MAURICE QUÉZEL-AMBRUNAZ - PIERRE GUILLAUME

# ANALYSE AUTOMATIQUE DE TEXTES PAR UN SYSTÈME D'ÉTATS FINIS

## 1. PRÉSENTATION

Toute procédure de reconnaissance ou de génération automatique de langues naturelles, c'est-à-dire toute procédure qui fait passer d'un niveau de langue à un autre niveau, repose sur la notion de modèle.

Le modèle lui-même comporte deux facettes, l'une est le type logique caractérisé par une classe de languages reconnus et traduits, l'autre, est le contenu concret des données linguistiques destinées à la représentation d'une langue particulière. Le type logique du modèle valable pour plusieurs langues est concrétisé par un système informatique contenant le métalangage d'écriture de la grammaire et des diverses données et contenant aussi l'algorithme d'exploitation de ces données.

Le système A.T.E.F., décrit en détail par ailleurs, est un support informatique permettant la reconnaissance et la transduction des langages d'états-finis. Il ne constitue évidemment qu'un maillon dans une chaîne de modèles texte-signification ou signification-texte (en traduction automatique par exemple) et d'autres modèles plus puissants sont nécessaires. Cependant, l'analyse de langues naturelles qui n'exige pas une puissance supérieure est déjà appréciable. En effet, elle comprend l'analyse morphologique et une partie de l'analyse syntaxique. Pour réaliser ces tâches, un système d'états-finis est évidemment beaucoup plus efficace qu'un système trop puissant. Ainsi le système A.T.E.F. est principalement utilisé pour l'identification des mots, l'analyse morphologique et le début d'une analyse syntaxique de langue naturelle.

Un tel texte se présente sous la forme d'une suite de caractères où le « blanc » joue le rôle particulier de séparateur de formes. Ces formes constituent à leur tour les éléments d'entrée du modèle. L'étape de morphologie doit permettre leur substitution par des quantités significatives pour le modèle suivant.

Une première stratégie possible consisterait en une simple consultation d'un dictionnaire de formes. Une réalisation basée sur ce principe se heurte rapidement à des contraintes dues au volume du dictionnaire actif. Une autre stratégie inclut l'utilisation de dictionnaires de racines et d'affixes. Chaque élément de ces dictionnaires pourra constituer un segment dans la forme origine.

Le rôle de la morphologie est alors de trouver à l'intérieur d'une forme, en fonction de règles de cohérences entre segments, les décompositions acceptables. Ces règles étant de type états-finis, on doit disposer d'un automate régulier en vue de la reconnaissance de telles unités. Le modèle associé au système A.T.E.F. comporte un tel automate, qui est un transducteur d'états finis. Lors de la définition d'un tel modèle on peut choisir entre deux types de réalisations: les règles de transition de cet automate peuvent faire partie intégrante de la description en machine de l'algorithme. Cette solution, favorable à l'efficacité globale du modèle, ne permet pas d'effectuer aisément des modifications portant sur les règles. De façon à préserver l'aspect polyvalent du système et à garder une souplesse d'emploi indispensable à des applications variées, on a préféré dissocier le fonctionnement de l'algorithme des règles. Ce second type de réalisation adopté pour A.T.E.F. implique un automate devant pouvoir accepter toutes les règles de cohérence constituant une grammaire valide. L'utilisateur a un contrôle aussi complet que possible du fonctionnement d'un tel système.

Le modèle est prévu pour permettre le traitement en parallèle d'un second niveau d'analyse. Au cours de la segmentation d'une forme on a accès aux résultats de la décomposition des 4 formes la précédant, et à la forme suivante. Ceci peut permettre d'orienter la segmentation de la forme en cours, et de restreindre les combinaisons entre les solutions associées à ces formes.

On fait ainsi intervenir des propriétés d'accords syntaxiques d'états finis. Les données externes accessibles à l'utilisateur sont constituées par:

— les déclarations de variables
— les formats
— la grammaire
— les dictionnaires.

## 2. LES DONNÉES

### 2.1. *Les déclarations de variables.*

Par leur intermédiaire, l'utilisateur définit les noms de variables et les valeurs associées.

*Les variables.*

Il appartient à l'utilisateur de se définir toutes les quantités qu'il juge nécessaires à la segmentation et à son interprétation.

Ces quantités sont représentées par des ensembles de valeurs. Ces valeurs se regroupent en classes disjointes auxquelles on associe un nom (nom de variable). Si une variable ne peut prendre qu'une valeur unique parmi celles qui sont déclarées, elle est dite exclusive. Sinon, elle est dite non exclusive et la valeur effective est prise dans l'ensemble des parties des valeurs déclarées.

