Session 8: INFORMATION PROCESSING AND LINGUISTIC ANALYSIS

QUESTIONS AND DISCUSSION

G. BROWN: First, I will call on Professor Oswald.

OSWALD: I just wanted to say that Dr. Edmundson has a very quiet way of exploding a bomb and I am not sure that everybody in the audience has begun to feel the fall-out. What he said was supposing we are able to perform automatic translation ideally and at fantastic speeds, how do we know what we want to translate? The whole question of determining in advance by some automatic process what we may then wish to subject to an elaborate translation program to get a very acceptable form of English is a terribly important question.

MATTHEWS: I have several questions for Dr. Oettinger. You gave an example of what you call translation from one notation to another notation. I think it was merely a notation inversion. The phrase structure of both of these notations was essentially the same and, in fact, these notations show unambiguously what the phrase structure is.

OETTINGER: Yes, that is the whole point. The simplicity of the pushdown store notion in this connection is I think the part that has not been obvious. If you look at Kitov's version of the Lyapunov version of the Rickenhouser paper published in 1958 -- this very simple problem is tackled with a sledge hammer consisting of hunting for inner parenthesis and right-most, first-left parenthesis, and so forth. This requires that either you have the whole input formula in storage at once or you rock tapes back and forth. Now, the fact is that these things all have the same structure; indeed they are all representations of the same thing, they are isomorphic. The whole point is that they are different and this, if Dr. Edmundson will forgive me, is the way trees come in. They are isomorphic precisely in the sense that they all represent the same tree, if you want to stick functors, at the main connective at the apex and at subsequent connectives underneath and finally the variables at the very bottom. But there are many ways of

representing this one structure, and this leads to different notational problems. The invariance, is reflected in the fact that all the algorithms going in any particular direction share this pushdown store business and share the delta m theorem. They differ in ways that one can slightly attribute to the way in which information about the structure is given explicitly in the notation. In the parenthesis-free notation the functor that you first hit is the main connector--it warns you right away what to expect. You know its degree, and it tells you exactly what comes afterward. If you have the same tree represented in the parenthetic notation, you do not know until you have somehow found the main connector, what the main structure is. I submit that this, although there is an obvious isomorphism that from a very practical point of view, is a major difference. A demonstration that a very simple technique is sufficient to account for this very simple isomorphism is fine I think.

EDMUNDSON: What is the role of the delta m theorem?

OETTINGER: If you have a formula split into three parts of which the middle part is well formed, for example, something of the form ($x_1 + x_2$); when you are about to work on this piece there will be something in the pushdown store. I distinguish the front end as delta h, the middle as delta m, and the tail end as delta t. So when you start working on delta m you will start putting things into the pushdown store as a function of delta m. The theorem states that when you are through with this middle part, everything that you have put in the pushdown store as result of this middle part, will be gone again.

EDMUNDSON: Have you formalized the operations which you have carried out? Because you are really talking not only about the language, but also about a metalanguage. What are the formal rules in the metalanguage ?

OETTINGER: The metalanguage is a new one that is due to Professor Iverson. The only place where it has appeared so far is in a report to the Bell Telephone Laboratories who support some of the research at our laboratory. It will appear in a book on automatic data processing

that Professor Iverson is finishing for the press. This formal language lends itself to the description of algorithms with a degree of simplicity and precision that I don't think is matched by anything that has been available to date.

EDMUNDSON: Will this appear in the presentation of your paper in the proceedings; will some portion of Dr. Iverson's notation appear?

OETTINGER: No, it would take me about four pages to describe the notation and this was not possible.

