16

Question-answering in English

S. Isard and H. C. Longuet-Higgins Department of Machine Intelligence University Edinburgh

Abstract

The problem we consider in this paper is that of discovering formal rules which will enable us to decide when a question posed in English can be answered on the basis of one or more declarative English sentences. To illustrate how this may be done in very simple cases we give rules which translate certain declarative sentences and questions involving the quantifiers 'some', 'every', 'any', and 'no' into a modified first-order predicate calculus, and answer the questions by comparing their translated forms with those of the declaratives. We suggest that in order to capture the meanings of more complex sentences it will be necessary to go beyond the first-order predicate calculus, to a notation in which the scope of words other than quantifiers and negations is clearly indicated. We conclude by describing a notational form for connected sentences, which seems to be a natural extension of Chomsky's 'deep structures'.

INTRODUCTION

In this paper we shall consider the problem of when an English sentence, or a series of sentences, provides enough information to answer a question, also posed in English.

John kissed Mary

Did John kiss Mary?

(1) (2)

(4)

The sentence (1) obviously enables the question (2) to be answered in the affirmative, and transformational grammar partly accounts for this by giving formal criteria by which a declarative sentence and a question can be recognized as having the same underlying structure. But transformational grammar does not explain why (3) and (4) provide an affirmative answer to (5) whereas (3) and (6) fail to provide an answer to this question.

John saw a flying saucer (3)

Mary saw it too

Did John see the flying saucer that Mary saw?(5)Mary saw one too(6)

Neither does transformational grammar concern itself with the notion of logical consequence, whereas we want to understand why 'Socrates is a man. All men are mortal.' enables us to answer the question 'Is Socrates mortal?'.

Various systems of formal logic raise an analogous problem – at least for 'yes/no' questions – within their own languages, and solve it by formalizing the concept of logical consequence. Thus in the first-order predicate calculus, we could say that a set of sentences Σ 'contains enough information to answer the question posed by' the sentence ϕ if either ϕ or its negation is derivable from Σ . The formalization may be either in 'syntactic' terms, involving derivations, or in 'semantic' terms, which appeal to the interpretation of sentences in a particular set theory; and in many cases we may not have an algorithm which will enable us to decide whether Σ answers ϕ . But at least there is a definition of what it means for Σ to answer ϕ ; we know what task there is no algorithm for.

It would seem that we might make substantial progress if we could translate English into the language of some logical system, and there is a widespread feeling that such a translation ought to be possible, for at least a large and important subset of English. Indeed, the applicability of theorems in mathematical logic to the rest of mathematics depends largely on the assumption that mathematics could, if necessary, be conducted in the language of firstorder predicate calculus. And, indeed, people familiar with logical notations – and with English – can become very adept at this sort of translation. What they cannot do, however, is to give a formal description of how they go about it. Quine (1959), after outlining some useful hints for his reader, says: '... in the main we must rely on our good sense of everyday idiom for a sympathetic understanding of the statement, and then re-think the whole in logical symbols'.

The problem of translating English into another representation, one in which we hope to be able to formalize the concept of logical consequence, is our main concern in what follows. We have been taking a 'syntactic' approach to this problem; our aim is to operate directly on the strings of words presented, rather than on our 'understanding' of them. While we do not yet have a fully-specified candidate for the representation into which we should translate, the need for one which can be related to English by formal rules has led us to structures very similar to those of the transformational grammarian.

SOME ENGLISH QUANTIFIERS

We begin by describing a method for translating a modest subset of English into a slightly modified first-order predicate calculus – modified just enough to provide a representation for questions. We can then go on to investigate the difficulties which arise when we attempt to treat more of the language;

ISARD AND LONGUET-HIGGINS

not the least of these being that the first-order predicate calculus is not adequate to express the whole range of meanings of English sentences.

The statements and questions which we shall consider now may be exemplified by examples (7) to (12):

Not everyone met John.	(7)
Someone didn't meet everybody.	(8)
No one told anybody anything.	(9)
Did anyone meet John?	(10)
Did anyone meet everybody?	(11)
Did anyone tell anybody anything?	(12)

More precisely, we consider

A. Declarative sentences

$D \rightarrow NP' (didn't) V(NP) (NP)$

containing a noun phrase, possibly followed by 'didn't', followed by a verb, possibly followed by one or two noun phrases.

