Bernard Vauquois; G. Veillon; Jean Veyrunes

Application des grammaires formelles aux modèles linguistiques en traduction automatique

Kybernetika, Vol. 1 (1965), No. 3, (281)--289

Persistent URL: http://dml.cz/dmlcz/125080

KYBERNETIKA ČÍSLO 3, ROČNÍK 1/1965

Application des grammaires formelles aux modèles linguistiques en traduction automatique*

B. VAUQUOIS, G. VEILLON, J. VEYRUNES

Cet article contient l'analyse des conditions pour l'exploitation économique des modèles formels linguistiques pour la traduction automatique.

I. NATURE DU PROBLEME

Pour parvenir à la réalisation d'un programme de traduction automatique, il est nécessaire de disposer d'une description de la langue source et de la langue cible ainsi que de règles de transfert d'éléments de la première langue en ceux de la seconde. Les conditions ideales seraient alors les suivantes:

a) Les descriptions sont exprimées au moyen de méta-langages qui permettent à la fois la représentation de tous les phénomènes de la langue naturelle considérée d'une part, et leur exploitation par un procédé automatique d'autre part.

b) Les éléments en correspondance dans une règle de transfert dérivent d'un élément unique dans un système interprétant au moins les deux langues considérées (langage pivot considéré comme invariant de la transformation).

Pour tendre vers ce but, probablement inaccessible, on est conduit à substituer à chaque langue naturelle A_i, un modèle L_i qui doit satisfaire au moins à la deuxième condition (a) (exploitation par un procédé automatique) et qui est la meilleure approximation possible de A_i.

Si on admet que le langue A_i peut être envisagée à différents niveaux [1], [2], (en général: morphologie, syntaxe, sémantique) on peut réaliser des modèles partiels $L_i^{(1)}$, $L_i^{(2)}, \ldots$ de la langue à ces divers niveaux. Ainsi, on obtient une séparation des difficultés; en outre, pour chaque niveau on peut choisir le système formel le mieux adapté pour réaliser le modèle correspondant. En contrepartie, il faut alors établir l'enchaînement de ces modèles successifs.

* Présenté au *Colloquium sur la linguistique algébraïque et la traduction automatique*, Prague, 18—22 septembre 1964.

Nous nous proposons d'étudier la réalisation de modèles au niveau morphologique et syntaxique. A ce propos les considérations théoriques sur le choix du type logique de modèle, les problèmes d'adéquation et ceux d'interprétation seront juste mentionnés; par contre on s'attachera à montrer l'application effective des grammaires formelles aux modèles linguistiques en abordant quelques problèmes pratiques.

II. MODELE MORPHOLOGIQUE

La réalisation d'un modèle, quelque soit le niveau auquel il se situe, doit reposer sur un certain nombre de données et d'hypothèses. Dans le cas présent, celui de la morphologie, nous admettons les suivantes:

a) La langue naturelle considérée comme langue source construit ses mots, plus exactement les *formes* de ses mots, à partir d'éléments continus; en d'autres termes tout „morphème" est représenté par une suite continue de caractères et aucun morphème ne peut se présenter sous la forme de plusieurs suites de caractères séparées par des intervalles dans lesquels viendraient s'insérer d'autres morphèmes.

b) On suppose ensuite que la forme d'un mot quelconque est obtenu par concaténation de morphèmes et que la chaîne ainsi construite ne possède qu'une seule structure.

c) On choisit alors comme type de modèle, un système d'états finis.

d) Enfin, on admet que le modèle obtenu représente un certain niveau de la langue naturelle et qu'il sera possible de le relier aux modèles correspondant aux niveaux supérieurs.

Ainsi, on considère seulement dans la langue naturelle l'ensemble des formes de ses mots (plus simplement: ensemble des formes F) et l'on cherche à reconnaître ces formes à partir d'un ensemble de morphèmes M au moyen d'un automate d'états finis, et à les interpréter en fonction des besoins demandés par les modèles de niveau supérieur (fig. 1).

