IMPLICATIONAL MOLECULES I A METHOD FOR EXTRACTING MEANING FROM INPUT SENTENCES

Robert P. Abelaon & Carol M. Raich
Yale University

Summary

A generally overlooked possibility for extracting pragmatic meaning from sentence inputs is discussed. This possibility draws upon tha psychology of tha subjective attribution of causes to stimulus events. Application to the design of conversational programs ia feasible, although the praaant work concerns application to the aimulation of a belief system. Simple sentences known by the system ara grouped into categorical types. An "implicational molecule" ia a meaningful aet of sentence types with linked elements, for example: A doea X, X causes Y, A wants Y. One important way in which meaning can ba ascribed to an input sentence is to use it as the basis for a molecule which ia then "filled" by other information in tha system. This notion ia defined mathematically, and a programming algorithm outlined. The possibility of recursive use of a molecule-filling routine gives tha process heuristic overtones.

Programs for the analysis of natural language have probed intensively into syntactic and semantic problema, but we are still juite a long way from the achievement of an "understanding machine". In free on-line discourse in English between persons and some of the conversational computer programa deviaed to date,[1*2] experience has suggested to the investigators that the programs indeed miss a great deal of the "meaning" of what is said.

Although the papera in other aeasiona of this conference offer persuasion to the contrary, in our opinion what ia lacking ia not primarily in the area of clever ayntactic or even semantic analyaia of the input proae. Rather, the most difficult problema lie in what has been called the "pragmatice" of language — the penetration beyond literal decoding of eentencea to the construction of plausible implications from those aentencea. Recently there haa bean much interest in social paychology in so-called "attribution theory".* This rafara to situations in which an individual ia given information about another pereon'a behaviors toward oertain objects and muat infer the appropriate causal influances underlying the behavior, Tha individual may attribute enduring personal traits to tha other, or enduring incentive propertiee to tha object, or a purpoaive orientation of tha pereon toward tha objacta in particular aituations, and ao on. It ia clear that tha attribution prooaaa ia a crucial part of tha pragmatic analysis of language, and that conversational computer programs with attribution capability would ba quite interesting.

A hypothetical example may clarify what auch a program might do. Suppose an individual types •n on-line sentence auch aa, "I went to three drugstores," A syntactically baaed program might respond, "How did you go to three drugstores?" A semantically based program might respond, "What useful things did you buy in three drugstores?" But a pragmatically based program ought to be clever enough to ask, "How come tha first two drugstores didn't have what you wanted?" In order to achieve thia response, the program would probably have to be equipped to understand that repeated actions signify persistent intention, that drugstores ara places where things are intentionally bought if available, that thia generally requires no extraordinary skill or resources from the buyer, that persistence generally connotes either marginal ability or environmental difficulty, and that therefore the buying intention met with availability difficulties in the drugstores.

Even such a simple example is pitted with problems, and one ahould note that the inference that at least two drugstores didn't have what the person wanted might be incorrect. No pragmatic program can be correct all the time, and it seems quite unlikely that a good program would turn out to make more intelligent attributions than people would — and people can, of course, be wrong* Any such program at thia atage of technology would also be at the disadvantage that ita English vocabulary must be sharply limited in order for ita operation to be manageable.

The notion of inherent program fallibility does not coexist comfortably with the ideals of artificial intelligence work, but I leave that problem to the artificial intelligence worker. As a psychologist, I am somewhat more interested in the imitation of real intelligence than the construction of ideal intelligence.

Thia paper ia concerned with an idea that may prove extremely uaeful in the design of programa for pragmatic analysis. Tha crucial concept is what I call an "implicational molecule". We have employed thia concept in the atudy of simulated belief systems, though it could also ba uaed in conversational programs, and would ba capable in principle of responding intelligently to examples such as tha 'three drugstoras' problem,

Implicational Molecules

Implicational molecules are seta of aentencea which ara "bound together" by psychological implication. Tha implications which bind moleculea have general applicability to claaaea of aentence elementa rather than having a domain only of particular aentence elements. There would not be one aet of implicationa for John and another for Joe except insofar aa John and Joe might be categorized aa individuals with behaviore explainable by very

different principles. If John were a "good guy" and Joa a "bad guy", for example, than thair behaviora might ba aaan aubjactivaly aa obaying diffarant ragularitias, and tha aat of molaculaa would ambody implicationa partaining to Good Guya and to Bad Guya.