Une répartition en variables morphologiques et syntaxiques est déterminée par leur utilisation. Celles qui sont introduites à partir du lexique sont de nature morphologique. Celles qui permettent d'interpréter la segmentation sont de nature syntaxique.

Dans les formats elles déterminent une valeur d'état associée à chacun de ceux-ci. Dans la grammaire, elles sont référençables en vue de tests de correspondances de valeurs et de leurs modifications. Le contrôle des dictionnaires se fait par utilisation d'une variable spécialisée (variable non exclusive DICT). La valeur de cette variable correspond aux différents dictionnaires validés à un moment de la segmentation.

Une variable prédéclarée (variable exclusive UL) désigne l'entrée du lexique représentative de la racine de la forme.

*Variables morphologiques.*

-EXC-

SEG : = (D, S, B, P).

-NEX-

PREF : = (IN, IM, IL, IR).
SDNA : = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11).
SDN : = (MENT, TE1, IT, E1, ETE1, EUR, ESSE, ABLE, IBLE, ATION, ITION).
PDRP : = (AM, AN, VA, VAM, VN).
PDRN : = (N, NAM, NAN, NVA, NVAM).
DICT : = (1, 2, 3).

*Variables syntaxiques.*

```
                                                    -EXC-
CAT   : = (VRB, NMC, NMP, PRP, AJQ, AJP, AJD, ART, ADV,
           PRE, CNJ).
DRV   : = (AJAV, AJNM, VBAJ, VBNM, VBAJAV).
NEG   : = (NG).
                                                    -NEX-
GNR   : = (MAS, FEM).
NBR   : = (SIN, PLU).
```

## 2.2. *Les formats.*

A chaque élément des dictionnaires on doit associer un certain nombre de valeurs qui caractérisent son comportement vis-à-vis du modèle. Ces valeurs sont utilisables d'une part pour déterminer les accords possibles entre segments, d'autre part, pour représenter les caractéristiques syntaxiques associées à ces segments. Les classes ainsi introduites dans le lexique s'avèrent être en nombre réduit par rapport aux éléments de celui-ci. Elles sont représentables par un couple de formats: morphologique et syntaxique. Les formats morphologiques sont utilisés comme arguments pour la recherche des règles à appliquer en vue de l'accord des segments. Les valeurs qui leur sont associées sont prises parmi les valeurs des variables de type morphologiques. Les formats syntaxiques permettront de compléter les valeurs résultant du découpage de la forme.

*Formats syntaxiques.*

```
PS      01 == .** NEG -E- NG.
SA534   01 == .** CAT -E- AJQ, PDRP -E- AM -U- AN, PDRN -E-
SA534   02 N -U- NAM -U- NAN, SDN -E- ITE1, PREF -E- IR.
SS2     01 == .** DRV -E- AJNM, CAT -E- NMC, GNR -E- FEM,
        SDNA -E- 2.
VS3     01 == .** GNR -E- FEM, NBR -E- SIN.
```

*Formats morphologiques.*

```
BA4     01 == .** SEG -E- B, SDNA -E- 4.
D3      01 == .** SEG -E- D, SDNA -E- 3 -U- 4 -U- 11.
PM4     01 == .** SEG -E- P, PREF -E- IR.
SM3     01 == .** SEG -E- S, SDN -E- ITE1.
```

### 2.3. *La grammaire.*

Elle est formée par un ensemble de règles décrivant la fonction de transition de l'automate. Chacune de ces règles est potentiellement applicable à tout instant. Chaque segment extrait d'une forme conduit, par l'intermédiaire des dictionnaires, à un ou plusieurs formats morphologiques.

Un format morphologique introduit un jeu de valeurs de variables référençables dans l'automate comme valeur argument (A). Les valeurs de variables provenant du début de segmentation de la forme constituent l'état courant (C). La partie affectation de règle permettra de faire évoluer cet état courant. L'application d'une règle est subordonnée à la réalisation de plusieurs conditions. Chacun des formats morphologiques associé au segment permet de sélecter dans la grammaire un premier sous-ensemble de règles. Celui-ci est constitué par les règles où le nom de format figure en partie gauche. Les conditions s'expriment comme des relations entre variables des états argument et courant: on guide de cette façon la segmentation (cohérence morphologique).

