A HYBRID SYSTEM FOR QUANTIFIER SCOPING

ARNOLD J. CHIEN
PRC Inc.
1500 PRC Dr., 5S3; McLean, VA 22102 USA
chien_arnold@po.gis.prc.com

1. Introduction

A prominent source of ambiguity confronting natural language processing systems is ambiguity of quantifier scope relations. For example, the sentence *Some target was hit by every arrow* has one reading on which the quantified noun phrase (NP) *some target* has wider scope than the quantified NP *every arrow* (some particular target got hit by all the arrows), and another reading on which *every arrow* has wide scope (each arrow hit some target or other). Many factors influence preferred scope readings. Semantic factors, for example: in *Sam served one beer to all customers*, we prefer wide scope for *all* because the alternative reading entails the unlikely scenario of patrons huddled around a single beer mug. Syntactic factors: e.g. embedded prepositional objects often scope over heads, as in *Every teacher at some high school joined the union*, whereas heads usually assume scope over NPs contained in a relative clause, as in *Every teacher who is at some high school joined the union.* Lexical factors (i.e. the lexical identity of quantifiers): e.g. *each* tends toward wide scope and *a* toward narrow scope. Linear order is a factor - leftmost quantifiers tend to have wide scope - and there are others as well.

Given the relevance of different factors, a question arises: how can a system determine a scope reading based on the *combination* of factors present in any given sentence?

The standard approach has two parts: first, assign measures to the scoping influences of specific factors taken individually, and second, integrate the individual measures. The first task is performed by various "specialists". A system may have a lexical specialist which represents the wide scope tendency of *each*, a specialist which represents the inverse scoping tendency of an embedded prepositional object, a specialist which represents the tendency of quantifiers to scope according to linear order, and so on. The system will prefer those scope orders for which $f_{int}(f_{spec_1}, f_{spec_2}, ...)$ is optimal, where f_{int} is the integrating function and each f_{spec_i} is a specialist. For example, in the system of Grosz et. al. (1987), the specialists are called "critics." Given a candidate scope order, the "left-right" critic deducts points for each deviation from left-to-right scope order; the "quantifier strength" critic (i.e. lexical specialist) uses a numerical ranking of quantifiers to add and deduct points depending on how closely the candidate order respects the ranking; and so on. The integrating function f_{int} simply adds up the critics' points, though Grosz et. al. allow that the critics' judgments may need to be variously weighted in some fashion. To my knowledge all current systems use an "integration of specialists" (henceforth IS) approach, though not always as explicitly as Grosz et. al.; e.g. f_{int} often is implicit in the order in which various specialized rules or preferences are tested in the clauses of a complex conditional. See e.g. van Lehn (1978), Woods (1978), Allen (1987), Hurum (1988), Moran (1988). (Note that the common categorization of IS systems does not deny the myriad differences of detail between systems; indeed the functional characterization is useful because it abstracts over these differences.)

There is an alternative to IS. In what I will call "hybridization," different factors are conjoined before any scope judgment is made. A system hybridized for lexical and syntactic factors has no lexical or syntactic specialists, but rather a single function, call it $f_{lex\text{-}syn}$, whose input is the conjunction of lexical and syntactic factors in a sentence. Given an input with quantifiers *q1* and *q2* and (relevant) syntactic features $s_1, ..., s_n$, such a system computes $f_{lex\text{-}syn}(q1, q2, s_1, ..., s_n)$ rather than $f_{int}(f_{lex}(q1, q2), f_{syn_1}(s_1), ..., f_{syn_n}(s_n))$.

The advantage of this is that scope intuitions can be recovered directly. Take the tendency for an embedded prepositional object to scope over a head NP. This tendency varies depending on the quantifiers involved, among other things. In e.g. *Every man on some committee abstained*, there is a preference for the embedded NP to assume wide scope, but in *A man on many committees abstained*, the preference seems reversed. A prepositional phrase (PP) specialist in an IS system will not know how the preference changes when *a* and *many* quantify the head and the embedded object; since it is a specialist, it does not consider lexical input. Rather, the system must turn to the lexical specialist, which for its part knows e.g. that *a* usually takes narrow scope, but not how the behavior of *a* and *many* varies with specific environments, such as embedded PP constructions. It is hard to see, then, how any integration of these specialists could prescribe a scoping of *a* over *many* in an embedded PP context, since both prefer the reverse scoping. (An additional ordering specialist may prefer the correct scoping but without *ad hoc* weighting, the integrated preference will still be incorrect.) But there is no problem in a hybrid system, because the values $f_{lex\text{-}syn}$(*every, some*, head-embedded-PP) and $f_{lex\text{-}syn}$(*a, many*, head-embedded-PP) are completely independent, as opposed to having a PP specialist in common, and can be specified however intuitions dictate. Scope judgments are based on all the lexical and syntactic factors present, rather than on each factor taken in abstraction from the others.

