On quantifier hierarchy and its paraphrase in a semantic representation of natural language sentences

Patrick Saint Dizier

CENTRE DE RENNES
IRISA

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P.105
78153 Le Chesnay Cedex
France
Tél.:(3) 954 90 20

Rapports de Recherche

N° 243

ON QUANTIFIER HIERARCHY AND ITS PARAPHRASE IN A SEMANTIC REPRESENTATION OF NATURAL LANGUAGE SENTENCES

Patrick SAINT-DIZIER

Octobre 1983

ABSTRACT

In this paper, we describe a rewrite system whose goal is to reorder the range of the quantifiers introduced by determiners that appear in natural language sentences so as to produce a correct formal logical representation of the meaning of these sentences. The range of the quantifiers is lowered or risen by these rules. If a sentence is ambiguous, then the rules allow to generate all the relevant configurations of quantifiers. Next, we explain how a paraphrase of the quantifiers and of their range is built so as to enable the user to say if the understanding (and eventually which one) is correct.

RESUME

Dans ce document, nous décrivons un sujet de réécriture dont le propos est de réordonner le rang des quantificateurs introduits par les déterminants qui apparaissent dans des phrases en langue naturelle de façon à produire une représentation formelle correcte de ces phrases. Le rang des quantificateurs est élevé ou abaissé par ces règles. Si une phrase est ambigüe, alors ces règles permettent de produire toutes les lectures possibles de cette phrase. Enfin, nous expliquons comment nous produisons une paraphrase du rang des quantificateurs de façon à ce que l'utilisateur puisse confirmer ou infirmer la(les) compréhension(s) du système.

1_ INTRODUCTION .

Our long term goal is to build an efficient and friendly interface between a man and a specialized computer, such as a text editor, an interactive query system, an electronic mail, ... We call this interface a user assistant. Its role is to be an expert of the functions of the application to which it is connected and to provide a friendly and helpful environment to people who wish to use the application. The user assistant is composed of a linguistic component, a user guidance module and an interpreter of the user's request. An overall description of this interface has been given in St_dizier 1983. In this paper, we will examine some aspects of the linguistic component.

The linguistic component of our interface is composed of a lexicon (application dependent), a set of morphological rules (application independant) that generate the different forms of verbs, nouns and adjectives and a set of rules that describe the different structures of french sentences and how to build a formal representation of what has been parsed. The result of the parsing process is a first order logical formula. In addition, this logical formula includes typed variables (as in Dahl 77) and a few modal operators have been added in order to represent the expression of time (Moore 81). This formal representation is quite similar to Sandewall 71, Colmerauer 77 and Dahl 77. For instance, the sentence:

"This car belongs to a doctor." is represented by:

this(x),a(y),car(x).doctor(y).to_belong_to(x,y)

The rules that describe the structures of french can be applied forward for analysis and backward for natural language generation.

In this paper, starting from Colmerauer's and Pique's approach, we first examine the problem of the hierarchy of the quantifiers introduced by determiners. We present a set of rewrite rules whose goal is to reorder the quantifiers in order to produce a correct formal representation of a natural language sentence. If the sentence is ambiguous, then all the relevant configurations are produced. Next, we show how to build a paraphrase of what the system has understood, and, in case of ambiguity, how we enable the user to choose the correct representation. The study of paraphrasing is, in this paper, limited to quantifier paraphrasing. Work is in progress to formalise and to build a context-independant paraphraser and will be published soon.

2_ THE BACKGROUND OF OUR WORK .

The starting point of our work are the hypotheses formulated by Colmerauer (79) and their revision by Pique (81), which we recall here:

Hypothesis 1 :

The quantifier introduced by the article of a verb subject governs quantifiers introduced by the complements closely linked to this verb if there are any. However, if the subject article is "un" (a),that of one of the complements is "chaque" "chacun des" (every, each) and there is a constraint of oneness on the verb argument quantified by the subject, the quantifiers introduced by the verb complements govern those introduced by the subject.

