USING A TEXT MODEL FOR ANALYSIS AND GENERATION

E. FIMBEL, H. GROSCOT, J.M. LANCEL, N. SIMONIN

CAP SOGETI INNOVATION
129, rue de l'Université
75007 PARIS
FRANCE

ABSTRACT

The following paper concerns a general scheme for multilingual text generation, as opposed to just translation. Our system processes the text as a whole, from which it extracts a representation of the meaning of the text. From this representation, a new text is generated, using a text model and action rules.

This process is done in six steps : word analysis, sentence analysis using a Functional Grammar, reference solving and inference, construction of the text pattern, sentence generation, and word generation. Different kinds of information are used at each step of the process : text organization, syntax, semantic, etc.

All the knowledge, as well as the text, is given in a declarative manner. It is expressed in a single formalism named Functional Descriptions. It consists of lexical data, a Functional Grammar, a knowledge network, action rules for reference solving and sentence generation, models of text, rules of structuration, and sentence schema.

Text representation, included in the semantic network, is composed of different kinds of objects (not necessarily distinct) : text organization, syntactical information, objects introduced by the discourse, affirmations on these objects, and links between these affirmations.

1 TEXT VERSUS SENTENCES

Human translation is a complex process whose activities are not yet entirely understood. Generation of text differs from generation of single sentences : our work stresses the necessary **processing of the text as a whole** and **restructuring of this text.** A translator must on the one hand rearrange a whole sentence in order to respect certain stylistic rules and on the other hand modify the order of these sentences. Some of them may be deleted and others added.

In any case, in order to be able to generate a coherent text by means of the information extracted from a source text, it seems most important to us to understand what the text is refering to.

We have restricted our scope to economic geography texts, taken from a French review named ATLASECO. These texts deal with industry and agriculture, involving related concepts (growth, decline, production, economical balance...). Our system parses a french text, integrates its informative content into a knowledge base, then generates a new text from information extracted from this base.

We present in part (2) the text representation used throughout the process. We describe in part (3) the knowledge representation, based on the **functional descriptions.** In part (4), we briefly describe the different steps of the process.

2 TEXT REPRESENTATION

A text conveys information by various means. For example, the order of sentences, as well as their syntactic structure, may be significant.

The general text representation we use is the same for understanding and structuring. This representation includes the different kinds of information needed for these two processes.

The text is represented by a set of interrelated objects (sentences, words, semantic objects...). There are five classes of objects, not necessarily distinct : the visible objects, the syntactic components, the discursive objects (defined later), the affirmations and the links.

2.1 Visible Objects

Visible objects are chapters, paragraphs, sentences, words and word sequences . They are related by positional links, describing the organization of the text, as shown by the following figure :

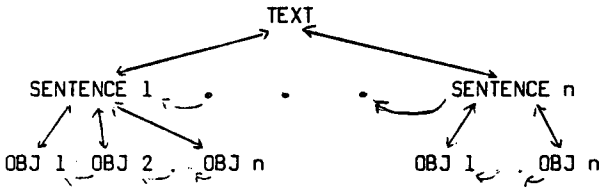

fig. 1

These links reflect the hierarchical organization of the text (chapters ...) and the dependent relationship of statements (order of sentences ...).

2.2 Syntactic Components

The sentences are represented by means of unordered "syntactic cases" (subject, determiner...), linking a component with its subcomponents. For example :

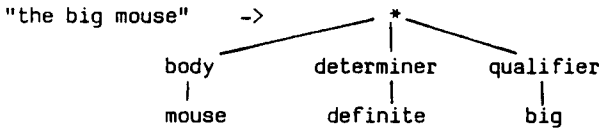

fig. 2

2.3 Discursive Objects

Discursive objects represent the semantic objects or entities introduced by the text. For example :

"The main crop is wheat, **whose production has strongly decreased. This decline** has strongly affected the commercial balance."

These two constructions correspond to a single discursive object, which could be called : "decrease of the production of wheat" .

The **meaning** of a sentence is represented by a network of discursive objects. The propositions, representing the discursive objects, are normalized into an **objective form.** For example : 'Peter met John" -> "meeting of John by Peter" "Mary is ill" -> "illness of Mary".

