

## 論説

## 情報学における真と偽\*

清野 武\*\*

## はじめに

コンピュータとか情報処理とかいうものについて、一般社会人はどのように感じ、何を期待し、あるいは何を恐れているのか、情報学の専門家には、かえってわかりにくい面が多く、しかも社会人のいただくもろもろの誤解に対して、専門家にも一半の責任がないとはいいい切れないので、これについて少し考えてみたいと思う。

その材料には、私が今までに、いろいろな場所で、いろいろな機会に、いろいろな人から聞いた話のうちで、多少とも情報学に関連のある節々を拾い集めたものを使うことにする。従って、種々の思い違いや認識不足が含まれているであろうし、また専門家の方々には、たいへん失礼な点もあるかも知れないので、あらかじめお許しを願っておく次第である。

## 統計のおとしあな

京都大学の大型計算機センターでは、毎年、利用者との懇談会を開いているが、その席では、利用者からセンタに対して、さまざまな要望が述べられるのがつねである。ことに昨年は新システムが導入された関係で、比較的事故が多く、これについての苦情が目立ったのも当然と思われた。

このような苦情に対する答弁の中で、センタの一教官から、コンピュータも機械である以上、ある程度の故障の起るのは当然であって、利用者が研究計画を立てるに当っては、そのことも計算に入れておくべきである、という主旨の発言があった。

もしこの席にメーカの技術者がいたら、この発言をどのように受取ったであろうか。利用者からの苦情に

対する防波堤として、伏し拝みたくなるのか、あるいはメーカに対する最大の侮辱として悲憤の涙にくれるであろうか。いずれにしてもこの話には、コンピュータシステムの信頼性、ハードウェアの安定性、あるいはソフトウェアの生産性など、情報工学的ないしはシステム工学的な多くの問題が含まれているので、まずこのあたりから話題を拾ってみたいと思う。

本題にはいるまえに、統計に関する二つの笑話を紹介しよう。

昔、入学試験の監督をしていた一人の教官が、受験票と受験生との照合のついでに、一つの統計をとってみた。それは当時の受験票には、戸主との続柄の記載があったので、長男、次男等の数を調べてみたのである。その結果、不思議なことに、長男が一番多かったというのである。

これによく似た話であるが、テレビで高校野球を見ていたとき、実況担当のアナウンサーが一つの統計を持出したのである。それは過去の高校野球の得点経過を調べたところ、先取点を取って勝ったチームが圧倒的に多いということである。

どちらもそういつわりのないもっともな話なのであるが、これらを称して correct nonsense というのであろう。コンピュータについても、これに近い話があるように思われる。

ひところ計算機センタの能力あるいはサービス状況を示す量として turn-around time (以下 t. a. t.) を使うことが流行したが、今でもバッチサービスを受ける利用者にとっては、重要なデータの一つであることに変わりはない。しかしこれがなんとなく一種の統計量のように思われ、コンピュータシステムの評価に役立つようにも考えられるところに問題がある。事実われわれも、新システムを導入する際に、各ジョブ種別について、それぞれ予定の t. a. t. を保証することをメーカに要求しようとしたことがある。

ちょっと考えればわかるとおり、t. a. t. は多分に運用の様式(たとえばジョブ種別ごとの優先順位)に依

\* The "true and false" in Information Science. by Takeshi KIYONO, Department of Electronics, Osaka Electro-Communication University

\*\* 大阪電気通信大学

この稿は、昭和53年3月22日、京都大学工学部電気総合館において開催された筆者の退官記念講演の内容を要約したものである。

存するものであって、システムの性能だけではきまらないのである。

はっきりいえることは、t.a.t. が安定に維持されている状況は、システムの処理能力が需要を上回っていること、つまりまだ余裕のあることを示しているということ、ならびに、ジョブ種別ごとの t.a.t. の上限は、各種別についての一つのバッチの大きさできまるということ、の二つである。