Hypothesis 2 :

In a construction including a noun complement, the quantifier introduced by the complement's article governs that of the common noun except if:

- The article of the complement is indefinite, and the article of the common noun is "tout", "chaque", "aucun" (every, each, all, no).

- The article of the complement is in the plural. It is then replaced by "quelque" (some).

- The article of the common noun is in the plural and the property attached to the complement is exclusive on the argument introduced by the common noun.

In these cases it governs the formula attached to the common noun.

Hypothesis 3 :

(a) When a verb has two complements, quantification is in the reverse order to that of appearance.

(b) When a noun has several complements, quantification is made by applying hypothesis 2 in the inverse apparition order.

These hypotheses enables us to represent formally a large number of natural language sentences. However, it turns out that in some cases these hypotheses fail as in:

(1) "One of the material that every engine is composed of is made of gold."

in this sentence, the quantification "every" introduced in the relative clause governs the quantification introduced by "one of the".

(2) "All the workers have not a lot of free hours."

The quantification introduced by "all" governs the other quantification of the sentence. The negation refers to the quantification introduced by "a lot of".

Futhermore, these hypotheses have the inconvenient to introduce a choice in the reordering of the quantifiers. This choice is correct in many cases but it fails in some others (See Pique 81). We think that many natural language sentences are ambiguous from the point of view of the range of the quantifiers, and that it is necessary to produce all the relevant interpretations. For example, if you say:

(3) "All the student have not solved a problem."

you may mean:

(3a) "All the students have not solved a problem, may be a different one for some students."

(3b) "No problem has been solved by any student."

(3c) "One problem, the same for all the students has not been solved."

In this paper, we propose a set of rewrite rules that can generate all the relevant interpretations of a natural language sentence from the quantifiers range point of view. The correct interpretation can be either selected by the user via a paraphrased produced by the interface or deduced from contextual information.

2_ A SET OF REWRITE RULES .

Our starting point is a left to right reading of an input sentence. A formal representation of the sentence is produced where the quantifiers introduced by the determiners appear in the determiners reading order. Then, a set of rewrite rules are applied in order to modify the order of the quantifiers so as to get a correct formal representation.

First, all the determiners and the negations that appear in a surface sentence are indexed according to their order of appearance. If in a sentence there are the determiners: the ... a ... all_the..., these determiners are indexed: the1 a2 all_the3.

Next, we substitute each determiner by an appropriate quantifier. Taking into account the specificities of our problem, interfacing computer applications in natural language, we distinguish four quantifiers: D, E, Q and U.

All the determiners belongs to one of these four sets:

D = {every, each}

E = {a, one, some, many, none, most, almost, few, who, whom, ...}

Q = {a lot of, a little, a few, ... }

U = {the (plural), all the, all , no, ... }

A range indicator then results in the substitution of a determiner by the quantifier it introduces. For instance, sentence (1) is composed of the following determiners:

one_of_the1 every2

the range indicator is: E1 D2

Finally, a set of rewrite rules is applied on the range indicator. The goal of these rules is to generate all the relevant configurations of quantifiers from the initial range indicator. The range of the quantifiers is lowered or risen by these rules. Let's note that in these rules there are no syntactic constraints.

Let n, m, p be three integer such as n < m < p, then we have the following rewrite rules:

R1 : Un Em ----> Em Un

R'1: Un Em ----> Um En

R2 : En Dm ----> Dm En

R3 : Qn Um ----> Um Qn

R4 : En Um ----> Um En

R5 : Dn Em ----> Em Dn if the property attached to the noun introduced by Dn is not exclusive on the noun introduced by Em.

and with the negation :

R6 : NOTn Xm ----> Xm NOTn

X = D, E or U.

R7 : Un NOTm ----> NOTm "somen"

R8 : Dn NOTm Ep ----> Dp En NOTm

These rules are applied from left to right on the range indicator until no more rule is applicable. When the negation is in the last position after the application of the rules, its scope is the verb to which it is linked. When the negation is not in the last position, it embraces in its scope all the quantifiers whose position is on its right.