STOCKWELL: This is a question directed from a linguist to a linguist on a linguistic matter. It seems to me the linguistics underlying MT has to be sound if there is any relation between them at all. In the field of linguistics there are quite obviously two major professional attitudes toward research at this time. One of them insists that one's methods, procedures and so forth for arriving at a decision be explicit, statable as procedures, repeatable by other investigators on the same data, getting the same description from the same procedures. The other attitude would suggest that only the results must be explicit and well formed, that we don't care how we get the description so long as it works well, this is all we can ask for at this date. In Professor Garvin's paper, he listed two procedures -- presumably explicit repeatable procedures -- for doing analysis. One was called "dropping" and the other was "substitution". The substitution technique in linguistic analysis assumes that there is an effective procedure for choosing the correct frames for making your substitutions. Such a procedure to my knowledge does not exist. Second, it assumes that there is some basis other than the intuitive judgment of the investigator or his informant for deciding after a given substitution in a frame whether the resulting expression is an instance of a right substitution. That is, whether the internal relationships are the same after the substitution as they were before it. All criteria for correct substitutability presuppose that the new string is the same in grammatical form as the original string, but the notion of sameness in grammatical form is precisely what the grammar is supposed to explain, not what is it supposed to take for granted. I can illustrate

this in a discussion of nominals from English. Let us say we set up a frame "blank is good". "We can get the "boy is good", "the man is good" and so forth. But then we would reject "the man is tall and is good", "I don't know whether he is good", "They said that the man is good", and so forth. These would obviously be rejected as being the same in form. But it cannot be objected that these are not the same as "the man", "the boy", or what have you because knowing when the substitution is a right substitution is precisely the fact of knowing just which are the nominals of English. In the event that members of this group wish to pursue this question further, and to give proper credit to the sources from which I have drawn for my own remarks, I should like to call attention to the fact that Noam Chomsky has discussed the substitution technique thoroughly in his Logical Structure of Linguistic Theory, and that Robert B. Lees has briefly but effectively made these same points in an article which I have seen in manuscript, Linguistically Oriented Grammars and the Substitution in Frames Technique.

G. BROWN: Does anyone want to counter that?

GARVIN: I think I ought to retreat into Dr. Oettinger's position and say that the proof is available in a paper, except that in linguistics we have no proofs. All we have is verification and degrees of plausibility. There is a discussion of exactly the same point in one of my previous papers. The one at the international congress of linguistics in Oslo where I said that the substitution technique is obviously workable if there is someway of defining the frame. I don't think this is too difficult because you can use relations other than substitutability for this and, of course, the criterion for rightness would then be that, if there are any relations present in the original utterance then, the utterance after the substitution should exhibit these same relations as defined by some other criterion. A very trivial example would be that if you are dealing with a unit where the major criterion is, let us say, fixity of order as happens in the morphology of certain simply constructed languages like American Indian not English. Then you could also say that all those things which are substitutable for each other within the same fixed order chain are functionally equivalent and the equivalence is there defined by the order relation which is the

criterion for right substitutability. I don't quite know what to do about English for the simple reason that I have never been able to devote enough time to analyze it. I would also say that just because the ordinary garden variety techniques of linguistics have not always been refined by everybody is no reason to reject them. The field of English is a brilliant example of the failure of linguists to apply techniques instead of their intuitive judgment. There are conferences every year about English where people argue about such important problems as whether or not there are 10 vowels in English or only 9.

MATTHEWS: Maybe Dr. Garvin can make his point a little clearer if he would say what he meant by the difference between the probability or generative grammar model, and the definitional model in which there is some discussion of levels. Why is it that the generative grammar is not a definitional model when actually it does define what sentences are and what the language is?

GARVIN: I will retreat again, I will say that the generative grammar is not a definition in the classical sense. That is, it does not define the nature of language by giving a genus and a differentia. It attempts to define particular sentences as to whether or not they are or are not sentences. There are apparently some arguments among the practitioners of the definitions as to whether or not it does or does not apply to any one particular given sentence other than those that he himself has generated. I am referring to the probability matrix model by Charles F. Hockett in his manual of phonology where he says literally, "this is an as-if mode of thinking". Now, my own simple little model goes straight back to Aristotle--or perhaps pre-Aristotlitean times -- since people have always made the cogent observation that language stands for something other than itself. Now, you can give it a very beautiful psychological definition, if you so desire it. Instead of saying that a sign is something that stands for something else, you can say that a sign is a stimulus object, the response to which is not governed by its substantive physical properties but by a convention pertaining thereto. If you will excuse me, I would also say that this kind of translation into an isomorphic but less intelligible terminology would probably be handled extremely well by the

technique that Dr. Oettinger proposed for translating parenthesis notations into parenthesis-free notations without making any difference to the content. I am, unfortunately, less interested in terminology in models than I am in finding out what there is in a particular language. Once you know what there is in it you can always find some isomorph or description to represent it properly.