毎週7日、毎日24時間稼動しても、毎日の需要を処理しきれない状況では、t.a.t. はただ延びるばかりであって、長いところで安定することはないはずである。

t.a.t. が単純な意味の統計的なデータでないことは明らかであるが、これに対して、コンピュータシステムの信頼性の目安としての MTBF は、名の如く平均値であって、ハードウェアについてみる限り、この量は実際の意味をもつものと考えられる。

しかしながら、ソフトウェア的なトラブルについても、簡単に MTBF の概念を持ち込もうとする意見があり、実際にそのような表現を用いているメーカーもユーザーもあるのであるが、これは果してどんな意味をもつのであろうか。

出来たてのソフトウェアは欠陥が多く、はじめのうちはそのためのトラブルが頻発する。それらが修正されるにつれて、トラブルの回数が減少し、形式的には MTBF がだんだん長くなるように見えるであろう。このあたりはハードウェアのふるまいによく似ているのである。

しかしハードウェアの故障は、今まで完全に働いていた部分がダメになるために起るものであり、その部分を交換すれば、完全にもとの状態に戻る性質のものである。これに反してソフトウェアは、良いものが途中で悪くなることはなく、悪いものが自然に直ることは絶対にはないのである。つまりソフトウェア的なトラブルは、本当の意味の故障ではなく、はじめから存在した欠陥のうち、いままでの使用条件では表面に出たこなかったもの、すなわち潜在的欠陥が、新しい条件(その中には、間違った使い方も含まれる)に対して、突然表面化することによるのである。つまり先天的な欠陥部分の露呈なのであって、ハードウェアの初期不良に近い性質のものといえるであろう。

それにもかかわらず、MTBF はソフトウェアのもつ欠陥の個数と無関係ではないから、完成度を示す目安としては意味をもつのだという主張もありうる。し

かしこの辺が統計的數字の危険なところであって、ソフトウェアについてのいわゆる MTBF は、実際にそのコンピュータにかけられたプログラムの性質(ソフトウェアの欠陥に触れるか、触れないか)に極めて強く依存することに注意しなければならない。ソフトウェアは論理であるから、真と偽しかなないのである。

ただ現実の問題としては、たとえば大学の大型計算機センタでトラブルが多く出ているシステムと同じものが、ある企業では何の問題もなく働いている、というきわだった違いが見られるように、潜在的欠陥は、なかなか表面に浮んでこないことも多いのである(後述)。

これとは別の意味で、統計的數字の危険性を示す一つの例を挙げよう。それはコンパイラなどにみられる最適化にかかわるものである。ただしここでいう最適化とは、算術式の中の共通因数を括出すとか、ループの中の不変部分を取り出すとかの、比較的単純なものに限ることにする。

かりに、あるメーカーが最初に第1号のコンパイラを提供し、その後、第2号という改良型を作ってくれたとする。両者の主な違いは、上のような最適化であると仮定する。この場合、最適化の効果を調べるには、多数のプログラムを二つのコンパイラで処理して、所要時間を比較するのが実際のであると思われる。

しかしながら、最適化による計算時間の短縮を、20%とか30%とかの数値で表わすことにはいろいろの問題がある。つまりどんな標本を使って観測したか、によって、見掛けの改善度は著しく違ってくるからである。最適化の余地のないほど上手に書かれたプログラムばかり集めて統計をとれば、最適化による改善度はゼロとなるであろうし、質の悪いプログラムを集めれば、見掛けの改善度はいくらでも高く出てくるであろう。

最適化については、まだまだ議論すべき問題が多いが、主題を離れるので省略する。しかしこの例をもっと一般的に見れば、統計とか平均値とかいう概念の中に、個体の質が埋没してしまうことの危険に対する教訓を読取ることができるであろう。たとえば、教育制度なども、統計的な平均のレベルを基礎に構成されるのが普通であろうから、ずばぬけた天才やその逆の場合は、いろいろな無理が出てくるのは当然であろう。