4_ CONJUNCTIONS PROCESSING :

When there are conjunctions in a sentence, we introduce brackets in the range indicator. Between brackets are all the elements tied by a given a conjunction. The brackets are indexed as if they were a single quantifier and a local index is assigned to each quantifier included in the brackets. For example, the range indicator of:

(4) "All the teachers and some students have a micro computer."

is (U1 E2)1 E2 .

When the rewrite rules are applied, the quantifiers included between the brackets are then considered as a single quantifier determined as follows:

(X1 .... Xn)j is equivalent to Yj; the value of Yj is :

X if all the Xk are the same quantifier,

D if there exist Xk = D

U if there exist Xk = U,

E otherwise.

5_ EXAMPLES .

We will now examine some examples that will illustrate the application of the rules:

(5) "One worker collects all the magnetic tapes."

The range indicator is: E1 U2 rule R4 is applicable, the range indicator is rewritten into U2 E1, then, no more rule is applicable and the final representation is:

(5a) all the(x),one(y),magnetic_tape(x).worker(y).to_collect(y,x)

(6) "Every statue wears a ornament."

The range indicator is D1 E2 ; rule R5 is not applicable because "to wear" introduces a unicity constraint: an ornament can be worn by only one statue. This information can be stored, for instance, in the lexicon. The final representation is:

(6a) Every(x),an(y),statue(x).ornament(y).to_wear(x,y)

(7) "Peter invited all the lovers of a musical instrument."

The range indicator is : U1 E2 ; rule R1 and R'1 are applicable:

(7a) a(x),all_the(y),

musical_instrument(x).to_be_a_lover_of(y,x).to_invite(Peter,y)

(7b) all_the(y),a(x),

musical_instrument(x).to_be_a_lover_of(y,x).to_invite(Peter,y)

As Pique (81) says, we prefer (7b) because we think that the sentence intends to mean that Peter is planning some musical party, for which several musical instruments are needed.

(8) "A lot of people have invited all my neighbours and some friends."

The range indicator is: Q1 (U1 E2)2 , (U1 E2) is equivalent to U and then R3 is applicable:

all_my(x),some(y),a_lot_of(z),

neighbours(x).friends(y).people(z).to_invite(z,x).to_invite(z,y)

If we consider now:

(3) "All the students have not solved a problem."

The range indicator is: U1 NOT2 E3, the three interpretations are produced:

* application of R6 and R1 : E3 U1 NOT2 (cf. interpretation 3a)
* application of R7 : NOT2 some1 E3 (cf. interpretation 3b)
* application of R6 and R'1 : U1 E3 NOT2 (cf. interpretation 3c).

6_ QUESTIONS .

The rewrite rules enables us to take into account the fact that questions like:

(10) "Who is looking after each gate?"

call for multiple answers:

"Who" is an element of the class denoted by the quantifier E; the range indicator is E1 D2, rule R2 is applicable:

(9a) each(x), who(y), gate(x).man(y).to_look_after(y,x)

The variable introduced by "each" governs the variable introduced by the questionning item. (9a) can be paraphrased:

(9'a) For each x, find the y for which car(x).man(y).to_look_after(y,x) is true.

7_ PARAPHRASING .

In an interface that supports a natural language communication, a paraphraser can be used to ensure that the system has correctly understood the user. If the input sentence (a request, a query, an instruction, ...) is not parsed correctly, the error can be caught before a possible action is activated. Futhermore, the user is assured the answer he receives is the answer to the question asked. The idea of using a paraphraser is not new, other systems have used canned templates to form paraphrases (Waltz 78, Codd 78), or a transformational grammar (CO-OP system, Kathleen R. McKeown, 79). These systems are able to generate paraphrases of questions whose form differ in a meaningful way from that of the original question.

The paraphraser provides the only way of error-checking for a casual user. It is possible to have the intermediate results printed, in which, for instance, we can find the parser's output. But the casual user is not able to understand these results. That's why the paraphraser has to respond in natural language. A paraphrase must clarify the system's interpretation of an ambiguous sentence without introducing additional ambiguity. This sometimes results in an inelegant natural language expression.