MATTHEWS: I am a bit surprised that you, of all people, tell non-linguists that the American Indian's language are somewhat simpler in structure than English.

GARVIN: This happens to be my experience. It took me two field trips to Idaho to describe Kutenai morphology with a precision which, I think, is probably more definitive than any existing morphological description of English. I would go one step further and say that I believe that from a great many points of view Russian is easier to describe than English. I wrote a paper on that called "The Relative Tractability of Morphological Data". Some linguists deny that some languages are simpler to describe than others, but in this, as in so many other things, I follow George Orwell.

KIRSCH: I have a question for Dr. Oettinger about the delta m theorem. You said, and I think correctly so, that if the middle part of a three-part expression is well formed, then before and after that decoding process the pushdown store will stay in the same condition. Now suppose, on the other hand, that this middle part is not well formed. I believe that the net result of this would be to wreck the whole transliteration process and that, consequently, just the opposite type of situation (admittedly in a different context) occurs to that which you suggested. Namely, that a non-well-formed subpart of an expression will ruin the whole transliteration.

OETTINGER: If the thing is not well formed you can not put something illformed in and get something well formed out. The point is simply that you have many chances each time you open a new nest, and that any nest that has not been tampered with before and that is itself well formed will come through.

KIRSCH: Your answer entirely skirts the main question of what happens when there is an error in formation of a subpart and how this will propagate itself.

OETTINGER: It will not propagate beyond the boundary of the next well-formed nest. This is the essence of the delta m theorem; it says that everything stops.

KIRSCH: An important question is what is the maximum amount of information that would ever occur in the pushdown store? This is related to what you might call the logical depth of the parenthesis notation, and I believe a rather important practical question which the linguist should be interested in. I believe that Professor Yngve has some opinions about the rather small size required for the pushdown store, and that is a practical consideration which I rather wish Dr. Oettinger had mentioned.

OETTINGER: Professor Yngve recently stumbled on the pushdown store as something having to do with sentence generation. Saul Gorn at the University of Pennsylvania has stumbled on it and so have I. The maximum depth of nesting in English being seven, then, somehow the length of the pushdown store should be fairly limited. If there is anything I can say about natural languages, it is only an approximation and, at this date, a guess that must be empirically verified, and I have not done the verifying. Let me state first what is the story about the model. The internal storage required by these algorithms is independent of the length of the formula. Now, in language, all I can say is that what I have said here today is only a first approximation model to predictive analysis. It is clear that the prediction pool is not a cleanly behaved thing the way the pushdown store was; the right thing is not always at the top. The reason for this is that there are deviations in natural language from this ideal pattern that you have in these artificial simple languages. Now every bit of empirical evidence that we have had to date in the practical analysis of Russian sentences suggest that this dependence on depth of nesting, rather than on length of sentence, holds true for a natural language.

KIRSCH: Is it correct to say, as in automata theory, that the problem of taking all well-formed Łukasiewicz expressions and converting them from one notation to the other, is a problem which no finite automaton can actually solve? It requires a machine which, in principle, has an infinite amount of storage and therefore cannot be a machine.

OETTINGER: Yes, I think you are right to a certain extent that theoretically you can certainly have a sentence or a Łukasiewicz-type expression with depth of nesting larger than any finite number you care to name. In this abstract sense it is certainly an infinite state machine rather than a finite state machine. However, in computing machines one tends to be fantastic because the magnitudes involved, when you are dealing with depth of nesting, grow at a much smaller rate than the length of the formula. If Professor Yngve and George Miller are correct and natural languages do, in fact, never exceed a certain depth of nesting, a finite pushdown store will always be adequate It can be large but it will always be smaller than the length of the sentence. One has every reason to believe that outside the kind of pathological sentences that Chomsky cooks up to prove an abstract point, nobody in his right mind will utter an infinite nested sentence.