My case for hybridization does not rely on counterexample, but on the fundamentally murky nature of IS. Consider an analogy. Suppose there is election data showing, for any pair of candidates and any state, the relative voting preference when the candidates ran in the state. How should we design a system to produce a preference given two candidates and a state? A natural approach would be to simply retrieve the datum based on the candidate and state input together. But on an IS approach, a "candidate" specialist would measure a tendency over all states of the relative performance of the two given candidates; a "state" specialist would measure a tendency over the relative performances of all candidates, taken pairwise, in the given state; then somehow the two measures would be integrated. The problem here is that *whereas the desired datum is a simple, the computation is based on complex abstractions over much data other than the desired, relevant bit.* That is the basic difficulty of an IS system, which the PP example was meant to illustrate.

Though semantic and pragmatic factors also influence scope, they are not central to my current concern: the design of a "base" scoping unit which can be ported to different domains and adaptively extended, and which can be improved incrementally as bits of real-world knowledge are gradually added to the system (as with Grosz et. al. 1987, Moran 1988, and Hurum 1988). Hence the focus on syntactic and lexical factors, which make up most of the domain-independent factors. I will return to this issue in section 3.

2. Implementation

A hybrid scoping system has been fully implemented as part of the PRC Adaptive Knowledge-Based Text Understanding System (Loatman et. al. 1986). Figure 1 shows the basic organization of the PAKTUS scoping module (PSM). I will describe input/output, the database, and the scoping algorithm in turn.

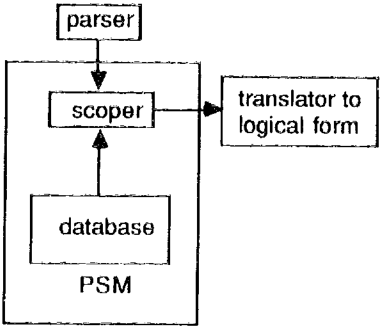

Figure 1. Organization of Scoping Module

2.1 Input/Output

Given a parse tree, PSM returns a list of the preferred scope orders of the quantified phrases. No degree of preference is computed. A scope order is represented by an ordered list of the phrases, not by a logical form.

Though eventually there will be translation to logical form, there is good reason for delaying this until after the scope determination. The problem with systems which translate a parse tree into an "unscoped" logical form as input to the scoping module (e.g. Hobbs and Shieber 1987) is that syntactic influences are not discernible to the module, since logical structure is not syntactic structure. For example, *Every teacher who is at some high school joined the union* and *Every teacher at some high school joined the union* have the same unscoped logical form: for Hobbs and Shieber, *joined-union(<every t and (teacher(t),at (t,<some h high-school(h)>))>)*. So the different syntactic influences are invisible. Though syntactic input can of course be added (e.g. Hurum 1988), doing so amounts to an admission that the translation was premature. It is more efficient to have the input to the module consist just of the parse, postponing the translation to logical form until after the scoping determination. Thus, the translator (not yet implemented) is not part of PSM.

2.2. Database

PSM encodes a function $f_{lex\text{-}syn}$ defined for 26 quantifier elements, including 9 quantificational adverbs such as *always*, and 49 syntactic environments. There are three "vertical" environments - embedded PP, reduced and full relatives - and 46 "horizontal" environments, where a horizontal environment is defined by a combination of grammatical roles, voice, and/or various ordering relations. Defining the mapping from a conjunction of quantifier pair and environment to a prescribed scope order for the over 9000 mathematically and syntactically possible conjunctions admittedly is a daunting task. This may be the main reason to prefer an IS approach. But while the required research effort has been lengthy and tedious, it has paid dividends in a body of data (150 pages, described in the appendix of Chien 1992), which subsumes existing consensus on lexical and syntactic scoping influences while going deeper and beyond. However, the corpus is naturally subject to continual correction and extension, and while this upgrading can be accommodated, the process is not modular. It seems to me that this is the tradeoff for the hybrid's greater precision.

