを使って文法を書くと、書いた文法がプロログのプログラムに変換できることが分かってきました。そうすると、そのプロログのプログラムをうごかせば、構文解析ができますから、パーザを作る必要がありません。プロログそのものが、パーザの代用をしてくれるわけです。ですから、この斜線を引いた部分のプログラム（パーザ）をつくる必要がなくなってきた。

3-2 意味処理

私事になりますが、以前にＬＩＳＰという言葉を使って意味処理用のプログラムをずいぶん書きました。ところがプロログを使うと２ケタぐらいプログラム・コードが減ることが分かってきました。

意味解析の方法をよく調べてみますと、その大部分がプロログに組み込まれている機能で置き換えることができることが分かってきたのです。意味解析には難しい問題がいろいろあるわけですけれども、今までですとそれにアタックする前にかなりの準備が必要でした。ところがプロログをうまく使いますと、準備に必要なプログラムをほとんど作る必要がなくなりますから、だれでも自然言語処理の困難な問題に直接アタックすることができるようになってきたと言えるかと思います。

3-3 推論問題解決

推論問題解決を行うときに、推論エンジンが必要になりますが、これもプロログがかなり機能を代行してくれる。この図では全部斜線になってプログラムする必要がないようになっていますが、これは正確に言うとうそで、古川さんに言わせると、もっと皮をかぶせなくてはいけないと思うんですけれども、私が考えているぐらいの小さな世界だと、このあたりはプロログがかなり代行してくれる。

そういうことで、知的インタフェース、特に自然言語の知的インタフェースをつくるという場合に、いちばん大切なのは知識ベースをどう構築するかということになってきます。その場合にもプロログを利用すると知識ベースの構築が自然に行えるという研究が最近進んできました。とくにプロログのパターン照合の機能が知識ベースの検索に有効であるということが分かってきました。以上の理由からわれわれはプロログをベースにした自然言語理解システムをつくろうとして研究を進めています。

〔質疑応答〕

竹内　いちばんはじめに出していただいた文章を生成するというところですが、さわりだけでも結構ですから、何かわれわれの役に立つようなことがあったら教えていただけませんか。

田中　変形文法学者は、深層構造（これは近似的に意味構造とみなしてもよいでしょう）を文法を使いながら変形して線条（単語の一次元の系列）化し、文章が生成されると考えています。文章生成システムの研究は、これまで少しないがしろにされていましたが、機械翻訳の研究が進むに連れて研究が活発化しています。変形文法の考え方を全てそのまま利用することはできませんが、基本的な考えを使うことはできます。ただし、ちょっと気のきいた応答をしたいとか、同じ言い回しは避けたいので省略しようとか代名詞化しようとか、そういう文体を考慮した文章生成は、結構大きな問題になり、文脈解析と同様に困難な問題が発生します。

いちばん最初に言ったウィノグラードのシステムでは、文章生成は非常に簡単です。積み木の世界は狭いですから、応答にしても決まりきった言い回しに

なりますね。そこで、決りきった言回しの一部を空白にした文を用意しておいて、空白に単語を埋め込んで色々な応答文を作り出す方法を使っていますが、これは技術的に簡単な方法ですが、柔軟性に欠けます。

ところで知的インタフェースを作成する上で何がいちばん問題かというと、文章生成ではなくて、文章解析だと思います。文章解析の難しさは、解析結果に様々なあいまい性が含まれることです。あいまい性の解消には常識の利用とか、文脈の利用が最終的に必要になりますが、それをどの様に行うかがまだはっきりと分かっていないのです。解析結果に誤りが含まれていたら、その後をいくら頑張ってもだめです。音楽のレコードでいうと、解析はピックアップの部分に相当しています。ピックアップが安物で、ここから変な信号が発生してしまうと、いくらアンプがよくてもだめである。それと似たようなところがあります。

国藤　本日お集まりいただいた方は、法律のエキスパートシステムをつくろうという人たちですね。そうすると、幅広い質問で恐縮ですが、自然言語処理研究者の立場から見て、法律の文章を言語理解するというか、書くときに、どういう自然言語の文法理論が適用可能であると思われますか。

田中　言語学者がやっている言語理論というのは、必ずしも計算機にのらない。言語学者は自分の頭の中でわかれば、わかったということになるんですが、頭の中でわかったということがどういうことかが分からないとプログラムもできませんし動かないですね。ところで法律の文章はどうなんでしょうか。結構長たらしいですから、解析が楽ではないかもしれません。

国藤　そうするとわれわれとしては、というのは、ここにお集まりいただいた方という意味ですが、法律の文章の論理的な構造をまず明らかにして、それ向きの知識表現とか論理をクリアする。そこからスタートすれば、ノウハウは自然言語処理とかいろいろなところのものが使える。そのように理解したほうがいいでしょうか。

田中　そうですね。

吉野　私は格文法とか、あるいはフレームとか、そういうものを一括して表現するためにプロログが良いと主張しているわけです。法律の世界の言語は限定されており、人工理論に近い言語ですから、主格とか目的格といったものがかなりはっきり決まってくるのではないだろうかと考えています。たとえば財産権というのは、財産権の対象がある。どういう財産権かいう財産権の主体がある。そうすると、財産権という言葉がくるときには、ここは主格で、ここは