L'introduction des états résultants de la segmentation des formes précédentes (désignées par P1, P2, P3, P4) permet d'opérer sur un contexte élargi (cohérence syntaxique). Les chaînages entre découpages évoluent en conséquence. On peut également faire intervenir les valeurs associées à la forme suivante (S). Ce type de condition sera pris en charge lors de l'analyse de cette forme.

*Exemple de condition d'application de règle.*

```
SEG(C) -E- S -ET-
(PDRP(A) -INC- VA -ET- DRV(C) -E- VBAJ
-OU- PDRP(A) -INC- VAM -ET- DRV(C) -E- VBAJAV
-OU- PDRP(A) -INC- VN -ET- DRV(C) -E- VBNM
-OU- PDRN(A) -INC- NVA -ET- DRV(C) -E- VBAJ -ET- SCHAINE
(A, O, 1) -NE- ' '
-OU- PDRN(A) -INC- NVAM -ET- DRV(C) -E- VBAJAV -ET-
SCHAINE (A, O, 1) -NE- ' ')
```

Les affectations se font à partir des valeurs arguments, courantes et éventuellement précédentes, vers les valeurs des états courants et suivants. Elles transmettent les valeurs le long des chaînes de découpages pourvu que leur traitement entre dans le cadre du modèle d'états finis.

*Exemple d'affectation de variables.*

DRV: = VBAJAV

*Contrôle de la segmentation.*

Un certain nombre de fonctions standard permettent d'intervenir au niveau de l'algorithme de segmentation. Soit en affectant une priorité absolue au segment en cours (-ARRET-), soit par l'arrêt de tout nouveau découpage si l'on estime avoir obtenu l'unique résultat désiré (-FINAL-), soit en ne prenant pas en compte les segments plus courts que le segment actif (-STOP-).

*Contrôle des dictionnaires et de la valeur de l'unité lexicale.*

L'utilisation des dictionnaires est de type état-fini. La segmentation correspond donc à un langage régulier sur le vocabulaire terminal des dictionnaires références. Une utilisation standard: désinence, base, préfixe a été intégrée à l'algorithme. L'utilisateur peut garder néanmoins le contrôle de ses dictionnaires à partir de la grammaire en utilisant la variable DICT. L'ensemble des dictionnaires valides en début de découpage des formes doit figurer dans la grammaire.

*Affectation de la valeur d'unité lexicale.*

Elle se fera de façon standard à partir de la seule entrée du lexique comportant une référence d'unité lexicale (dictionnaire de type base). Sinon l'affectation de cette valeur peut être imposée par la grammaire. (utilisation de la variable exclusive UL).

Exemple: UL(C): = 'UL100'

*Segmentation impossible.*

Si aucune segmentation valide n'est trouvée pour une forme, on crée une référence à un format morphologique particulier (MODINC). Celui-ci peut figurer en partie gauche d'un ensemble de règles constituant une sous-grammaire. Elle doit contenir une règle dont l'application est inconditionnelle (MOTINC), qui assure un résultat pout toute forme figurant dans le texte d'entrée. En parallèle est créée une entrée dans un dictionnaire temporaire associé au texte traité. Il est d'ailleurs possible de créer en cours de traitement de telles références qui trouvent

leur utilité lors de la reconnaissance des noms propres, par exemple (-TRANS-).

*Mots composés et tournures.*

De façon a permettre la reconnaissance des mots composés, une fonction (-SOL-) s'interprétant par l'extraction d'une solution correspondant à l'état courant a été introduite. Inversement, l'utilisation d'un dictionnaire de tournures figées peut conduire à associer une seule solution à un groupe de formes (TOURN).

*Autres fonctions.*

Fonction d'élimination: elle correspond à l'élimination de l'ensemble des solutions associées à la forme et permet de garder une représentation où des éléments non significatifs sont supprimés. Un certain nombre de fonctions permettent l'accès aux chaînes littérales constituant les formes. Leur transformation se traduit par des réductions du nombre des articles du lexique (cas des redoublements de consonnes).