In our system, the elements of the sentence are divided into two classes: the presuppositions and the assertions. The role of the presuppositions in the paraphrase is to indicate to the user which underlying knowledge he uses when he makes an assertion.

8_ THE FORMULATION OF THE PARAPHRASE .

The paraphraser has been designed to be context-independent and thus, a change in the interfaced application requires no changes in the paraphraser. The input of the paraphraser is the formal representation produced by the parser (St_dizier 83). At the present time, contextual information is very limited since no running contextual database is used.

The first part of the paraphrase is composed of the string:
"Assuming that there are ...."
followed by the expression in quasi-natural language of the typed variables the formal representation of the input sentence contains. It ends by : " .....in our current world."

The second part of the paraphrase is composed of the remaining of the formal representation, i.e.:

- the expression of the quantifiers and that of their range,
- the assertions, represented by the verb predicates,
- the additional information linked to the typed variables and to the assertions, represented by predicates of adjectives and adverbs.

This second part of the paraphrase begins by " you mean that ..." if it is an affirmative sentence and by "you want to know .." if it is a query. In this paper, we will only explain the incidence of quantifier paraphrasing on the whole paraphrase. We will first give two examples:

(10) "Every statue wears a nice ornament." is paraphrased:

"Assuming that there are statues and ornaments in our current world,

you mean that for each statue there is an ornament such as [(a) the statue wears the ornament and (b) the ornament is nice.]"

The second part of the paraphrase is multiple if the input sentence is ambiguous:

(3) "All the students have not solved a problem."

is paraphrased:

"Assuming that there are students and problems in our current world,

You mean that:

(a) There is a problem, for all the students such as [the students have not solved it],

(b) For all the students, there are no problem such as [a student has solved it],

(c) For each student, there is a problem such as [a student has not solved it]."

To formulate a paraphrase, each quantifier D, E, Q and U is replaced by an appropriate string of words. Roughly speaking, we can say that:

E is replaced by: "there is a", "there are some" (depending on the determiner they introduce), or by "the" if the sentence is a query,

D is replaced by: (1) "for each" if it is not in the scope of an E

(2) "for all" if it is in the scope of an E.

Q is replaced by : "There is a little, a lot of ....."

U is replaced by: (1) "for each" if it is not in the scope of an E
(2) "for all" if it is in the scope of an E.

If elements of E are in the scope of a negation, they are transformed by the equivalence NOT E <==> U NOT except for the last E that is paraphrased by "There is no ..."

We will now explain how the last section of the paraphrase, written here between square-brackets, is built. First, some transformations are done on the original formal representation of the sentence:

(1) The original determiners are replaced by "a" or "the", depending on how the original words they introduce are quantified and if the input sentence is an affirmative or a query.

(2) Next, the formal representation is divided into substructures:

(a) The main verb, its subject and object(s) (if there are some),

(b) The clauses linked to the subject, to the objects and the additional information linked to the verb (adverbs, ...). These clauses describe some particular aspects of the request or of the question. There are as many substructures as there exist different clauses. Sometimes, some transformations are applied to a substructure in order to get a complete natural language sentence (for instance: noun_predicate(x) adjective_predicate(x) ----> noun_predicate(x) adjective_predicate(y) to_be(x,y))

For instance (11) "Every high statue wears a nice ornament." is represented by:

Every(x),an(y),statue(x).high(x).ornament(y).nice(y).to_wear(x,y)

This representation is divided into three substructures:

(11a) The(x),the(y),statue(x).ornament(y).to_wear(x,y)

(11b) the(x),statue(x).high(y).to_be(x,y)

(11c) the(x),ornament(x).nice(y).to_be(x,y)

(3) Finally, the parsing rules are applied backward on each substructure, and natural language sentences are produced:

(11'a) The statue wears the ornament.

(11'b) The statue is high.

(11'c) The ornament is nice.

In order to get a more elegant paraphrase, some transformations can be applied to these sentences. For instance, when only one noun is quantified by "there is a ..." it can be replaced by "it".

In more complex sentences, additional transformations are needed, such as (for french): subject-verb inversion, suffix-hopping, "has" and wh-questionning item deletion. But these transformations are out of the subject of this paper. Paraphrases are also used in our system to enable the user to confirm the resolutions of anaphoras.