B. Questions begining with the word 'Did'

$I \rightarrow \text{Did } NPV(NP)(NP)?$

followed by a noun phrase, then a verb, then possibly one or two noun phrases.

A noun phrase is either a proper name, or a word formed by combining one of the quantifiers 'some', 'any', 'every', and 'no' with one of the variables 'one', 'body', and 'thing':

		some		070
$NP \rightarrow \begin{cases} John \\ QX \end{cases}$	Q→	any every no	X→	body thing

Furthermore, the first noun phrase of a declarative may be 'not every X':

$$NP' \rightarrow \begin{cases} NP \\ Not every X \end{cases}$$

Let us emphasize that we are not claiming that every string of words produced by these rules is an English sentence (e.g., Anybody saw John). It is just that we are restricting our attention to those sentences which can be so produced.

We would like to have rules which transcribe such declarative sentences into predicate calculus formulae, such as

$\sim \forall x M x j$	(7')
$\exists x \sim \forall y M x y$	(8')
$\forall y \forall z \sim \exists x T x y z,$	(9')

where Mxy stands for 'x met y' and Txyz stands for 'x told y z'. We would also like, for reasons which will become apparent, rules for transcribing questions into modified formulae of the type:

∀x?Mxj					(10′)
$\forall x ? \forall y M x y$: · · ·			(11')
$\forall x \forall y \forall z ?Txyz.$			·		(12')

In every formula there appears a matrix consisting of a predicate symbol corresponding to the verb, followed by a string of variables and constants; these are understood to correspond to the 'variables' and proper nouns appearing in the sentence, and in the same order. The matrix will be preceded by a string of quantifiers and negations – and possibly a question mark; we have found that the transcription rules which appear below produce unique and acceptable orderings of these symbols from unambiguous sentences of the specified type.

If, as we hope to demonstrate, there is any unique transcription of 'some', it must surely be into an existential 3 rather than a universal quantifier; and 'every' must surely become a universal quantifier, ∀. 'No one did so-and-so' seems to be the direct contradiction of 'Someone did so-and-so', so that there is a prima facie case for transcribing 'no' as $\sim \exists$. The word 'any' is much trickier to deal with. In sentences of the type under consideration it can only appear after a 'not', a 'no', or a 'did'. 'Anyone met John', and 'anyone didn't meet John' are not English, although we can say 'John didn't meet anyone'. One could take the line that 'John didn't meet anyone' should be transcribed as $\sim \exists x M j x$ (like 'John met no-one'). This would be in accord with the approach of those linguists who view 'any' as a variant of 'some' which appears in 'negative contexts'. See, for instance, Klima (1964). It is possible to write a set of rules which transcribe 'any' as '∃' and which are as good for the purpose at hand as our rules which follow. However, the fact that 'any' behaves in other connections as a universal quantifier ('Anyone can do that') leads us to prefer the transcription $\forall x \sim M j x$ in which 'any' becomes a universal quantifier immediately preceding the '~'; Quine (1960, p. 139), among others, discusses the behaviour of 'any' viewed as a universal quantifier.

The following rules enable us to construct the string of quantifiers, negations (and possibly a question mark), which are to precede the matrix in the transcription of an English sentence of the specified type.

(a) We define a 'transcription function' T which takes occurrences of words into occurrences of logical symbols:

246

(b) If an occurrence of a quantifier Q precedes an occurrence of 'not' or 'no' then T(Q) must precede T(not) or T(no).

(c) T(any) must directly dominate T(not) or T(no) or T(did), for some occurrence of 'not', 'no' or 'did' which precedes the occurrence of 'any'.

(d) If an occurrence of 'not' or 'no' or 'did' precedes an occurrence of 'every', then T(not) or T(no) or T(did) precedes T(every).

(e) If an occurrence of 'not' or 'no' precedes an occurrence of 'some', then T(not) or T(no) must *not* directly dominate T(some).

(f) ? must precede any occurrence of \exists .

(g) The transcriptions of occurrences of 'not', or 'did' must appear in the same order as the occurrences themselves.

When we say that a symbol σ directly dominates a symbol τ we mean that σ precedes τ and either there are no intervening symbols (not counting variables) or that all the symbols in between are identical with each other, and with either σ or τ .

The above rules were in fact used to generate the formulae (7') to (12')from the sentences (7) to (12). Confining our attention to declarative sentences for the moment, we claim that any unambiguous sentence from our subset of English can be transcribed in exactly one way without violating (b) to (e), and that this transcription represents the meaning of the sentence. For an ambiguous sentence, (b) to (e) will allow two or more transcriptions, corresponding to the various interpretations of the sentence. Consider, for example, the sentence (13)

Not everyone saw somebody.