Database implementation was motivated by the desire to make access to the large volume of data as efficient as possible. There are three levels of data objects. The first, top-level, object has slots corresponding to pairings of grammatical roles (subject, direct object, etc.; for their relevance to scope, see Ioup 1975). In each slot are pointers to several second-level objects, called "rule groups". In these, a "conditions" slot contains procedures which test for syntactic properties such as voice and linear ordering, and another slot contains pointers to third-level objects called "rules". In these, a conditions slot contains procedures to test for the lexical identity of a quantifier pair, and an "actions" slot contains procedures which effect a scope preference. Thus the latter procedures are invoked only after the collective syntactic and lexical properties of the input are verified. But checking the conditions in stages via the object hierarchy permits large aggregates of data to be eliminated from consideration at each stage. Data objects of all levels total about 325, including a second top-level object for vertical relations, cf. 2.3 below.

Database organization is illustrated in Figure 2. If a direct object and adverbial in a clause are quantified, the rule groups in the appropriate slot of RULEGRPS are tested. If in addition the clause is passive and the adverbial immediately precedes the main verb, then RULEGRP25 is activated and its rules tested. If, finally, the direct object is quantified by *some* and the adverbial is a "monotone decreasing" quantifier such as *never, seldom,* or *rarely* (Barwise and Cooper 1981) then RULE112 is activated and the procedure "setparams" invoked. The effect of this - in the context of the algorithm explained in the next section - is to register a preference for the object to scope over the adverbial, as e.g. in *He was seldom seen by some agent.* (The alternative scoping is awkward, better expressed with polarity-sensitive *any* replacing *some*; for the treatment of *any*, see Chien 1991.)

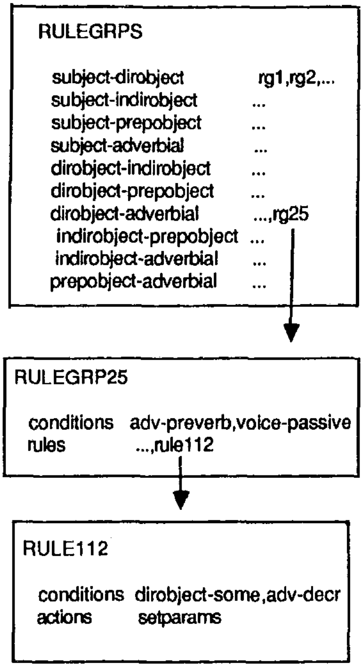
RULEGRPS
subject-dirobject rg1,rg2,...
subject-indirobject ...
subject-prepobject ...
subject-adverbial ...
dirobject-indirobject ...
dirobject-prepobject ...
dirobject-adverbial ...,rg25
indirobject-prepobject ...
indirobject-adverbial ...
prepobject-adverbial ...
RULEGRP25
conditions adv-preverb,voice-passive
rules ...,rule112
RULE112
conditions dirobject-some,adv-decr
actions setparams

Figure 2. Database Hierarchy

It should not be thought that a hybrid system cannot exploit generalizations in the data. PSM can and must do so, for even with a structured database, search would be relatively slow if there were as many actual data structures as abstract data points (i.e. values of $f_{lex\text{-}syn}$). But in fact each rule represents a cluster of like points, grouped together by quantifier categories - e.g. "decr" in RULE112, or the category of universal quantifiers - by boolean combination, or by other generalization, thus gaining economies in the database. To illustrate generalization by syntactic information alone, consider the verb objects in *He sent a firm each invoice*: they appear to scope in order regardless of how they are quantified. To capture this phenomenon, the relevant rule registers a preference without checking for the lexical identity of the quantifiers. Note that this strategy subsumes cases which in an IS system would be handled by an *overriding* specialist, i.e. a specialist f_o such that $f_{int}(f_o(x), ...) = f_o(x)$. In such cases IS is not problematic, but hybridization is equally straightforward.