それはともかくとして、統計や、それに基づく規則性、さらには抽象的、形式的方法論がもたらした利益は莫大なものであるとしても、それが人間の精神に悪

い作用を及ぼす危険性があるとしたら、大いに警戒しなければならない。

“……、またそれは規則性を好み、例外を嫌うように人間を変えてしまう——そのような世界は、われわれの世界の人間性に反するものである。われわれの世界では、例外が規則性よりも大切なのである。例外を好み、それを理解することは、つねに人間のための最も人間的なタスクとして残らなければならない。……” (Zemanek).

情報処理に関する研究、教育上の仕事においても、われわれは無意識のうちに規則性を重んじ、例外を無視することに慣れてしまっている。もしいわゆる“情報化社会”が、このような統計的社会を意味するとしたら、それこそ全くたいへんな間違いといわなければならない。人間の創造的活動とは、それ自体むしろ例外的な現象ではないであろうか。

### ソフトウェアの信頼性と生産性

昭和2年に京大の電気工学科を卒業され、阪大教授から愛媛大学学長となられた故熊谷三郎先生が、京大の電気工学教室懇話会に招かれて、一席の講演をされたときのことである。当時横綱双葉山が全勝街道を薫進していたので、先生は次のような話をされたのである。

双葉山がいかに大力無双であるとしても、われわれ普通の体格の人間が力を合わせれば、必ずや有限の人数によって、彼を倒すことができるであろう。しかし凡庸な頭脳をいくらたくさん合わせても、林重憲先生の演算子法の理論を生み出すことは、絶対にできないのである。

この話は、天才的、創造的発見はむろんのこと、それほどでなくても、ある種の知的活動というものは、数量では決して置換えられないこと、換言すれば、人間をマスとして捕える統計的手法には、大きなおとし穴のあることを教えているように思われる。

大型ソフトウェアの生産も、この種の知的活動とみられるのであって、ここでは頭数をふやして、それぞれに必死の努力を強いてみても、結局はものにならないであろう。

スポーツの世界でしばしば見られるように、個人技の総和以上と思われるような力が発揮される感動的な場面は、まさにチームワークのお手本であるが、ソフトウェアのためのチームについても整然たるチームワークが要求されることは当然である。しかもテ

ームの能力は、その構成員の一番弱いところに強く依存するという事実に注意しなければならない。

蒙古軍が巨大な地域を短期間に席捲してしまったのは、馬の力に負うところが多かったといわれているが、それも第一線から兵站部に至るまで、行軍の速度が揃っていたために威力を発揮しえたのである。

プログラミングの能力に個人差のあることは、他の仕事と同様であるが、適性のない人とある人との生産性の開きは、1:2とも、1:10ともいわれている。しかも困ったことには、1の能力の人を10人集めてチームを作っても、10の能力の人にはかなわないのである。ヨーロッパには、驚くほどの少人数で、極めて信頼性の高い商品を売出しているソフトウェアの会社もあるということである。

ソフトウェアに対する適性の開発、あるいは少なくとも、適性の発見方法の開発は、極めて深刻な問題とと思うがどうであろうか。

ソフトウェアの生産性については、それを作る人の面だけでなく、ソフトウェアの特殊性を無視して議論することはできない。

昔ある役人が、文部省のある会議の席上で、部分的合理化は全体的不合理化に通ずることが多い、という意味の発言をされたことが思い出されるのであるが、これはまさに、システムの問題の本質に触れるものといえるのであって、今日のソフトウェアシステムの生産性の議論における重要な論点の一つなのである。

最近 U. C. Berkeley の Prof. Ramamoorthy が日本で行った講演の中で、ソフトウェアのきわだった性格について指摘している。それは、(1) エラの波及効果、(2) 全体に対する実動部分の比率、の二つに要約される。