9_ CONCLUSION .

In this paper, we have described a set of rewrite rules whose goal is to reorder quantifiers introduced by the determiners of natural language sentences. We have shown that when a sentence is ambiguous, all the relevant configurations of the quantifiers are produced. Finally, we have described briefly how paraphrases are built in order to enable the user to say which interpretation is correct. This job has been implemented in PROLOG and the interface is connected to a text editor and to an application, called CIGARE, whose role is to help people in scheduling meetings.

However, additional basic research remains to be done in the field of time and events representation. A lot of job also remains to be done to formalise and to build an efficient context-dependent paraphraser.

REFERENCES

CODD E. F. (1978)

Rendez-vous version 1. IBM research report RJ 2144, San Jose Ca.

COLMERAUER A. (1977)

Un sous-ensemble interessant du francais.

GIA Marseille, rapport interne.

DAHL V. (1977)

Un systeme deductif d'interrogation de banques de donnees en espagnol.

These de 3eme cycle, Univ. d'Aix-Marseille II.

MCCORD M.C. (1982)

Using slots and modifiers in logic grammars for natural language.

Technical report no 69-80, University of Kentucky, Lexington USA.

McKEOWM K. R. (1979)

Paraphrasing using given and new information in a question-answer system.

The Moore School, Univ. of Pennsylvania.

MOORE R.C. (1981)

Problems in logical form.

SRI projects 1605 and 1894, note technique no 241.

PIQUE J.F. (1982)

On semantic representation of natural language sentences.

Logic programming conference, Marseille 1982.

PEREIRA F., WARREN D. (1981)

An efficient easily adaptable system for interpreting natural language queries.

Logic programming conference, Marseille.

SAINT-DIZIER P. (1983)

Knowledge representation in a man-machine interface.

Proc. of the "The mind and the Machine Conference 83", Middlesex Polytechnic, London.

SAINT-DIZIER P. (1983)

Modelling human-computer interactions in a man-machine interface.

Logic Programming Workshop 83, Albufeira, Portugal.

SAINT-DIZIER P. (1983)

Analyse et conduite du dialogue dans une interface homme-machine.

Journees BIGRE 83, Cap d'agde, Octobre 83.

SANDEWALL E.J. (1971)

Representing natural language information in predicate calculus.

Machine intelligence no 6, Edinburgh university press.

WALTZ (1978)

An English language question answering system for a large relational database.
CACM 21 no 7.

Liste des Publications Internes IRISA

PI 172 **Comparaison de groupes de variables : 2ème partie : un exemple d'application**
B. Escofier, J. Pajès , 37 pages ; Juillet 1982

PI 173 **Unfold-fold program transformations**
L. Kott , 29pages ; Juillet 1982

PI 174 **Remarques sur les langages de parenthèses**
J.M. Autebert, J. Beauquier, L. Boasson, G. Senizergues , 20 pages ; Juillet 1982

PI 175 **Langages de parenthèses, langages N.T.S. et homomorphismes inverses**
J.M. Autebert, L. Boasson, G. Senizergues , 26 pages ; Juillet 1982

PI 176 **Tris pour machines synchrones ou Baudet Stevenson revisited**
R. Rannou , 26 pages ; Juillet 1982

PI 177 **Un nouvel algorithme de classification hiérarchique des éléments constitutifs de tableau de contingence basé sur la corrélation**
B. Tallur , Juillet 1982 ;

PI 178 **Programmes d'analyse des re'sultats d'une classification automatique**
I.C. Lerman et collaborateurs , 79 pages ; Septembre 1982

PI 179 **Attitude à l'égard des mathématiques des élèves de sixième**
J.Degouys, R. Gras, M. Postic , 29 pages ; Septembre 1982

PI 180 **Traitements de textes et manipulations de documents : bibliographie analytique**
J. André , 20 pages ; Septembre 1982