This normalization is very useful for co-reference solving.

The text itself may be a discursive object. Hence, it is represented by an object which mentions topic, date and positional links between sentences of text. During the Understanding Process, this object will be enlarged by new information, or modified (for instance topic may become more explicit).

2.4 Affirmations

Implicit within the text, as well as simply introducing objects, is the **evaluation** of certain objects. "The production is increasing" introduces "the production" and **affirms** "the increase of production".

There are various statement values ("stated positively in discourse", "dubious"...). These values may be given by the author, a figure, an economical organization, etc.

2.5 Links

The links are mainly operators (and, or, because...) that correlate the affirmations, which are discursive objects or links : "A **because** B" ; "A **and** B, **but** C".

They reflect **internal structure** of discourse. For example : "the production increases **but** the deficit remains important".

During Understanding phase, their main role is to transmit statement values to the objects they link.

The structuring stage isolates argumentative effects of the discourse, and builds new links (possibly the same). The choice of these links is important because they have to be coherently acceptable from the reader point of view.

3 KNOWLEDGE REPRESENTATION

We use **the same representation for all types of knowledge in the system** : lexical definitions, grammar rules, semantic rules,etc. They are given separately in the formalism of **functional descriptions** and integrated into the knowledge base that is represented by a **functional descriptions network.**

The language of **functional descriptions** comes from the functional grammars (DIK 78), which is a linguistic formalism that can be related to case grammars (FILLMORE 68). Historically, Kay (81) used **functional descriptions** as a general tool to represent grammars, independently of any specific linguistic theory, after which Rousselot (84) then used **functional descriptions** to represent any kind of knowledge (grammar, semantic rules, scripts..) in a system of story understanding.

We will use "DF" instead of "functional description".

A **DF** may represent any kind of knowledge. For example , "William cut himself" may be described by the following figure :

```
[action  = cut
 tense   = past
 actor   = [name  =  William]
 obj_act = <1 actor>]
```

fig. 3

A DF is an unordered set of identifier-object pairs. The identifiers are not pre-defined : they may be added or removed in any way. This allows us to put various kinds of information in the same description : syntactic, semantic, etc. The objects may be other DFs or **paths.**

A path is a notation whose aim is to point to objects that are already defined . We have two kinds of paths :

- A local path <1 list-of-identifier> points to an object in the current DF, as defined by the list of identifiers. For example, in fig. 3, the path <1 actor name> refers to "William".
- A global path, <g starting-point list-of-identifiers> points to an object in the knowledge base. It is defined in the same way as the local path, except that the starting point is the current DF.

The internal representation of a DF is a labelled graph. For example, the representation of the DF in fig. 3 is shown in fig. 4.

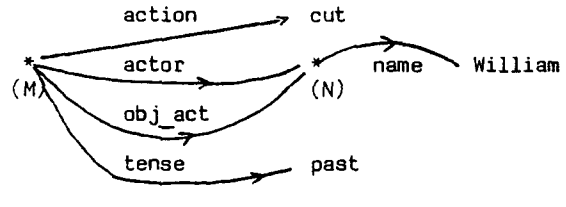

fig. 4

The system takes the DF, as shown in fig. 3, as input and converts it directly into the labelled graph as shown in fig. 4, which is then integrated into the knowledge base.

A **functional description network** is the graph that represents a **set of interlaced DFs,** i.e. the knowledge base. This network is a labelled graph whose nodes correspond to DFs, and whose links to elementary object properties. These links are labelled with identifiers.

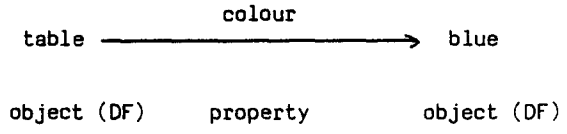

fig. 5

Each identifier (e.g. **colour**) may be given different properties (e.g. inheritance properties). This is done through attaching a node to the identifier.

The graph may contain **Dynamic Paths** , which allow **access** to the objects. A dynamic path is a function <f starting-point list-of-identifiers> whose value is a node on the graph. For example, in fig. 4, the dynamic path <f (M) actor name> points to the node "William". Dynamic Paths are very similar to substituable variables in formal systems.