第1の性質は、いま述べた局所的改変がシステムの他の部分にいろいろな影響を与えること、別な表現によれば、一つのエラの訂正が新しいエラを生み出さないう保証は存在しないこと、を意味している。

第2の性質は、システムソフトウェアについてみると、のべつに使われている部分は、全システムのうちの小さな部分にすぎないこと、具体的にいえば、全体の時間の90%以上にわたって使われている部分は、システム全体のおよそ10%に過ぎないこと、をいうのである。

つまり第2の性質のために、前述の潜在的欠陥がなかなか表面化しないし、たまたま発見されたエラを修正すると、思いもかけない所に新しいエラが(しかも

潜在的に) 植付けられる, というやっかいな状況におちいりやすいのである。

あるメーカのソフトウェアの工程表に, 一つのモジュールを作るのに, コーディング1カ月, デバッグ2カ月というような計画が記載されているのを見たことがあるが, これはいかにも綿密なテストを前提としているようにも見えるが, このような意識では, とても完成はおぼつかないのであって, 事実, いつまでもトラブルが続いたのである。

実際にわれわれの経験したいくつかのシステムのうちには, 新システムと取換える最後の日まで, ソフトウェアによる事故がみられ, 訂正不能のままほおわれた例が少なくなかったのである。

そんなわけで世の中には, 専門家の間でも, ソフトウェアは永久に完成しないものだ, というさかしらな発言が聞かれるのであるが, このような言葉は, 完全なソフトウェアを作ろうと努力している良心的な技術者の意気を沮喪させ, 逆に, 良心的でない技術者——がもしあれば——をますます墮落させてしまうであろうし, 一方では, 情報工学そのものを侮辱するものである。このような簡では, 学問, 技術の進歩など, とても望めたものではない。

もとより, いわゆるソフトウェアエンジニアリングが, まだ工学的に独立した分野にはなり切っていないことは事実である。だからといって, とにかく何かを作ってみて, その上でテストをやって, まずい所を修正してゆこう, という態度は, 少なくとも大型ソフトウェアに関する限り, 拒否されるべきである。巨大なシステムの問題点を, 個々にもれなくテストすることは, 目的のシステムを作るのと同じくらい, あるいはそれ以上に, むずかしいに違いない。

工場でたいへんな時間と人手をかけて, 綿密, 周到なテストを経たはずのソフトウェアも, 計算機センタに持込んだとたんに, トラブルが続発するという実情をどう考えるべきなのであろう。もはやテスト一辺倒のやり方はその限界を露呈しているのであって, ここに一般化, 抽象化, 証明という基本を追究する必要を痛感する次第である。

### プログラミングと言葉

コンピュータを知らない人々にプログラミングという概念を理解してもらうのは, 非常にむずかしいので, いろいろな誤解が生まれがちである。そのためか, プログラミングはむやみにむずかしいものだという先入

観に捕われるばかりで, 何がむづかしいのか, なぜむづかしいのか, を考えることなしに, 次のようなわなにおちいってしまうことがある。

それは文部省関係の雑誌(文部時報 1972年7月)に出ていたのであるが, 当時の大学学術局の課長の一人が, 情報処理教育についての座談会で話していることの中に,

“現代のコンピュータは未熟であって, ちょっとむづかしいこと(筆者註: プログラミング)をやらないと電算機が使えない。しかし将来は, ぐちで話せばすぐにそれが使えるようになるだろう。だからこういうこと(筆者註: 情報処理教育)に時間を掛けるのはむだであろう, という議論があるんですが。”  
という一節がある。

このように, コンピュータが人のしゃべる言葉を理解すること, あるいはさらに, コンピュータが人間と音声で会話をするというようなことが, 実にすばらしいことであって, これがコンピュータの進歩の究極の目標であるかのように信じている人が少なくないようである。