PI 181 **Algorithme assurant l'insertion dynamique d'un processeur autour d'un réseau à diffusion et garantissant la cohérence d'un système de numérotation des paquets global et réparti**
Annick Le Coz, Hervé Le Goff, Michel Ollivier , 31 pages ; Octobre 1982

PI 182 **Interprétation non linéaire d'un coefficient d'association entre modalités d'une juxtaposition de tables de contingence**
Israël César Lerman , 34 pages ; Novembre 1982

PI 183 **L'IRISA vu à travers les stages effectués par ses étudiants de DEA (1è année de thèse)**
Daniel Herman , 41 pages ; Novembre 1982

PI 184 **Commande non linéaire robuste des robots manipulateurs**
Claude Samson , 52 pages ; Janvier 1983

PI 185 **Dialogue et représentation des informations dans un système de messagerie intelligent**
Philippe Besnard, René Quiniou, Patrice Quinton, Patrick Saint-Dizier, Jacques Siroux, Laurent Trilling , 45 pages ; Janvier 1983

PI 186 **Analyse classificatoire d'une correspondance multiple ; typologie et régression**
I.C. Lerman , 54 pages ; Janvier 1983

PI 187 **Estimation de mouvement dans une sequence d'images de télévision en vue d'un codage avec compensation de mouvement**
Claude Labit , 132 pages ; Janvier 1983

PI 188 **Conception et réalisation d'un logiciel de saisie et restitution de cartes élémentaires**
Eric Sécher , 45 pages ; Janvier 1983

PI 189 **Etude comparative d'algorithmes pour l'amélioration de dessins au trait sur surfaces point par point**
M.A. ROY , 96 pages ; Janvier 1983

PI 190 **Généralisation de l'analyse des correspondances à la comparaison de tableaux de fréquence**
Brigitte Escofier , 35 pages ; Mars 1983

PI 191 **Association entre variables qualitatives ordinales «nettes» ou «floues»**
Israël-César Lerman , 42 pages ; Mars 1983

PI 192 **Un processeur intégré pour la reconnaissance de la parole**
Patrice Frison , 80 pages ; Mars 1983

PI 193 **The Systematic Design of Systolic Arrays**
Patrice Quinton , 39 pages ; Avril 1983

PI 194 **Régime stationnaire pour une file M/H/1 avec impatience**
Raymond Marie et Jean Pellaumail , 8 pages ; Mars 1983

PI 195 **SIGNAL : un langage pour le traitement du signal**
Paul Le Guernic, Albert Benveniste, Thierry Gautier , 49 pages ; Mars 1983

PI 196 **Algorithmes systoliques : de la théorie à la pratique**
Françoise André, Patrice Frison, Patrice Quinton , 19 pages ; Mars 1983

PI 197 **HAVANE : un système de mise en relation automatique de petites annonces**
Patrick Bosc, Michèle Courant, Sophie Robin, Laurent Trilling , 79 pages ; Mai 1983

PI 198 **Une procédure de décision en logique non-monotone**
Philippe Besnard , 59 pages ; Mai 1983

PI 199 **A formal proof system for infinitary rational expressions**
Philippe Darondeau, Laurent Kott , 28 pages ; Mai 1983

PI 200 **Etude générale d'un réseau constitué de deux stations hyperexponentielles**
Jean-Yves Le Boudec , 12 pages ; Mai 1983

PI 201 **Langage de Dyck et groupe symétrique**
Yves Cochet , 13 pages ; Juin 1983

PI 202 **On the observational semantics of pair parallelism**
Philippe Darondeau, Laurent Kott , 14 pages ; Juin 1983

PI 203 **Les langages fonctionnels : caractéristiques, utilisation et mise en œuvre**
Daniel Le Métayer , 162 pages ; Juin 1983

PI 204 **On exact and approximate itterative methods for general queueing networks**
Raymond A. Marie, William J. Stewart , 27 pages ; Juin 1983

PI 205 **On quantifier hierarchy and its paraphrase in a semantic representation of natural language sentences**
Patrick Saint-Dizier , 17 pages ; Juillet 1983

Imprimé en France
par
l'Institut National de Recherche en Informatique et en Automatique