The **processing** of a DF network can be done at two levels :

- **elementary level** : insert or delete links (i.e. properties) or nodes (i.e. objects) ;
- **form level** : each node of the network can still be viewed as a description, which corresponds to a complex set of links. This allows us to work on the **form** of the objects (pattern matching and merging).

The main algorithms that operate at the form level are those of **compatibility** (pattern matching) and **merging.** "Compatilibity" is a boolean function that decides whether two descriptions may correspond to the same object of the real world. For example, could the events described by "William speaks to himself" also be described by "Somebody speaks to Ted" ? :

```
[action  = speak
 actor   = [name = William]
 obj-act = <1 actor>]

[action  =  speak
 actor   = [ is-a  = human]
 obj-act = [ name  = Ted]]
```

fig. 6

These two descriptions are not compatible, because the name of the "obj-act" is "William" on one hand and "Ted" on the other.

"William cut himself "

```
[action  = cut
 tense   = past
 actor   = [name   =  William]
 obj-act = <1 actor>]
```

"X cut Y"

```
[action  = cut
 tense   = past
 actor   = [is-a  = human]
 obj-act = [state = changed]]
```

```
[action  = cut
 tense   = past
 actor   = [ is-a = human
             name = William
             state = changed]
 obj-act = <1 actor>]
```

fig. 7

Two compatible descriptions can be **merged.** The result is a new description, more complete than either of the originals. For example, the first two DFs of fig. 7 are compatible, and give the third one as the result of merging.

These two algorithms, inspired by the functional unification introduced by Kay, are very powerful and are used at each step of text processing. Their precise definitions use mathematical transformations of labelled graphs and are given in (GROSCOT, ROUSSELOT 85).

4 TEXT PROCESSING

In this section, we describe the following different steps of the text processing :
- word analysis
- sentence analysis
- reference solving
- construction of the text pattern
- sentence generation
- word generation

4.1 Word Analysis

The word analyser uses a knowledge base about the standard inflexions of the initial language and a dictionary of words and sequences of words. Each item in the dictionary contains a semantic definition, its syntactical category, and its type of inflexion with its roots.

We generate for each word a description of its syntactic features. All solutions are generated. For example, the analysis of the word "burns" gives at least these two DFs :

```
1) [word     = burn
    number   = plural
    category = substantive
       ... ]

2) [word     = burn
    category = verb
    tense    = present
    person   = 3
    number   = singular
       ... ]
```

fig. 8

4.2 Sentence Analysis

The analyser is based on the work of Rousselot (84). It works sentence by sentence and assumes three important functions :
- recognition and construction of syntactic components of a sentence
- control of semantic constraints
- generation of the semantic representation of the sentence, i.e. an affirmation or a link

The analyser uses a declarative grammar, which does not depend upon the way in which the analyser works. The grammar is a DF, in which each identifier-object pair represents a grammar rule. These rules allow the analyser to split a syntactical category into constituants, to verify the constraints, and to build the associated semantic representation.

The analyser works in a top-down manner by means of an **agenda** which allows the separation of the controls. Also included in the analyser is a **graph** that contains the partial analysis which minimizes processing time during the backtracks.

The starting point of this analyser is the set of DFs which have been obtained and graphed after word analysis.

We show here, in an example, the result of the analysis of the sentence "Agriculture is a success" :

```
[object             = success
 evaluated-object   = [object     = agriculture
                       def-undef  = definite]
 statement-value    = true]
```

fig. 9

4.3 Reference Solving

This stage determines what the pronouns and noun phrases point to : known objects (already in the knowledge base), or new objects introduced (directly or indirectly) by the text.

The process uses **action rules**, written by means of DFs. The rules interpreter works in a "lazy" saturation mode : It locally saturates the knowledge base regardless of whether or not all inferences have been made.

The process has three main features :

- it is directed by the syntax (definite articles, pronouns..)
- it identifies the objects at the semantic level, by testing the compatibility of discursive objects
- it uses positional links to define the possible references in each case. For example, a demonstrative pronoun has to be found among the preceding objects of the text

It has two important effects :

- it constructs a coherent network of discursive objects (the "meaning" of the text)
- it integrates the text into the knowledge base : each time that the text refers to a known object, links are created from this object to the text

4.4 Construction of the text pattern

Once the initial text has been analysed, its informative content is integrated in the knowledge base, thereby making it possible to question this base and to reformulate the so-obtained information with the aim of generating a new text.