*Exemple de règles grammaticales.*

```
VAR(C) : = VAR(A), GNR (C) : = MAS, NBR(C) : = SIN, DICT(C)
   : = 3
RP1   : PM1 - PM2 - PM3 - PM4 ==
        VAREM (C) : = VAREM(A), NEG(C) : = NG /
        SEG(C) -E- B -ET- PREF(C) -E- PREF(A)
        -ET- (DRV(C) -E- DRVO -ET- PDRN(C) -INC- N
        -OU- DRV(C) -E- AJAV  -ET- PDRN(C) -INC- NAM
        -OU- DRV(C) -E- AJNM  -ET- PDRN(C) -INC- NAN
        -OU- DRV(C) -E- VBAJ  -ET- PDRN(C) -INC- NVA
        -OU- DRV(C) -E- VBAJAV -ET- PDRN(C) -INC- NVAM).
RD31 : D3 ==
        VAREM(C) : = VAREM(A), VARNM(C) : = VARNM(A) -U-
        VARNM(C) /
        GNR(A) -E- FEM -ET- NBR(A) -E- SIN -ET-
        SEG(C) -E- S -ET- (DRV(C) -E- AJAV -OU- DRV(C) -E-
        AJNM) /
        TCHAINE(0, 'LL', 'L'), TCHAINE(0, 'NN', 'N'), TCHAINE(0,
        'SS', 'S'),
        TCHAINE(0, 'TT', 'T').
RS21 : SM3 ==
        VAR(C) : = VAR(A), NBR(C) : = SIN / SEG(C) -E- SEG0 /
        TCHAINE(0, 'AL', 'ELLE').
```

```
RA5  : BA3 - BA4 - BA6 - BA7 - BA8 - BA9 - BA10 - BA11 ==
       VAREM(C) : = VAREM(A), VARNM(C) : = VARNM(A) -U-
       VARNM(C), DICT(C) : = 3, -ARRET- /
       SDNA(C) -I- SDNA(A) -NE- SDNA0 -ET- DRV(C) -NE-
       DRV0
       -ET- (PDRP(A) -INC- AM -OU- PDRP(A) -INC- AN
       -OU- (PDRN(A) -INC- NAM -OU- PDRN (A) -INC- NAN)
       -ET- SCHAINE (A, 0, 1) -NE- ' ').
```

2.4. *Les dictionnaires.*

Il en existe de deux types: les dictionnaires d'affixes n'introduisant pas de référence au lexique (UL) et les dictionnaires de bases associant à celles-ci des unités lexicales. Chaque article du dictionnaire est composé d'un segment auquel sont associés un format morphologique et un format syntaxique, éventuellement un nom d'unité lexicale.

```
CHEV == BN9 (SN3, CHEVAL).
```

Au nombre de six au maximum, on peut leur adjoindre un dictionnaire de tournures du type base mais contenant plusieurs formes par article.

De façon interne, l'organisation des dictionnaires est basée sur l'utilisation d'une fonction de hash-coding déterminée par le premier caractère du segment et la longueur de celui-ci.

On détermine ainsi un certain nombre de classes. A l'intérieur une organisation séquentielle monotone permet une recherche dichotomique des articles par l'automate.

*Exemples*: *entrées d'un dictionnaire de type base*

```
RE1EL == BA4 (SA534, RE1EL).
```

*entrées d'un dictionnaire d'affixes (préfixes)*

```
IR    == PM4 (PS).
```

*entrées d'un dictionnaire d'affixes (désinences)*

```
E     == D3 (VS3).
ITE1  == SM3 (SS2).
TE1   == SM2 (SS2).
```

## 3. L'ALGORITHME

### 3.1. *Son principe.*

L'algorithme a pour rôle de segmenter les formes constituant le texte d'entrée. Le sens de la segmentation est constant pour une langue donnée et s'effectue à partir de l'extrémité droite ou gauche de la forme. Ce paramètre sens de l'analyse doit être précisé au moment du traitement des données linguistiques.

A chaque étape de la segmentation, la chaîne de caractères restant à analyser sert d'argument pout la recherche des segments dans les différents dictionnaires validés. Cette chaîne sera désignée par *A*. Les segments pris en compte sont ceux qui constituent les sous-chaînes initiales de la chaîne *A*. L'ordre selon lequel ils sont retrouvés est fonction du dictionnaire où ils figurent, et à l'intérieur d'un dictionnaire de leur taille (ordre décroissant). Pour un même segment, les formats associés sont traités dans l'ordre où ils apparaissent dans la grammaire. Il en est de même pour l'application des règles.

### 3.2. *Son contrôle.*

La fonction -INIT- a pour rôle de réinitialiser l'automate et doit être notée en partie affection de règle grammaticale. Elle se traduit par l'annulation des liaisons établies avec les solutions précédentes. On découpe ainsi le texte en unités syntaxiquement autonomes équivalentes à des phrases.

La fonction -ARRET- permet d'ignorer toute nouvelle sous-segmentation.

La fonction -FINAL- donne une priorité absolue au découpage en cours.