性質を表すものであり、ここは客体である。この財産権を指すＩＤ、つまりネーム、論理学でいう変項や定数が来るところですが、そういうかたちで表現できる。

たとえば義務というのはどういうことかというと、する義務とか、しない義務とかいうかたちですから、これがＩＤでここに変数がくるわけです。そして、する義務というかたちで、義務という言葉が出てきたら、そのあとには「する」とか「しない」とかが来るだけだ。そうすると義務の変項のところに「する」が来る。

「する」とはどういうことかというと、普通、行為のＩＤであって、必ず最初にＩＤを置くわけですが、それはある時点でするのだ。ある場所でするのだ。だれがするんだ。だれに対してするんだ。いかにするんだ。そういう要素から成り立っているから、するという行為も、少なくとも法律の世界で取り扱われる範囲内においては、ある程度の型を決めることができるから、これも格文法的に、するという述語の最初の部分は時を表すとか、２番目は場所を表すとか、３番面は主格を表すといったかたちで決められる。

そしてこれは内容というんですが、事実関係が以下のような内容を持っている。内容というのは文の意味ですから動詞で結ばれていて、「した」という内容を持つ。したというのは、まただれがしたか、だれに対してしたか、どうしたか、何をしたか。

そういうかたちで、法律の世界は自然言語の一種ですが、人工言語に近いわけですから、これを分析して一つは格文法的にとらえる。

もう一つは、この格文法の格が実はフレームになっているわけです。財産権というものはこういう四つの？仕事を持っている。ここには物が来るとか、そういうかたちで格文法とフレームとを結びつけたかたちを述語論理的にとらえる。それをプロログで解像的にとらえていけば、これだけで、ユニット節で知識が全部いっぺんに表現できるわけです。

主格のところは必ず「だれだれが」と来るし、目的格のところは「何々を」と来るわけですから、これを自動的にやれば、このプロログを自動的に日本語に生成することもできる。また、法律の文章ですからぎこちないかたちではありますが、マニュアル的に日常言語に書けば、つまりこのルールに従って書けば、それをまた置き換えてこういうプロログも形成することができる。

そういうかたちで私としては田中先生とちょっと違うわけですが、法律の文章について、これは文法だけではなくて世界との対応関係があるわけですから、

世界との対応と文法、論理構造、こういうものを吟味していったらいいのではないか。そしてそれを実験していったらいいのではないか。ＩＣＯＴにも提案しているのですが、私はそう考えています。

法律の世界で義務とか財産権とかいう概念について、条文や判例、いろいろなものに従って言語の経験的分析をし、格が確定できるものは確定していく。スロットの値がいくつあるかを確定できるものは確定していく。それを組み合わせてやっていけば、文と世界、論理、文法、これを統合するような言語理論ができるのではないか。したがって知識処理の方法ができるのではないかと考えています。

これは規範文の論理、形式化のところでお話しする予定だったのを時間の関係で省略しましたが、ご意見があったのであえて述べさせていただきました。専門の先生方がおられるので、この点についてご批判、ご教示が得られれば幸いです。

田中　人間が法律の文章を読んで、その文章に含まれている知識を人間がまず書いてみることが必要だと思っています。法律の文章を計算機で解析してそこに書かれた知識を自動的に抽出するのは、現在の自然言語処理技術では、難しいのではないか。そういうことなんです。吉野先生のおっしゃるように法律の文章で述べられている知識は、述語論理で比較的表現しやすいと言うのには異論はありません。

吉野先生のユニット節による知識表現での一つの問題は、素人には各引数がいったい何を意味しているかよくわからないということです。吉野先生は、ここの引数はなにを意味しているかが分かっていますからいいですが、それが覚えられない人もいます。私どもの知識表現のやり方では、各引数が果たしている役割を明確にするために、各引数を役割名と値の対で表現しています。そしてこの役割名と値の対毎に、ユニット節を書いて行くわけです。こうしますと、新しい役割名が作り出された場合にも、それに対応したユニット節をただ付け加えれば良いことになります。幾つもの引数をもった述語で一つの知識を表そうとしますと、あらたな引数を導入しようとした場合に、それが以前の述語と述語名は同じでもアリティが異なりますから異なる述語として扱われますから、問題がでてきます。ところが我々の知識表現法では、その様な問題は起きないことになります。これは知識の蓄積修正といった観点からは好ましいと考えられます。

加賀山　今、法律の言語処理のところで、インタフェースの問題がありまし

たが、こういうことを聞きたいとかいうときに自然言語が使えたら非常にいいのではないかと考えているんです。その場合に、構文解析という観点から言えば、英語のようにローマ字でわかち書きをしてスペースを入れながら入力していくと、構文解析しやすいと思いますが、日本語の場合には同音異義語が多いですね。だから漢字で入れたいという気がするんですが、たとえば「日本の動物と英国の植物」などという漢字がバッと出てきた場合に、あの構文解析はどのようになされるのか。スペースが入っていれば非常にわかりやすいんですが、ああいうのはどういう具合になされるのか。

田中　わかち書きにはいろいろなやり方があります。もし、漢字が含まれている場合には漢字からひらがなへの変り目で強制的に切れ目を先ずおいて、後で誤って入れた切れ目をなんらかの手段で検出して修正する。どうして漢字からひらがなへの変り目で強制的に切れ目をいれるかといいますと、これでだいたい90%ぐらいがうまく切れるということが分かっているからです。いずれにしましても、この方法で１００パーセント正しく分かち書きができるというわけではありませんが、これが最もよく使われている方法の様です。

吉野　いろいろご議論があると思いますが、時間の関係で、これで田中先生のご報告ならびにディスカッションを終了させていただきます。(拍手)