This stage builds the text structure (paragraphs, sub-paragraphs, sentences...) using two complementary approaches : the **content** of information and the evolution of the **visible structure** of the text. Indeed, text generation is a reflective process : generation of a single sentence of the text must take account the preceding and following text. All the processes used for structuring are written by means of rules.

The starting point of structuring is a set of information extracted from the knowledge network.

First, all the information is inserted into a text model by means of their topics. Some obvious redundancies are eliminated : only one occurrence will be kept, in the most appropriate place.

When this is done, the position of the information, may be inadequate for a coherent text. Therefore, the text structure is changed : some information is enhanced ; the paragraphs are balanced ; the information is reordered according to chronological order and significance at the deepest level of the text structure.

At this step, output text structure is clearly apparent. But it is not sufficient because the text has now become a sequence of unrelated statements. Links are then created between the discursive objects introduced in the text (conjunctions, pronouns...) and ellipses are used to avoid repetitions.

4.5 Sentence Generation

From each DF of the text describing output information, this stage generates the description of a suitable sentence.

Syntactic patterns are associated with concepts (evolutions, appreciations, numbers...). Action rules combine these patterns to create the description of a sentence, i.e. a list of word descriptions.

4.6 Word Generation

Word generation uses the same organization of the knowledge base (about standard inflexions) as word analysis.

From a list of word descriptions, including syntactic features, the system constructs the output word. For example :

```
[word   = burn
 cat    = verb
 tense  = present
 person = 3              -->    B U R N S
 number = singular]
```

fig. 10

5 ANALYSING ECONOMIC GEOGRAPHY TEXTS

The texts we have analysed, taken from the French review ATLASECO, describe the main features of the agriculture in different countries. They contain an average of 20 sentences, which contain from 10 to 45 words.

The system used :
- 400 lexical items, which represent 2400 words
- 70 grammar rules
- a semantic network containing 220 concepts, 80 of them being domain-dependant
- 80 semantic rules for inference and reference solving
- 1 text model, containing 50 paragraphs and subparagraphs
- 100 rules for text structuring
- 30 sentence patterns for generation

Our system was able to extract the significant information that such a text conveys, and produce a new text from this information. As a matter of convenience, we used French as the target language for validation, however the process described here should also be able to work in a manner independant of the target language, by chaining certain parts of the knowledge base.

Now, we are in the process of adapting our system to generate French appliance operation manuals from the corresponding ones.

REFERENCES

ATLASECO
Atlas économique mondial
Le Nouvel Observateur, 1982.

DIK Simon
"Functional Grammar"
Publications in Language Science.
Foris Publications, Dordrecht Holland, 1978.

FILLMORE Charles
"The case for case" in "Universals in Linguistics Theory", E. Bach and R. Harms (eds), p.1-90, 1968.

FIMBEL Eric
"Les reseaux miroirs : un mécanisme d'inférence général ; application à un système d'assimilation de textes"
Thèse de l'université Paris 6, mars 1985.

GRIZE Jean-Blaise
"Introduction à la logique naturelle et approche logique du dialogue"
Approches formelles de la sémantique naturelle.
Publication CNRS-UPS-UTM-ADI, 1982.

GROSCOT Herbert, ROUSSELOT François
"Un langage déclaratif uniforme et un analyseur syntaxico-sémantique"
Proceedings of Cognitiva, Paris, juin 1985.

GROSS Maurice
"Méthodes en syntaxe"
Hermann, 1975.

KAY Martin
"Unification Grammars"
Xerox internal publication, 1981.

McKEOWN Kathleen R.
"Generating natural language text in response to questions about database structure"
PH.D of the university of Pennsylvania, 1982.

ROUSSELOT François
"Réalisation d'un programme comprenant des textes en utilisant un formalisme déclaratif pour représenter toutes les connaissances"
Thèse d' état, Paris 6, 1984.

SIMONIN Nathalie
"Utilisation d'une expertise pour engendrer des textes structurés en français"
Thèse de l'université Paris 6, mars 1985.