EDMUNDSON: In addition to no-one in his right mind, no-one in his wrong mind will utter such a sentence -- for mathematical, not linguistic, reasons.

MCDANIEL: I don't know if Dr. Oettinger formed his example for the sake of simplicity, but it seemed to me that it was only valid for commutative functors where a functor b is the same as b functor a. Can he assure us that this holds for non-commutative functors?

OETTINGER: I believe so. What you may have noticed is a permutation in the order of the subscripts of the variables as they appeared in the parenthesis expression as opposed to the parenthesis-free expression. This is a phenomenon that has already been noted by Burks, Warren and Wright in their paper on evaluation. I would like to make one further point. In a discussion some time ago with Professor Yngve it became clear to me that an algorithm for translating from parenthetic to parenthesis-free notation that he presented in COMIT at the Paris meeting last June, had not only a tree isomorphism, but

also an operational isomorphism to what is done here. When I talked about this at the University of Pennsylvania, P. Z. Ingerman (to whose paper the ACM Proceedings I referred to in the write-up) assured me that his technique bore strong similarities.

HAYS: I would like to challenge Dr. Oettinger's frequent references to fail-safe. He said, over and over again, that the system is fail-safe for Łukasiewicz notation. This is a little bit of a misnomer because that system is perfect for that.

OETTINGER: No! Not for what I described here today. You have to put some complications into this algorithm to make it fail-safe. This is one made to be as simple as possible, just to give the idea of how you do these things in the pushdown store.

HAYS: But I think that the result you referred to was that if the formula is well formed you got a translation, and otherwise you get the result that is not well formed. Is that right?

OETTINGER: Yes.

HAYS: Then in that sense it is perfect.

OETTINGER: Yes.

HAYS: So we would not say fail-safe for that. We would not use fail-safe for this in a natural language because a natural language is not Łukasiewicz language. In fact, two functors which have the same representation of a natural language can have different numbers of arguments. Is that right?

OETTINGER: Yes.

HAYS: So you cannot go through with the pushdown store and go through the sentence one time only and get a result.

OETTINGER: Yes. I said that I was talking on the theoretical plane

and that is approximate. Mr. Sherry in his exposition shows the use of his so-called end wipe in his prediction pool. This is something that you would never need in an ideal Łukasiewicz language. It is our little gimmick to insure that nests will end when they should and not stay around to plague us when they are over. Normally, in the ideal case, the prediction is right at the top and because each functor has a finite and fixed number of arguments there is never more or less than what you need. But you are absolutely right in a natural language. Say I have a noun, it can have any number of adjectives corresponding to a functor of some arbitrary degree. The first adjective sets up a prediction for a master, if there is a noun afterwards you know that you are at the end of the nest. If you do not have a noun then this dangling prediction stays in the hindsight or somewhere else to plague you. You think you have completed this nest but you really have not because there is an argument left over, and in that case you would go back and question your final choice of the word as an adjective, because if it is an adjective it must have a master. If it does not have a master then maybe it is not an adjective. When you can not make a prediction vanish by other means you use the end wipe which has the property that any time you have not satisfied a prediction above the end wipe in the prediction pool. When you go through the end wipe they all get erased above it indicating that you have assumed that this is the end of the nest. If it is wrong then hopefully you have something in hindsight. If you have nothing in hindsight then you may be in one of those instances where this method fails, as it will in some instances. It is a practical method as opposed to a theoretical one.

HAYS: I think that this question of the variable number of arguments for functors with the same representation means that Mr. Mersel in his use of what I think is mainly Dr. Garvin's grammar code is going to want more than three codes meaning "we will get this", "we will not get this", "we don't know", but that, in fact, he would like to have a code meaning "we should get it, but not necessarily" and others -- so that he can represent disjunctions of conjunctions of complements.

OETTINGER: Yes, but instead of having this everywhere this is what Mrs. Rhodes' urgency numbers or prediction span indicators do. They tell you something about the degree to which you expect

things. You can shade it as fine as you want. Absolutely, maybe anything you want. In Dr. Edmundson's terms, we are now in a probability logic.