(13)

The \forall representing 'everyone' must follow the ~ representing 'not', by rule (d), but the only restriction on the placing of the \exists representing 'somebody' is that it should not immediately succeed the ~, which would then directly dominate it [rule (e)]. Two readings are therefore possible:

$\sim \forall x \exists y S x y$	(13')
$\exists y \sim \forall x S x y$	(13")

either of which is a possible interpretation. (Actually the former, in which the order of the quantifiers is the same in the formula as in the sentence, is the preferred reading, and we believe this to be true of ambiguous sentences in general.) Or consider (14):

Everyone didn't see somebody.

(14)

Here rule (b) dictates that $\forall x$ must precede ~ and by rule (e) $\exists y$ must not immediately follow ~. This leaves open the two interpretations (14') and (14")

$\exists y \forall x \sim Sxy$		(14')
$\forall x \exists y \sim S x y$		(14")

(of which the latter is perhaps preferable). But one could also interpret the sentence in either sense of (13), and this suggests that rule (b) might be

relaxed by adding 'unless \forall is "every" and is the first word of the sentence'. Whether one allows this exception or not is, of course, a matter of personal linguistic style.

There are a number of 'sentences' produced by our little grammar, including double-negative sentences and negative questions, which sound a bit strange, and which some people might reject as not belonging to English; for example, 'Everybody gave nobody something'. We do not wish to argue here over whether these 'really' are sentences. In as far as such sentences are amenable to interpretation, our rules yield their interpretations, with one exception, namely (15)

Not everyone met anybody, (15)

which is, so it seems, interpretable as either (15') or (15''), or both:

<i>∀y~∀xMxy</i>		(15')
$\sim \forall x \exists y M x y$		(15")

Our rules predict only the former reading. If we had written them to anscribe 'any' as \exists , we would have got only the latter. Together with a

transcribe 'any' as \exists , we would have got only the latter. Together with a number of other examples, this suggests that the proper treatment of 'any' might be to write it *either* as an existential quantifier inside the scope of a 'negative word' or as a universal quantifier outside the scope. (E.g., 'Few students solved any of the problems' can mean either that, given any problem, few students solved it, or that there were few students who did any problem-solving.)

Sentences involving more than one negation also fall into the 'marginal' category. An interesting case is

No one didn't see anybody (16)

which according to rules (b) and (c) may be written as either

$$\forall y \sim \exists x \sim Sxy \tag{16'}$$

or

$$\exists x \forall y \sim Sxy. \tag{16"}$$

But if more than one occurrence of 'not' or 'no' precedes an occurrence of any, the preferred interpretation seems to be that in which T(any) directly dominates the transcription of the 'not' or 'no' to which it is closest.

In general our rules interpret double-negative sentences in the 'proper' rather than the 'vulgar' fashion; for example, they interpret 'Nobody saw nothing' as a paraphrase of 'Everybody saw something' rather than 'Nobody saw anything'.

Our transcription of a question gives a predicate calculus formula with a '?' in its prenex. To every such formula there corresponds another, in which the '?' is replaced by a ' \sim ', and which can be obtained by transcription of a negative declarative sentence.

Consider, for example, the question (17):

Did John tell anyone everything?

(17)

which our rules transcribe into $\forall x ? \forall y \text{ (John told } x y)$ (17') Replacement of '?' by '~' in (17) gives (18') $\forall x \sim \forall y \text{ (John told } x y)$ (18')

which is the transcription of

John didn't tell anyone everything.

On the basis of (18) we would undoubtedly wish to reply 'no' to the question (17). So if a question differs from a (single-negative) declarative only by the presence of '?' rather than ' \sim ' in its transcription, we can answer 'no' to the question on the basis of the declarative. More generally, we shall obtain a 'no' answer if a declarative is available from which can be deduced the negative formula which matches the question. Thus in the above situation we might have been told (19)

John told no one everything (19)

i.e., $\sim \exists x \forall y \text{ (John told } x y)$

This implies (18'), so on the basis of (19) we could answer (17) in the negative.