### 3.3. *Modalités de fonctionnement.*

Les résultats de l'analyse d'un texte peuvent être destinés à fournir uniquement une entrée au modèle suivant. Dans ce cas aucun résultat externe des découpages n'est fourni.

Pour la mise au point du modèle lui-même, on peut obtenir deux types de sortie des résultats:

— un détail complet des applications des règles et des grandeurs calculées

— une sortie du découpage des formes et des valeurs associées seulement.

## 4. RÉALISATION, APPLICATION DU SYSTÈME

Le système A.T.E.F. a été implémenté sur ordinateur IBM 360/67. Il utilise un mode de fonctionnement conversationnel par l'intermédiaire des systèmes d'exploitation CP et CMS.

La gestion des programmes et fichiers constituant le système A.T.E.F. est effectuée par un moniteur de procédures de commandes conversationnelles (composant EXEC de CMS).

L'intéraction utilisateur-système est du type question-réponse. Elle est prise en charge par l'interface conversationnel du moniteur.

Deux composants principaux correspondent d'une part à la préparation des données (DICMOR) et d'autre part au traitement des textes (MORPHO).

Les programmes ont été rédigés en langage d'assemblage de façon à atteindre des performances optimales.

Le système opérationnel depuis juillet 1972 a permis la mise au point des analyses morphologiques des langues russe, française, japonaise et allemande. Des applications sur des langages tels que les mots composés en chimie sont en cours.

```
                                    SYSTEME A.T.E.F.
DETAIL DE L'EXECUTION , TEXTE : 1                                   CODE LANGUE : F1

** OCCURRENCE : IRRE1ALITE1
MORPHE :  ITE1                          REGLE : RS2
      ETAT COURANT : K0(UL0,S,/,SDNA(2),SDN(ITE),$,NMC,AJNM,/,GNR(FEM),NBR(SIN))
RESTE : IRRE1ALE
MORPHE :  ALE                           REGLE : RD1     FORMAT : D8     VS3
MORPHE :  ALE                           REGLE : RD3     FORMAT : D8     VS3
      ETAT COURANT : K0(UL0,D,/,SDNA(2,9),SDN(ITE1),$,NMC,AJNM,/,GNR(FEM,NBR(SIN))
RESTE : IRRE1
MORPHE :  E
MORPHE :  E                             REGLE : RD2     FORMAT : D3     VS3
                                        REGLE : RD31    FORMAT : D3     VS3
      ETAT COURANT : K0(UL0,D,/,SDNA(2(3,4,11),SDN(ITE1),$,NMC,AJNM,/,GNR(FEM,NBR(SIN))
RESTE : IRRE1AL
MORPHE :  AL
MORPHE :  AL                            REGLE : RD1     FORMAT : D8     VS6
MORPHE :  ITE1                          REGLE : RD3     FORMAT : D8     VS6
                                        REGLE : RS21    FORMAT : SM3    SS2
      ETAT COURANT : K0(UL0,S,/,SDNA(2),SDN(ITE1),$,NMCAJNM,/,GNR(FEM,NBR(SIN))
RESTE : IRRE1EL_E
MORPHE :  E
MORPHE :  E                             REGLE : RD2     FORMAT : D3     VS3
                                        REGLE : RD31    FORMAT : D3     VS3
      ETAT COURANT : K0(UL0,D,/,SDNA(2,3,4,11),SDN(ITE1),$,NMCAJNM,/,GNR(FEM),NBR(SIN))
RESTE : IRRE1EL
MORPHE :  RE1E_
MORPHE :  RE1EL                         REGLE : RA3     FORMAT : BA4    SA534
                                        REGLE : RA5     FORMAT : BA4    SA534
      ETAT COURANT : K0(RE1EL,B,/,PREF(IR),SDNA(2,3,4,11),SDN(ITE1),PDRP(AM,AN),PDRN(N,NAM,NAN),$,NMC,
AJNM,/,GNR(FEM),NBR(SIN))
RESTE : IR
MORPHE :  IR                            REGLE : RP1     FORMAT : PM4    PS
  ⟶  DECOUPAGE : IR RE1EL E ITE1    K0(RE1EL,P,/,PREF(IR),SDNA(2,3,4,11),SDN(ITE1),PDRP(AM,AN),PDRN(N,
NAM,NAN),$,NMC,AJNM,NG,/,GN(FEM),NBR(SIN))
MORPHE :  RE1E_                         REGLE : RA7     FORMAT : BA4    SA534
```

*Exemple de sortie de traitement.*