しかし, コンピュータに向って口で話す, とは何を意味するのか。たとえば spoken program というようなものはどれほどの意義をもっているのか。

もし前述のように, プログラムが非常にむづかしいものであるならば, 書いたものに頼らずに, 頭の中にあるアルゴリズムを正確に, 間違いなく話すことは, もっともっとむづかしいはずである。せいぜい紙に書かれたプログラムを読むのが関の山であろう。

もちろん, 読み書きのできない人でも, 立派に会話ができることは事実であるが, プログラムのように緻密な論理的構成を要する記述に関しては, 事情は全く違うのである。われわれの日常の会話をそのまま文字にしたらどんなものになるかを考えてみるがよい。それは甚だしく非文法的であり, しばしば非論理でさえありうるのである。

それでもなお日常の言葉が, ある程度まで人間同志のコミュニケーションに役立っているのではないか, だから, そのように便利な自然言語でプログラムを書けば, やかましい構文規則などに捕われることなしに, コンピュータが使えるはずではないか, という主張もありうるであろう。前述の文部省の課長もこのことをいいたかったに違いない。

もとより私は, 音声の研究とか, 自然言語の研究とかの意義を否定するものではない。ただ上記のような

誤った観念を一般の人々に植付けることは、情報処理技術のためにも、情報学のためにも、非常な害をもたらすであろうことを恐れるのである。

古くは Goethe が指摘しているように、言葉はしばしば概念に先立って現われ、それによって議論することができ、やがてその概念を明確にする、という生きた力をもっている。また、小さな子供が自分の経験した事柄を母親に伝えようと懸命に話している姿は、まことに印象的であり、言葉のすばらしさを示す一つの感動的な場面と思われる。

このような言葉のもつ特別な力とコンピュータの力とが結付けば、実にすばらしいことになるに違いない、という発想は、多くの人々にかなり魅力的にうつることであろう。

しかしわれわれが、お互に会話を交わし、相手のいうことを理解するのは、単なる論理的帰結ではなく、かなりしばしば、創造と飛躍が働いているように思われる。つまり言葉のもつ力の一部は、それを聞く人の力と対応して、はじめて真価を発揮するのであって、この部分は機械にとっては極度に困難な仕事といわなければならない。

世の中には、人間と会話をするプログラムというものがあるけれども、これはいわば見せかけの知能にほかならないのである。それは、相手の意図を真に理解しないでも、見掛け上は一応つじつまの合う会話がある程度はできる、という事実を利用したものといえるであろう。

さきに述べたように、人間が言葉を理解するには、一種の創造性が必要であるが、翻訳という作業にも当然このことは当てはまる。そして、コンピュータによる言葉の処理として、初期の時代から強い関心を集めたのは、いわゆる機械翻訳の問題である。多くの人々の長年にわたる研究により、機械翻訳は非常な進歩を示しているが、翻訳結果の評価が人間の場合のようによくゆかないために、厚い壁が立塞がっているように考えられる。つまり構文分析等の論理的な処理と、若干の pragmatics の範囲では、あまり多くを期待することはできないのである。

Witgenstein の主張するように、形式的なシステムの内部では、同義語反復的な真のみが存在しうる、とすれば、コンピュータというシステムの中でも、結局は同義語反復の域を越えることは困難であろう。

その昔、翻訳はやがてコンピュータの仕事となるであろう。それが衛星中継を含む通信技術の発達と相ま

って、外国人とお互に自国語で会話を交わすことができるようになるであろう、というような予想が、有名な学者によって書かれ、多くの人々もそれを期待したのである。たしかに通信機もコンピュータもますます小型化されてはいるが、翻訳のほうは、今日の大型計算機システムを使っても、人々がかつて期待したほどには、うまくいかないのが実情であろう。

言葉の問題についての議論はこの程度にとどめておくが、Zemanek が日本で行った“形式的言語”という講演の中から、示唆に富む言葉を紹介してみたい。彼は、“最も知能の高い専門家といえども、自分にわかっていることしかプログラムすることはできない”という事実には注意している。これはあまりに当然とはいえず、いわゆるコンピュータを過信することの誤りは、この事実を忘れるために起るのである。