Varb elements aa wall aa noun elementa ara groupad into oatagoriaa for purpoaes of molacula definition* Rapraaantativa oatagoriaa might ba Poeaeaaion, Wanting, Doing, Liking! Pravantion$_t$ Hurting, and ao on, all rapraaanting kinds of relations batwaan apaoifiad oatagoriaa of santanoe subjacts and aantanoa objecta.

A aat of ganaral aantanoaa gains a molacular charaotar whan tha aantanoa alamanta ara chainad batwaan aantanoaa ao aa to oaptura a particular pragmatic implication* For axampla, whara A ia an actor, X an action, and Y an out coma, tha aat of thraa aantanoaai A Doaa X, X Causes Y_t A Wanta Y, conatitutaa a molacula axpressing tha idaa of purposive action : individuals do things in order to achieve desired outcomes. The particular structure of sentence terms ia important. The set A Wanta X, X Causae Y, A Doaa Y, ia a collection that doaa not make much sense* In the specification of a molecule, therefore, all the components are interdependent* The alteration or permutation of category memberehipe will generally spoil the meaningful character of the sentence set* This does not preclude, however, the possibility that a given sentence type can appear in mora than one molecule. For example, the sentence 'A Doaa X[9] from the Intention molecule could also be involved in a 'Servitude' molecule: B Bosses A, B Wanta X, A Doaa X*

Implicationa! molecules may have interesting properties in connection with a number of areas of human thinking learning, memory, inconsistency resolution, and others.[4] Many of theae properties ara poetulated in Geatalt theory, particularly aa formulated by Haider,[5] with the molecule acting aa a kind of cognitive Geatalt* For present purpoaaa one property — the completion tendency — ia moat important. Given all sentences of a molecule except one, there ia a tendency for thia omitted aantanoa to ba inferred It ia in thia aenae that I have apoken of implication or attribution. If wa ara told that A doaa X, and X causes Y, we ara likely to infer that A wanta Y* Even if wa ara told only that A doaa X, wa may find it plauaible to believe that there exists some Y which X causes and which A wanta.

By meana of molecule completion, therefore, it ia poaaible for extra sentences to be adduced from given aentencea, that ia, for pragmatic meaning to ba aupplied to the givens* Thia idea ia the baaia for tha computer program routine I will describe shortly. First it will ba useful to detail tha aetting within which tha program operates.

Simulation of a Belief Syatam

A program built to use implicational molecules of course must have acceas to a memory store or dictionary of moleculea. It ahould ba obvious that thia atore of moleculea ia not fixed by a universal principle. Some implicational moleculea may be operative in one cognitive system but not in another* Dramatic examples may be provided by conaide ring implicationa a paychoanalyat might draw from information about human aotiona and feelinga in oontraat to a layman[1s] implicationa. One auch funny little molecule ia the fear-wish paradigms A Faara X, A Wanta X. Or this might be embellished to include the meaning of dreams A Feara X, A Wanta X, A DreamsY, Y Stands for X* Aa another case in point, a politician might often interpret public events by different principles than would the Man in tha Street.

The programmer, then, must provide moleculea to the system, appropriate to its purposes. He must also specify the elemental vocabulary of the system, and assign theae elements to the categoriea used to define molecular aentencea. The category of "actors", A, for example, in the foregoing illustrations, is a aat of particular noun elements denoting the individuals, groups and institutions cognized by the system. Verb elements are alao subject to categorization, ao that each verb category auch aa Doaa, Wanta, etc*, might contain a number of apacific verb forma. Finally, the memory of the system ahould contain a sat of particular aentencea "known" to the ayatem — theae are ita apacific "beliefs".

At this stage of program development, it ia not raaaonabla to contemplete human-sized elemental vocabularies, or seta of categoriea, or lists of beliefs, or stores of molecules. The full richness of unconstrained natural language cannot be handled at tha same time that tha pragmatic analysis problem ia being tackled. It waa decided to limit the range of topical concern of the ayatem, and to standardize the syntax of stored belief sentences in the most rudimentary Subject-Verb-Object construction* Even at that the memory of the ayatem needs to be very large.