A generalization can also be based on syntactic information together with partial lexical information, i.e. one quantifier only. It appears e.g. that *sometimes* in preverbal position always scopes over a direct object, as in *She sometimes polishes each trophy*, regardless of how the object is quantified. To implement this, the rule group that looks for this configuration of adverbial and direct object has in its rules slot a rule whose condition for firing is only that the adverbial is *sometimes*. Here is a generalization over the data points $f_{lex\text{-}syn}(sometimes, x, e)$, for all NP quantifiers x, where e is this syntactic configuration. Note that the organization of the database precludes an overriding determination based on lexical information alone, since syntax must always be checked first. But I am unaware of any lexical preferences which are exceptionless across syntactic environments.

The number of rules is further reduced by the use of a default preference: PSM initially assumes scope order to match linear ("natural") order. This enables the elimination of rules prescribing natural order, unless the preference is very strong in that it cannot be undone by any conflicting preference in a sentence with more than two quantifiers. This is explained below.

2.3. Scoping Algorithm

PSM determines the scope order only of quantifiers all of which are horizontally related, or all of which are vertically related (as in Epstein 1988). So, for *Every athlete who took some steroids won a race* the system scopes *every athlete* and *some steroids*, likewise *every athlete* and *a race*; then the scoping of *some steroids* and *a race* is treated as already indirectly determined.

The top-level scoping procedure calls the horizontal scoping procedure (H-SCOPE) for the top-level clause of the parsed input. It then substitutes, for each top-level NP in each of the resulting scope orders, an order returned by the vertical scoping procedure (V-SCOPE) for that NP. V-SCOPE simply returns its argument NP unless it has an embedded NP. The recognized vertical relations are embedded PP, relative clause, and reduced relative (or any combination). Van Lehn's "embedding hierarchy" (van Lehn 1978) - in which these relations induce inverse scope order, natural order, and ambiguity, respectively - is subsumed by the preferences in the database, which capture the variation of hierarchy preferences as quantifiers vary.

For sentences with two quantifiers, H-SCOPE basically just does a lookup. But for more than two, it is non-trivial to determine an overall order from a set of pairwise orders. H-SCOPE first assumes the default natural order and initializes a "record of imposed orders" (RIO). This is a list of quantifier pairs, registering the prescriptions which have been followed to date in a given order; it insures that they will not be later undone. RIO is initialized with strong natural orders, i.e. naturally ordered pairs which must stay that way. The main body of H-SCOPE is a loop through the applicable rule groups, then a loop through a group's rules. If a rule fires, it sets one quantifier to L(eft), the other to R(ight). How this prescription is realized depends on the overall order under consideration, and on RIO. If e.g. L does not already precede R, R may be postposed to L or L may be preposed to R, non-equivalent options if L and R are not contiguous in the order; an option is not pursued if it undoes a pairwise order in RIO. Resultant new overall orders either replace or supplement the original, the former if the rule prefers the inverse pairwise order to the natural, the latter if the preferences are equal. The results are then each operated on by the next applicable rule.

For *A person in each house on both streets saw several men who were robbing some banks*, PSM returns [*both each a several some*] in .7 seconds (Macintosh IIx Common Lisp 2.0, scoping time only). *Rarely did a park supervisor serving several districts in two counties assign everyone many trees with no large branches on some limb which might fall on a passerby* gets 4 scopings, all with *rarely* widest and *a passerby* narrowest, in 1.283 seconds.

3. Conclusions

As noted, semantic and pragmatic factors have deliberately been unaddressed. But a few words are in order on their eventual incorporation.

There are of a number of issues that always arise where semantic processing is concerned: compositionality, knowledge representation, etc. But what I want to address is an issue peculiar to the current system: namely, should semantic (read: semantic/pragmatic) factors be incorporated by hybridization or integration? That is, should $f_{lex\text{-}syn}$ be replaced by $f_{lex\text{-}syn\text{-}sem\text{-}prag}$, i.e. a function that considers all relevant factors before making any scope judgment? Or should $f_{lex\text{-}syn}$ be integrated with semantic specialists? There are problems with either alternative.