彼はさらに、“論理学および数学の世界でわれわれが安全でいられるのは、その世界では semantics を形式的に定義することができ、また意味論的な正しさを証明することができるからである”ことを指摘している。これはまさに、広く情報処理の可能性と限界に対する一つの極めて端的な表明であると思われる。

これはまた、“コンピュータの疑似知性的外見から引出されるのは欺瞞だけである”という警告につながるものとみてよいであろう。

繰返していうが、自然言語の研究、あるいは自然言語の処理に関する研究は、学問的にも実用的にも、ますます重要性を加えてゆくことは明らかである。しかしここでいいたいのは、“自然言語の処理”と“自然言語による処理”とは別の問題だということである。

### いわゆる創造性について

コンピュータが自然科学の研究を始めとして、いろいろな創造的活動に寄与することはいうまでもない。しかしここでも、創造的活動に役立つということ、それ自体が創造性をもつということとは、全く違うのだということに注意しなければならない。

たしかに、人間の創造性の極致のように入れられる芸術の世界でも、コンピュータが活躍している。たとえば、コンピュータアートは、美術の世界に一つの新しい分野を開拓したかのようにみえる。しかしここでも、作品はコンピュータを通して作られるのであって、コンピュータによって作られるわけではないのである。

音楽の世界ではどうか。ひところ、いわゆるコンピュータ音楽が、作曲家にとって代るように喧伝され、

コンピュータの専門家も賞讃を惜まなかったのであるが、現状はどうであろうか。

もともと、数学的作曲法には、偶然性を出発点とするものと、既存の秩序をよりどころとするものがあるようにいわれている(カオスとコスモス)。

たとえば、音符の系列を乱数的に発生させ、作曲の規則に反するものを捨てて、一つの曲を組立ててゆく方式がある。このような音楽は、ある意味で新奇であろうが、伝統的な音楽に慣れた耳には、聞くに耐えないものである。

これに反し、過去の何らかのクラスに属する楽曲を分析して、音程の遷移確率を計算し、そのような確率分布をもつ乱数を発生させて旋律を作る方式は、一応耳に親しみやすい楽曲を生出すことが知られている。しかしこれは要するに過去の曲の統計的な平均にすぎないのであるから、一応の安定感はあるにしても、実に陳腐なものしか作れないので、およそ独創性とは縁の遠いものである。

しかしコンピュータの処理能力は、楽曲分析などの面では威力を発揮するはずであり、事実、Mozartらしい音楽とか、Bachらしい音楽などを自由に作りうるということが知られており、たしかにコンピュータの一つの応用として興味あるものに違いないが、これによって創造性を主張することはできない。

人間の知的活動の中には、何らかの発想があって、それから何らかの展開があって、その結果を何らかの基準で評価し、必要に応じてフィードバックする、というサイクルが進行している場合が考えられる。

いま述べた音楽の場合には、発想に相当するところを数学的乱数により、評価のところを単純な論理的判定によって置換えていることになる。しかし芸術の世界は、数学的な評価の基準(評価関数)が存在しないことを特徴とするかのように思われる。ただ作曲の規則に違反しない、という必要条件だけでは、決して音楽にはならないのである。

従って、このような思考のサイクルを機械化するときは、評価のステップが不完全にならざるをえない。それを避けるには、適当に人間が関与するという方法をとればよいのであるが、その瞬間にコンピュータは見せかけの創造性を失って、単なる道具と化するわけである。いわゆるインタラクティブな処理は、ある意味では、始めからコンピュータの知的能力に見切りをつけていることになる。MACのことを皮肉にも、man-aided computationと呼んだ人があるが、計算の