However, it is quite eaay to produce a great variety of output constructions by maana of keyed program exits. It might be possible to accept moderately complex input which ia preprocessed into simple sentence form by some auch maana aa tha DEC and REC functions of Waizenbaum'a ELIZA program,[1] but we have not done thia. Inatead, tha input ia limited to the aimpla sentence forms cognized by tha central ayatem*

In exercising a belief ayatem auch aa deacribad, our intent haa been to design a program which will operate upon input aentencea in auch a way aa to understand or explicate them. Part of this explication would involve a response of credulity or incredulity, along with aubjective reasons why tha input ahould be believed or disbelieved. Thia aspect of tha program ia what in previous papers[6,r] I have called The Credibility

Teat. Tha remaining aspact of tha program consists in tha Attribution Procaaa — tha discovary or construction of pragmatic implications from credible inputs,

Tha output from this program can in principle serve to determine further system activities, such as changes in the stored memory structures or the formulation of questions to elicit further information from the input source, etc. At present, however the output to each input aentence triggers no further processes. A response is printed and the eequence terminates unless another completely independent input sentence is presented to the program.

Interaction with our program thus need not be in the conversational mode. We have found that the frustrations of attempted time-sharing! when the program involves string and list processing and a very large memory! can outweigh the potential advantages. Our present program version! written in SN0B0L3, is batch processed on the 7094.

The particular vocabulary concept categories. beliefa' and molecules of one realization of the system concern the topic of United States foreign policy. There are some 500 nouns in 15 categories 100 verbs in 11 categories. 300 beliefs, and 12 molecules. The ideology presently represented is the extreme right-wing point of view of a well-known ex-ex-Senator. Other points of view could of course be handled in the same programming system by changing the memory store. Our system design intention was that the replies by the automated Senator to the input statements presented him should simulate the replies the real Senator might conceivably make. Early tests of such a simulation were not notably successful! but more recently with the inclusion of the implicational molecule concept in the simulation system$_f$ a fair degree of success has been achieved. The details of this teat are to be reported in a separate paper. How implicational molecules are used in the simulation program will now be described.

Using Implicational Molecules in the Simulation

Tha heart of the simulation program is a sub-routine which attempts to "fill molecules", The input is a single sentence (in the simple Subject-Verb-Object form), and the output is either a filled molecule containing the input aentence and the appropriata other sentences needed to complete the molecular set, or a failure signal indicating that tha system cannot explicate the input sentence. Following failure to explicate! the sub-routine may be entered again in an attempt to explicate the opposite of the input sentence$_t$ thus enabling the syatem to distinguish between unacceptable and inexplicable inputs.

If the molecule-filling routine succeeds, then of course the other sentences comprising the completion of the molecule provide the verbal materials for the explication. Interestingly, the "name" of the molecule also provides valuable •xplicative material for use in embellishing the output. To exemplify this, consider an input sentence of the form, "A Does X". Suppose it turns out that this is explicable via the purposive action molecule: that the particular x performed by this particular actor, a, causes a particular y which a wants. The name of this molecule may be considered to be "PURPOSE (Y,A,X)", and the program may plausibly include in its verbal output the notion that y was a's purpose when he did x.

The details of the program concern the heuristic procedures by which other sentences are collected alongside the input sentence to fill a molecule. These procedures are entirely algorithmic when certain depth-and-sequence-of-search parameters are arbitrarily fixed at simple values, although the necessity of fixing these values to achieve algorithmicity indicates that the process is essentially heuristic. In presenting the program details here, we will adopt simplifications to emphasize the algorithmic aspects.

Table 1 indicates the notation. A vocabulary of V elements are each assigned to one or more of N categories. Categories of noun elements are indicated by $X_i(i=1,2,....I)$, of verb or relational elements by $R_i(i=I+1, I+2,....N)$. Individual noun elements are denoted by $x_j(j=1, 2,....J)$, and verb elements by $r_j=(j=J+1, J+2,....V)$. Category assignment specifications $(x_j \varepsilon X_k, X_l,.....)$, $(r_j \varepsilon R_k, R_l,.....)$ are associated with each element. Ordinarily each element will be a member of only one category set, but this need not necessarily be the case.