The problem with full hybridization is that the database would have to be remade *from scratch*, since the value $f_{lex\text{-}syn\text{-}sem\text{-}prag}(blah)$ is not a function of $f_{lex\text{-}syn}(blah)$. That is, $f_{lex\text{-}syn\text{-}sem\text{-}prag}(blah)$ is not the result of combining $f_{lex\text{-}syn}(blah)$ with other judgments based on *blah*: that would be a mixed IS/hybrid model, the second alternative. As noted in 2.2, new syntactic or lexical factors cannot be added to PSM in a controlled way. The same is true for any new factors. My goal in this paper has been to show that syntactic and lexical factors are well-behaved enough that non-modularity restricted to these factors is a burden which however is bearable, and worth bearing. But if all factors including infinite complex meanings are hybridized, the problems become intractable. It would be perhaps impossible to determine even a large portion of the function $f_{lex\text{-}syn\text{-}sem\text{-}prag}$. And even if it were only excruciating but not impossible, the effort would have to be largely duplicated whenever the data was extended. It's not for nothing that modularity is a hallmark of good design. (Note also, incidentally, that scoping would have to entirely follow translation, unlike Figure 1.)

As a working hypothesis I have adopted the second alternative. Yet the argument of section 1, extended to semantic factors, suggests that if the system is to capture the complex and subtle variations in human scope judgments, these factors should be not integrated but hybridized. To back away from this because it makes the engineering too hard may be understandable, but we should not forget the joke about the guy looking for lost keys where he knows they aren't because the light is better there. Modularity may be imperative for approaching complex problems, but there is no *a priori* reason why the mind must be modular. Indeed Fodor (1983) has speculated that much of it may not be, and hence he is pessimistic about cognitive science.

Obviously this is a deep issue, and I do not claim to have resolved it (for more, see Chien 1992). Nor am I saying either that in computational linguistics we should model human minds or that we should just design practical systems. I am suggesting that these goals ultimately may be incompatible - not because minds are too imprecise (e.g. Glymour 1987), but because they are too precise.

References

Allen, J. 1987 *Natural Language Understanding*. Benjamin-Cummings, Menlo Park, California.

Barwise, J. and R. Cooper. 1981 Generalized Quantifiers and Natural Language. *Linguistics and Philosophy* 4(2): 150-219.

Chien, A. 1991 How to Scope and Translate *Any*. *Georgetown Journal of Languages and Linguistics* 2(3-4): 223-233.

Chien, A. 1992 Modularity and Quantifier Scoping. Manuscript.

Epstein, S. 1988 Principle-Based Interpretation of Natural Language Quantifiers. *Proceedings of the Seventh National Conference on Artificial Intelligence*: 718-723.

Fodor, J. 1983 *The Modularity of Mind*. MIT Press, Cambridge, Massachusetts.

Glymour, C. 1987 Android Epistemology and the Frame Problem. In Pylyshyn, Z., ed., *The Robot's Dilemma*. Ablex, Norwood, New Jersey, 65-75.

Grosz, B., D. Appelt, P. Martin, and F. Pereira. 1987 TEAM: An Experiment in the Design of Transportable Natural-Language Interfaces. *Artificial Intelligence* 32(2): 173-243.

Hobbs, J., and S. Shieber. 1985 An Algorithm for Generating Quantifier Scopings. *Computational Linguistics* 13(1-2): 47-63.

Hurum, S. 1988 Handling Scope Ambiguities in English. In *Proceedings of the Second Conference on Applied Natural Language Processing*: 58-65.

Ioup, G. 1975 Some Universals Concerning Quantifier Scope. In J. Kimball, ed., *Syntax and Semantics, Vol. 4*. Academic Press, New York.

Loatman, B., J. Hermansen, S. Post, and C. Yang. 1986 PAKTUS Version 1 User's Guide. Report SD-RD-86-2, PRC Inc., McLean, Virginia.

Moran, D. 1988 Quantifier Scoping in the SRI Core Language Engine. In *Proceedings of the 26th Annual Meeting of the Association for Computational Linguistics*: 33-40.

Van Lehn, K. 1978 Determining the Scope of English Quantifiers. Report AI-TR-483, AI Lab, MIT, Cambridge, Massachusetts.

Woods, W. 1978 Semantics and Quantification in Natural Language Question Answering. In M. Yovits, ed., *Advances in Computers, Vol 17*. Academic Press, New York, 2-64.