To obtain a 'yes' answer to a question we may proceed by erasing the '?' and converting any universal quantifiers which precede it into existential quantifiers. The answer is 'yes' if the resulting declarative formula is available, or follows logically from other available formulae. Thus the question (17):

Did John tell anyone everything? (17)

i.e., $\forall x ? \forall y (John told x y)$ (17')

is answered 'yes' if the formula

 $\exists x \forall y \text{ (John told } x y) \tag{}$

is available, as it is if we have been told (20)

John told someone everything.

Given this way of interpreting questions, we wish to make the same claims about the way in which our rules operate on ambiguous questions as we made about the way they operate on ambiguous declaratives. There is one additional point, however, relating to questions containing negations. If we were to ask someone 'Did no one meet anybody?' and he were to answer simply 'yes' it would not be clear whether he meant, 'Yes, no one met anybody', or 'Yes, someone met somebody'. For this reason, we might not want to give 'yes' and 'no' answers to questions containing negations, but the procedure of replacing the '?' with a '~' or erasing it still indicates what information the questioner has asked for.

EXTENSIONS

Ì

Plainly, the only interesting thing that this translation procedure purports to do is to get the order of the prenex right. Keeping this in mind, we can see

(18)

(19')

(20')

(20)

that there will be a number of ways of enlarging the class of sentences treated without seriously altering our rules. First of all, we can introduce noun phrases consisting of 'the' followed by a noun, and treat these in the same way as we treat proper nouns. Slightly more ambitiously, we could let ordinary nouns appear in the places where 'variable' words appear, and then instead of writing ordinary quantifiers, we would write 'relativized quantifiers', in the sense of Tarski, Mostowski and Robinson (1953), $(\forall x^{\text{Man}})$, $(\exists y^{\text{Dog}})$. Formulae with relativized quantifiers could later be converted to ordinary formulae, if this seemed convenient, by the standard method of rewriting $(\forall x^{\text{Man}})\psi$ as $(\forall x)(\text{Man}(x)\rightarrow\psi)$ and $(\exists y^{\text{Dog}})\psi$ as $(\exists y)(\text{Dog}(y)\wedge\psi)$. This same trick can be used to represent sentences which have relative clauses attached to nouns, for example, 'Every man that I know is here', putting the whole relative clause inside the relativized quantifier. This will work as long as no quantifiers appear inside the relative clause itself. If quantifiers do appear inside a relative clause the situation may become more complicated. To begin with, we must decide whether the quantifiers in a relative clause should come into the prenex of the main formula or stay with the clause. Then, how do we account for 'Everyone who met anyone enjoyed the party', where there is no negation present, and why is there no sentence 'Someone who met anyone enjoyed the party'?

The failure of these rules to handle relative clause sentences actually points to a more general failure to cope with sentences having any subordinate clauses at all. But for simple sentences, those without subordinate clauses, the rules do get quantifier order right, even if the sentences are, say, passives, or involve a preposition, as in 'No one gave anything to everybody'.

GRAMMAR AND SCOPE

If we are to take advantage of this, and extend our rules so as to cover more than simple active sentences, we should introduce rules relating 'active' and 'passive' matrices in our formulae. The most sensible way to do this would appear to be to retain the actual words, rather than introduce constants and predicate symbols, and then use grammatical transformations. Of course, these transformations are not properly stated in terms just of strings of words, but rather in terms of strings with their grammatical structure indicated. (That is, if we view a transformation as an instruction to do something, it might be an instruction to move, say, a noun phrase to the end of the sentence, and in some situations this would amount to moving the word 'John', in other cases, the word 'Bill'. But a grammar would never have an explicit instruction 'Move the word "John" to the end of the sentence'. Thus we must not only keep the words of the sentence, but indicate their structure as well, for example, indicate that 'John' is a noun phrase.)

The structure of a sentence is normally expressed in 'tree' form, and we wish to adopt this form. (Figure 1 represents a conventional parsing tree.)



Figure 1

There are a number of ways in which the concept of quantifier scope might be incorporated into syntactic trees. We do not want to discuss the merits of particular ways here, but simply note that some device must be adopted which will indicate which quantifiers fall within the scopes of which others.

In fact, it will be necessary to indicate the scopes of some other words besides quantifiers. 'Not' certainly, and also some verbs. There is, for instance, the notorious sentence:

Mary wants to marry a Norwegian.