There is a list L of "legal sentence types", xaRpxceach specifying a semantic ally and pragmatically reasonable subject-verb-object sequence. Nonsensical combinations of categories, such as inanimate subjects and animate verbs, do not appear. Nor do subjectively unbelievable triplets — for example, our simulated right-wing politician does not allow for the possibility of a Free-World-Nation Attacking a Free-World-Nation, and this sentence type does not appear on its list L.

There is a list B of all specific belief sentencea^i $x_{j_a} r_{j_b} x_{j_c}$ the system. This list is partitioned into blocks of belief sentences of each given legal sentence type (for example, all specific examples of Communist-Nations Subverting Free-World-Nations). This simple blocking arrangement has been found to save a great amount of time during search and matching procedures using the belief list.

Finally, there is a list M of molecules. Each molecule is a particular set of s legal sentence types (most typically, s=3) together with a specification of how particular elements are to be linked across sentences. In general, the molecular set Mh may be denoted^t

$$M_h \equiv (X_{i_1} R_{i_2} X_{i_3} ; X_{i_4} R_{i_5} X_{i_6} ; X_{i_7} R_{i_8} X_{i_9} ;....)$$

with specific identifications of subscripts, for example, i1=i7the index for the category, Actors;

$i_3 = i_4$ = the index for the category, Actions; $i_6 = i_5$ = the index for the category, Outcomes; i_2 = the index for the verb category, Doing, etc..

The abstract molecule Mh nay be said to be raaliaad by a concrete case whenever all sentanoaa in the abstract molecule spacification ara appropriataly exemplified by sentencee which ara preaumed trus. A santanca asaumas this truth value, $PTRUE_t$ aithar by virtua of baing tha input to tha "fill-molecule" routine, or bacausa it is on tha baliaf list, or bacausa it is calculatad to ba appropriata to add to tha baliaf list. This lattar calculation involves a subtlaty to ba diecusaad shortly*

In tha process of attamptad concrata realizations of a molecule, tha aavaral catagoriaa X_9R acquira alamantal representative x,r. Tha catagoriaa may ba conaidarad "free variables" until tha program algorithm assigns spacific alamants to craata "fixed variables". This tarminology ia naoaasary bacausa of tha linkage of X-variablea across sentences. It is a convantion of tha program that whenever the aama X-category appears in different sentences, the elements representing these categories should either be identical or highly similar. ("Highly similar", in tha apecial senae intended here, means that one element ia an instance of the other. These instance relationships ara stored in the memory. Details have been given in previous papers. 7,8,9)

If all free variables in a molecule become appropriately fixed, the molecule ia successfully filled. The conditions of such success are apparently quite straightforward mathematically. Dejot ing the contents of the concrete molecule

$$m_{h_k} = (x_{j_1} r_{j_2} x_{j_3} ; x_{j_4} r_{j_5} x_{j_6} ; x_{j_7} r_{j_8} x_{j_9} ; \ldots)$$

where $x_{j_1} \in X_{i_1}$, $r_{j_2} \in R_{i_2}$; $x_{j_3} \in R_{i_3}$, etc.,

a realization of the abstract molecule M_h is achieved if $x_{j_p} \sim x_{j_q}$ whenever $i_p = i_q$ (the symbol $\sim$ denoting "identical or highly similar") and a truth function t calculated on each sentence of m_{h_k} yields the value PTRUE:

$$PTRUE = t(x_{j_1} r_{j_2} x_{j_3}) = t(x_{j_4} r_{j_5} x_{j_6}) = \ldots.$$

The apparent straightforwardness of this condition suggests that the program algorithm might be rather simple. While there are some complicatiens involved in trying to accommodate molecules with varying numbers of sentences and varying element linkages across sentences, and there is a certain amount of iteration over multi-element lists, most of the essential steps are indeed straightforward. First, categories are assigned to the input elements. Then from list M a molecule is found which contains a sentence of that categorical type.