ような形式的、論理的処理においてすら、評価方法がはっきりしないときは、人間が介入せざるをえないのである。

しかし私は何も、コンピュータが道具であっては悪いというつもりはない。現に、今日のコンピュータアートにおけるコンピュータの役割は、画像の創造ではなく、むしろ画像の加工や処理、ならびに各種の出力装置の制御という面に位置づけられ、精巧な道具として立派に活躍している。だから、道具は道具として、正しく評価すべきなのであって、これに欺瞞の衣装を着せることを私は非難しているのである。

### その他の欺瞞性

コンピュータは未熟であるとか、情報工学はまだ学問として確立されていないとかいう議論の中には、しばしば情報とか情報処理とかいう概念について、われわれのいう工学とは、およそかけ離れた期待あるいは要求が前提になっている場合が多いようである。

いうまでもなく、コンピュータは要するに“ビットを扱う論理の機械”である。従って、ビットで表わせないものは情報工学の対象外であり、論理を超えた処理は、始めからできないのが当然である。しかし、もしそれ以上のことができるかのように、一般社会人に思わせたり、錯覚させたりする人があるとすれば、その行為は悪質な欺瞞であるといわなければならない。

一つの昔話を紹介しよう。東京オリンピック(1964年)の組織委員長をつとめられた与謝野秀氏が、その後イタリア大使としてローマにおられた頃のことである。ある晩餐会のあとの雑談中に、大使夫人がつぎのような意味のことをい出された。

“アメリカの政府は、国策の決定にコンピュータを使っていると聞かすが、どうしてベトナムであのような失敗をしたのか。コンピュータとはそんなにダメなものなのか。”

当時ベトナム戦争は泥沼の様相を呈していたのであるが、人間の失敗をコンピュータのせいであるかのように、意識的に国民をだましたとすれば、甚だ巧妙な責任転嫁というべきであろう。

この例にみられるように、コンピュータが自発的にいろんなことをやってくれると考えている人が案外多いのであって、コンピュータを過信または盲信する人も、コンピュータを頭から軽蔑している人も、人間の責任を忘れて議論することが多くて困るのである。そして、社会科学的な問題、あるいは政治的な問題まで、

そこにコンピュータが介在すれば、すべてコンピュータの責任であると思ひ込み、ときには情報学そのものを非難攻撃するのは、全く見当違いの八つ当たりというべきであろう。

もし情報学が非難攻撃に値する要素をもつとすれば、それは出来ないことを出来るかのように錯覚させる不用意な啓蒙的情報の罪であると思われる。もちろん、出来るか出来ないかを、あらかじめ断言できない場合も少なくない。すでに述べた通り、コンピュータはビットを処理する機械であるが、情報の意味をどこまでビットとして扱えるか、言語についていえば、構文的、形式的に扱える範囲はどこまでか、ということが問題になる。

機械翻訳が数学の世界ほど安全でなく、その可能性に限界のあることはすでに述べた。しかしその限界がどこにあるかは必ずしも自明ではない。この問題は画像の処理なども無関係ではないのであるが、私個人としては、やや pessimistic な感じをもっている。ただし私自身はその専門家ではないので、今後の研究の進展を見守りたいと思う。

誇大広告や不当表示はたいへん不愉快なものであるが、コンピュータの世界でも、欺瞞と隣合せの宣伝がないとはいえない。あるメーカーが、プログラミング言語は将来“human language”の方向に進むべきことを示唆しているが、これがもし、前述のように、自然言語(natural language)によるプログラムがものすごく魅力的に感じられるという心理に付け込もうとしたものであれば、甚だ狡智にたけた宣伝といわれなければならない。

いうまでもなく、プログラミングのむづかしさは、プログラミング言語と関係はあるが、本質的には、アルゴリズムの正確な記述ということが問題なのであって、それに適した言語が強く要求されているわけである。このような目的に対し、自然言語が a priori に最良であるという根拠はひとつも存在しないのである。