(21)

This is ambiguous, one reading being that Mary has a particular man in mind, and the other that she does not. We think that the best way of expressing the difference is to say that in the latter case, the existential quantifier falls within the scope of the verb 'want', while in the former case it does not. A proposal to this effect appears in Bach (1969). This represents an extension of predicate calculus, where quantifiers cannot appear within the arguments of predicate symbols, but notice that the interpretation on which there is no particular Norwegian does not have a natural predicate calculus representation.

It is, of course, difficult to assess the rightness or wrongness of such a notational device unless we can point out some consequences of adopting it. One consequence is that we have two ways of representing a sentence which has two different meanings, and this is desirable, but there are surely many other ways of achieving it.

Earlier in the paper we introduced a new bit of notation, the question mark in the modified predicate calculus, and justified it in two ways. We displayed rules linking it with English and we gave rules for manipulating it which produced appropriate 'yes' and 'no' answers. We are not prepared, at this point, to do either of these things for the present 'scope' notation, but we can produce some justification for it beyond saying that it 'feels right'. This is that the notation partitions existentially-quantified variables according to the verbs within whose scope their quantifiers fall, and English makes a similar partition which shows itself when we try to make further references to these variables, using pronouns such as 'he' or 'it'. Notice that the question

'How tall is he?' as a reply to (21) is only appropriate if we understand (21) as 'There is a Norwegian that Mary wants to marry'. Otherwise we have to say something like 'How tall does she want him to be?' In general, we will not be able to use 'he' or 'it' to refer to a variable whose quantifier falls within the scope of a verb without repeating the verb, or at least using a modal or subjunctive construction ('How tall should he be?') to indicate that the referent does not exist 'in reality' but only within some understood context such as 'What Mary wants'. Things which 'really exist' are those whose existential quantifiers do not fall within the scope of any other operator, and only they can be directly referred to as 'he' or 'it'. Further discussion along these lines appears in Karttunen (1970).

BANYANS

Supposing that we have a case where a pronoun in one sentence makes reference to an object introduced in a previous sentence, we must be able to express this in our notation. We have decided to do it by letting the trees representing the two sentences share the node representing the object. Thus,

John saw a flying saucer. Mary saw it, too. (22) would be represented by linked trees which (omitting details) would look something like (23).



But 'John saw a flying saucer; Mary saw one too.' would receive the different representation shown in (24).





(23)

(24)

ISARD AND LONGUET-HIGGINS

(25)

(26)

We have adopted the word 'banyan' for such sets of linked trees.

Using these banyans we can see a way of generalizing our questionanswering procedure to more than 'yes-no' questions. The procedure described for predicate calculus, when transferred to trees, amounts to trying to find in the information store a tree which corresponds exactly to the question except for the question mark itself. The answer is then determined by what on the information tree corresponds to the question mark on the question tree. Thus (25) represents a situation in which we get the answer 'no'.



Now we can do essentially the same thing to answer 'who-what-where'-type questions if we replace the 'question word' with a question mark. If we do this, as in (26), the information 'John saw Mary' will provide the answer 'Mary' to the question 'Who did John see?'



Of course, in the predicate calculus case, we did not actually have to find one of the two sentences corresponding to the question, but just infer one of these sentences from the given information. The same will be true here, if we introduce rules, corresponding to axioms of modal logic, which allow us to deduce trees from one another. In saying this we are not, however, claiming that any existing logical system will prove adequate for elucidating completely the semantics of natural language.

Acknowledgements

We are grateful to the Royal Society and the Medical Research Council for financial support in the course of this work, and S.D.I. would like to thank the National Science Foundation for the opportunity to attend the 1969 Advanced Research Seminar in Mathematical Linguistics at the University of Illinois.

We both express our warmest thanks to Professor Michael Geis for a great deal of helpful advice and constructive heckling.

REFERENCES

Bach, E. (1968) Nouns and noun phrases. Universals in Linguistic Theory, pp. 91-122. New York: Holt, Rinehart & Winston.

Karttunen, L. (1970) Discourse referents. Proc. Int. Conf. comput. Linguistics, Stockholm, 1969. (To be published).

Klima, E. S. (1964) Negation in English. The Structure of Language, pp. 246-323. Englewood Cliffs, N.J.: Prentice-Hall.

Quine, W.V. (1959) Methods of Logic. New York: Henry Holt.

Quine, W.V. (1960) Word and Object. Cambridge, Mass.: MIT Press.

Tarski, A., Mostowski, A. & Robinson, R.M. (1953) Undecidable Theories. Amsterdam: North Holland.