Next, the free variables of the categorical type sentence are fixed to correspond to the input, and this fixedness is transferred to the appropriate element of another sentence type in the same molecule. Acceptable concrete sentences of this type are generated from the appropriate block of the belief list B, thus fixing another free variable. At this point, one remaining sentence in the molecule typically has both X elements fixed, and the value of the PTRUE function is calculated for it. Success either completes the molecule or, in case sentences remain in the molecule specification, new free variables are fixed by concrete sentence generation and/or the PTRUE function is calculated for new fixed-variable sentences•

The PTRUE function involves one or two steps. The first is a matching procedure against the appropriate block of the B-list, looking for a known sentence with all elements "similar" to the tested sentenoe. ("Similar" is a more relaxed version of "highly similar", mentioned previously.) If this test fails, a second procedure is undertaken, a procedure which lends ths whole routine much greater interest.

This procedure introduces recursion in ths molecule-filling routine. The second way in which the function PTRUE can be satisfied for a sentence is by entering that sentence in turn as input to the filling routine. A successful exit endows the sentence with presumptive truth. The existence of this realistic program device means that the attempt to complete molecules assumes an open-ended character, with partial completions leading to new searches which may have partial completions, and so on. There is undoubtedly a psychological limit to the depth such a recursion might reasonably take, and a presently arbitrary program parameter is used to define this limit. Hopefully, empirical work now in progress will shed light upon the appropriate value of this parameter.

In order to make the foregoing abstract explanation of the routine more concrete, Table 2 gives some of the categories, elements, and molecules for the right-winger. The components shown of course represent only a small fraction of the total storage. Notice that noun elements can bs compound concepts such as situations-helpful-to-the-Communists. One of the molecules, the crucial one for present purposes, is labeled "No-Win Policy" by the system, and has the four argumentst Liberals, Western governments, situationa-helpful-to-the-Communists, and standing-up-to-the-Communists. The sentences of the molecule arei Weetern governments promote situations-helpful-to-the-Communists} Standing-up-to-the-Communiats prevents situations-helpful-to-the-Communists1 Liberals control Western governments; Liberals fear standing-up-tc-the-Communists•

Now suppose that the input sentence is "Britain not-interferes-with Czech invasion." The sentence type is "Western governments promotes situations-helpful-to-the-Communists." The program consults its moleoule list, and finds that ths No-Win Policy molecule contains a sentenoe of this

type. Second molecule sentences are now generated by consulting the belief list to locate sentences of the type, "standing-up-to-the-Communists prevanta situationa-helpful-to-the-Cfcmmunists." Not all aantancaa of this typa are appropriate, however, since the specific sentence object muat be similar to the fixed variable of the first sentences Czech invasions. Thus, "Allied airlift overcomes Berlin blockade" is not appropriate. A list of candidate sentences is constructed, and we suppose that the first such sentence is "U.S. shows Of force stops Red invasions." For the third sentence of the molecule, the required sentence type is "Liberals control Western governments." Here, the "Western governments" variable is fixed by the input as "Britain", but "Liberals" is a free variable (because it has not occurred before). Suppose "Wilson controls Britain[11] is the only appropriate sentence of this third type.

Passing finally to the fourth sentence of the molecule, we see that the type, "Liberals fear standing-up-to-the-Communists" has been constrained by fixing the liberal as "Wilson" and standing-up-to-the-Communists as "U.S. shows of force" or something similar. Now the program instead of generating possibilities, must simply see if anything at all on the belief list fits the specification. If, say, "Wilson fears U.S. nuclear actions", is present, then the molecule is successfully filled. The output can now be constructed by chaining together the four realized sentences of this molecule, the name of the molecule, "No-Win Policy" and suitable pre-keyed interstitial material to produce a florid and smarmy account of why Britain does not interfere with the Czech invasion.

If, on the other hand, the belief list containa nothing to suggest that "Wilson fears U.S. shows of force"! then this sentence itself is input to the fill-molecule routine. Another molecule is found which will accommodate a sentence of this typei say, the "Liberal Weakness" molecule — "Liberals fear atanding-up-to-the-Communists", "Liberals oppose armaments", "Armaments promote standing-up-to-the-Communiets." If the second and third sentences here can be realized, say with "Wilson opposes British troop commitments" and "Britiah troop commitments aid U.S. shows of force", then "Wilson fears U.S. shows of force"would be considered presumptively true by the program, thus filling the original molecule.