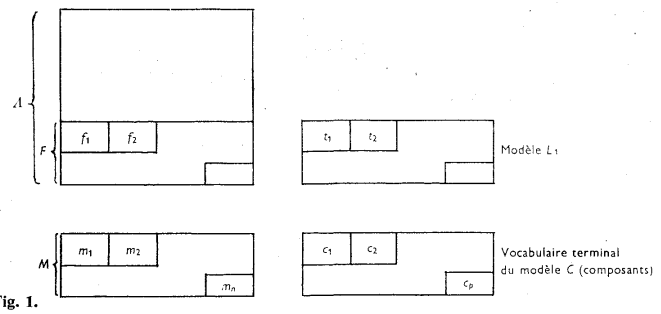

Fig. 1.

Le dictionnaire est une application de M dans l'ensemble des parties de C. On trouvera ailleurs [3], [4], [5] le contenu détaillé des éléments de C et des termes $t_1, t_2, \ldots$ Ici, on se borne à indiquer ce qui est nécessaire pour la réalisation pratique de la grammaire d'états finis. Supposons d'abord que l'on ne s'intéresse qu'à la reconnaissance des formes sans en chercher l'interprétation; il s'agit donc uniquement de décider si une chaine de morphèmes correspond à une forme ou non.

Connaissant l'ensemble des formes et ayant admis que ces formes constituent un langage d'états finis, si l'on cherche les classes d'équivalences des chaines de mor-

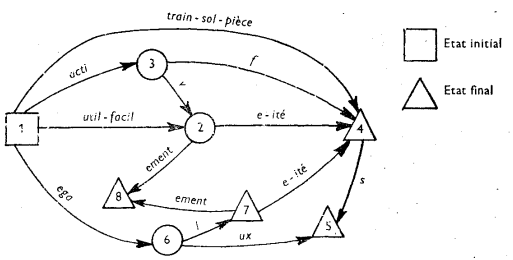

Fig. 2.

phèmes d'après la relation d'équivalence invariante à droite de Nerode, on sait alors que l'on obtient l'automate d'états finis minimum [6].

Exemple:

Actif	*Utile*	*Facile*	*Egal*	*Train*
Actifs	*Utiles*	*Faciles*	*Egale*	*Trains*
Active			*Egaux*	*Sol*
Actives			*Egales*	*Sols*
Activement	*Utilement*	*Facilement*	*Egalement*	*Pièce*
Activité	*Utilité*	*Facilité*	*Egalité*	*Pièces*
Activités	*Utilités*	*Facilités*	*Egalités*	

La recherche de l'automate minimum conduit au résultat suivant (fig. 2).

Cela suffit pour résoudre le seul problème de décision. Comme on doit interpréter la suite des morphèmes utilisés et des états parcourus, la recherche des classes d'équivalences des morphèmes, c'est-à-dire des correspondants (et non plus des chaînes des morphèmes) conduit en pratique à une augmentation du nombre d'états.

En effet si le dictionnaire fournit:

$$\left.\begin{matrix} util \\ facil \end{matrix}\right\} \to C_1\,, \quad acti \to C_2\,, \quad v \to C_3$$

$$e \to C_4\,, \quad ité \to C_5\,,$$

la transition de l'état 2 à l'état 4 par le symbole C_4 devrait générer pour le terme en cours:

ADJECTIF	*MASC.*	*SING*	
	FEM.	*SING*	si l'on vient de l'état 1 ,
ADJECTIF	*FEM.*	*SING*	si l'on vient de l'état 3 .

En considérant le corpus de toutes les formes d'une langue naturelle, il est évident que la grammaire „minimum" est trop volumineuse pour être réalisée pratiquement. Plusieurs procédés, assez voisins, recherchent une simplification.

I. Meltchouk [7] choisit les morphèmes „a priori" et construit un système de classes. A chaque morphème est associé un certain nombre de classes. La concaténation de deux morphèmes est permise si, et seulement si, il existe au moins une classe commune associée à l'un à l'autre. Le résultat fournit une nouvelle association de classes.