商業主義的な宣伝についてとやかくいうのはあまり意味がないかも知れない。むしろ恐ろしいのは、ある種の誤解を招く表現が、学問や技術に関して用いられることである。一般に科学的な事柄を社会人に理解してもらうことは、非常にむづかしいので、てっとり早く判ってもらうためには、比喩や類推に頼らざるを得ない。しかし必ずしも善意の短絡とばかりはいえない場合もあるように思われる。

政府が毎年 10 月の始めに開催している“情報化推進週間”というものがあるが、その昭和 49 年の PR 用のパンフレットには“白黒写真をカラーに……”という見出しに対応して、表紙に美しいカラー写真が出ていたことがある。

その仕掛けは要するに、白黒写真(ここでは地図が使われていた)の濃度に従って、人為的に色をつけたもので、学問的には疑似カラー写真といわれるものである。しかし疑似という 2 字を取除けば、たちまち誤解を招くことは明らかであって、もしそのような誤解を当てこんで、情報処理技術ないしは情報化の有用性を宣伝しようとしたものであれば、その根性は商業主義と選ぶところがないことになる。

それがただ素人をだますだけならば、それほど大したことはない、という人もあるかも知れないが、このような感覚が、本当の学問や技術の有用性に対する不信感を一般社会人に植付けることを私は恐るのであって、いまの例も、もし情報学の欺瞞性として受取られたらたいへん困るのである。

世間でいうところの情報公害も、この種の不信から来ていることが絶無ではないと思うがいかがなものであろうか。

## む す び

上に述べたいくつかの例によって、私は、

- (1) 情報学では出来ないことを出来るかのように宣伝して、人をだましたり、誤解させたりして、何らかの利益をえようとすること、
  - を非難したつもりである。このほかの欺瞞性として、
  - (2) 情報学がもてはやされて、例えば研究費等の面で幾分優遇されるようになると、本来の情報学でないものを便乗させようとすること、
- が考えられる。

すでに古くから指摘されているとおり、本来の情報学(Informatik)というものは、user sciences(情報処理を応用する諸科学)とは違うのである。両者は当然、相互刺激的な協力関係に置かれるべきであるが、決して混同されるべきではない。

情報科学が応用を志向して、応用情報学の方向に発展することは望ましいが、応用情報学(anwendungsorientierte Informatik)と、情報処理の応用とは別のものである。

これとは別の現象として、

- (3) 情報学の責任でない事柄を、情報学あるいは

コンピュータの責任として非難攻撃し、たとえば“情報公害”などと称して、情報学の発展を妨害しようとする事、  
が挙げられる。

昔、情報処理学会の大会で“コンピュータを過信するな”というパネル討論が行われたことがあったが、そこで取上げられたのは、ほとんど末端的なミスに対する非難であった。専門の学会としては、むしろ“コンピュータを過信させるな”という感覚で企画すべきではなかったかと思われる。

ここで示した第2、第3のタイプの欺瞞性は、断乎として排すべきであるが、第1のタイプには、むしろ内部的な欺瞞性というべきものも皆無ではないかも知れないので、自らいましめなければならないと考える次第である。

最近来日した MIT の Prof. Minsky が人工知能について講演し、その一部が新聞にも報道されたが、ひ

と昔前にみられた楽観的な予言にくらべて、今日の人工知能の研究は極めてじみちであり、その上に立っての予測も、はるかにひかえめになっていることが目立っている。

この傾向は、むしろ喜ぶべきではないであろうか。というのは、夢と希望をもつことは必要であるとしても、大風呂敷は有害無益だからである。

ひところ“情報化社会を主体的に生きる”という言葉が流行したが、今私は、この言葉の意味を、情報学の可能性と限界、換言すれば、情報学における真と偽を、謙虚に認識すること、にはかならないと考えている。そしてさらに、この真と偽を見分ける拠り所を与えることは、ほかならぬ情報学の任務でなければならぬと思うのである。

(昭和 53 年 5 月 24 日受付)