Failure of thia recursion leads to an iteration down the earlier list of candidate sentences in an attempt to find another suitable combination. Exhaustion of all candidate lists leads to abandonment of one molecule and attempts at filling others, afterr which the final failure exit occurs.

With a program limitation on deep recursion, this routine usually takes no more than half a minute. Of course this sstimate depends on the parameters of the particular memory, and we will need more experience to derive more general estimates.

References

1. Weizenbaum, J. ELIZA - a computer program for the study of natural language communication between man and machine. Comm. Assoc. Computing Machinery, 1966, 9, 36-45.

2. Colby, K. M., & Enea, H. Heuristic methods for computer understanding of natural language in context-restrictedon-line dialogues. Math. Bio-sciences, 1967, 1, 1-25.

3. Kalley, H. H. Attribution theory in social psychology. In D. Levine (Ed.) Nebraska Symposium on Motivation, 1967. Lincoln, Nebraska: University of Nebraska Press, 1967.

4. Abelson, R. P. Psychological implication. In R. P. Abelson, E. Aronson, W. J. McGuire, T. M. Newcomb, M. J. Rosenberg & P. H. Tannenbaum (Eds.) Theories of cognitive consistency: A sourcebook. Chicago: Rand McNally, 1968.

5. Heider, F. The psychology of interpersonal relations. New York: Wiley, 1958.

6. Abelson, R. P. Computer simulation of 'hot' cognition. In S. S. Tomkins & S. Massick (Eds.) Computer simulation of personality. New York: Wiley, 1963.

7. Abelson, R. P., & Carroll, J. D. Computer simulation of individual belief systems. American Behavioral Scientist, 1965, 8, 24-30.

8. Abelson, R. P. Heuristic processes in the human application of verbal structures in new situations. Proc. XVIII International Congress of Psychology, Symposium 25. Moscow, 1966.

9. Abelson, R. P., & Kanouse, D. E. Subjective acceptance of verbal generalizations. In S. Feldman (Ed.), Cognitive consistency. New York: Academic Press, 1966.

Table 1

Notation for Fill-Molecule Program

Categories: $X_1 X_2 \cdots X_{15} R_{16} R_{17} \cdots R_{25}$

Elements: $x_1 x_2 x_3 x_4 \cdots x_{88} r_{89} r_{90} r_{91} \cdots r_{127}$

Assignment specifications: $x_1, x_2 \in X_1$; $x_3 \in X_1, X_2$;

$r_{89}, r_{90} \in R_{16}$; $r_{91} \in R_{17}$, etc.

Legal sentence types: $L=(X_1 R_{16} X_2;\ X_1 R_{17} X_{15};\ X_1 R_{22} X_3;\ X_2 R_{20} X_{15};$ etc.)

Belief list: $B=(x_1 r_{90} x_4;\ x_2 r_{89} x_3;\ x_2 r_{91} x_{88}; \ldots)$

(blocks of beliefs)

Molecule list: $M=\{(X_1 R_{16} X_2,\ X_1 R_{17} X_{15},\ X_2 R_{20} X_{15});\ (X_1 R_{16} X_2,\ X_1 R_{22} X_3,\ X_4 R_{22} X_2,\ X_4 R_{16} X_3);$ etc.)

Molecule realization: $m=\{x_2 r_{89} x_3,\ x_2 r_{91} x_{88};\ ?)$

In general, a realization of abstract molecule

$$M_h=(X_{i_1} R_{i_2} X_{i_3},\ X_{i_4} R_{i_5} X_{i_6},\ X_{i_7} R_{i_8} X_{i_9}, \ldots)$$

is achieved by a subsumed concrete molecule

$$m_{h_k}=(x_{j_1} r_{j_2} x_{j_3},\ x_{j_4} r_{j_5} x_{j_6},\ x_{j_7} r_{j_8} x_{j_9}, \ldots)$$

if $x_{j_p} \neq x_{j_q}$ whenever $i_p=i_q$ and $x_{j_1} r_{j_2} x_{j_3} \in B$ or otherwise satisfies PTRUE and $x_{j_4} r_{j_5} x_{j_6} \in B$

or otherwise satisfies PTRUE, etc.

Table 2

A Specific Example for the Fill-Molecule Program