Le centre de Berkeley et le CETA ont recours à deux grammaires d'états finis consécutives. L'ensemble des chaînes de morphèmes admises par la première contient, outre celles qui appartiennent au langage, des solutions parasites, mais on en a éliminé un grand nombre. La deuxième grammaire est alors chargée de n'admettre que les solutions exactes et d'en donner leur interprétation.

Cette deuxième grammaire serait encore trop volumineuse. Le principe adopté au CETA est alors le suivant:

à partir de classes de morphèmes choisies a priori, on établit des sur-classes correspondant à des interprétations déterminées. La grammaire est construite au moyen de ces sur-classes. Elle a construit, et permet de reconnaître, des chaînes de morphèmes inadmissibles.

Ainsi les classes:

$$\left.\begin{matrix} facil \\ util \end{matrix}\right\} : C_1 , \quad ega : C_2 , \quad acti : C_3$$

sont regroupées dans

$$\gamma_1 = \{C_1, C_2, C_3\} \quad \text{(bases adjectivales)} ,$$

les morphèmes

$$F, FS, VE, VES$$
(système de désinence)

forment une sur-classe C_4.

Les morphèmes tels que ***ite, ement***, constituent des dérivations interprétables etc. ...

En associant aux morphèmes „bases" non seulement leur sur-classe, mais aussi les sur-classes suivantes admises (éventuellement les précédentes si on utilise des morphèmes „préfixes"), on rétablit l'adéquation de la grammaire en interdisant toute

transition par les sur-classes non mentionnées. Ce principe conduit à faire le choix correct d'une *sous-grammaire*, il permet de traiter facilement les particularités lexicographiques toujours nombreuses dans les langues naturelles.

III. LE MODELE SYNTAXIQUE

Le principe de sélection d'une sous-grammaire dans une grammaire préétablie qui est apparu dans le modèle morphologique va être de nouveau utilisé, avec une efficacité beaucoup plus grande dans le modèle syntaxique.

Tout d'abord, précisons les données du modèle et le résultat auquel on veut parvenir.

Le modèle morphologique a établi une correspondance entre les formes de la langue naturelle et les termes de L_1 en passant par le chemin *ABCD* (fig. 3). Chaque

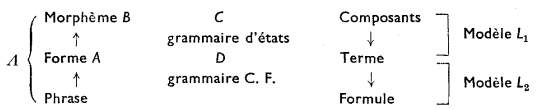

Fig. 3.

terme fournit en particulier, en vue de l'enchaînement de L_1 avec le modèle de niveau immédiatement supérieur L_2 (modèle syntaxique), une catégorie syntaxique et des valeurs de variables grammaticales.

Ainsi la forme française ,,BOIS" conduit aux deux termes dont une partie du contenu est la suivante:

Forme		*Terme*			
	catégorie syntaxique	Genre	Nbre	Personne	Temps – mode nbre
	1) *SUBC*	*MAS*	*SIN, PLU*		
BOIS	2) *VERB*			1,3-*SIN*	*PRES-IND*

On cherche alors à reconnaitre des suites de formes (d'occurrences de la langue naturelle) qui constituent une phrase.

De même qu'en morphologie on a choisi ,,a priori" des morphèmes auxquels ont été associées des classes, en syntaxe on définit aussi le vocabulaire terminal du modèle au moyen de catégories syntaxiques munies de variables grammaticales.

On admet que le type du modèle est ,,context-free" (C. F.).

La structure formelle du modèle L_2 comprend ainsi ce vocabulaire terminal V_T, un vocabulaire non-terminal V_N à définir dont un élément distingué S, et un ensemble fini de règles.

Comme il s'agit d'un problème de reconnaissance, les règles s'écrivent

$$\omega \rightarrowtail A$$

où ω est une chaîne sur le vocabulaire complet et A un élément de V_N.

Ce formalisme doit reconnaitre les phrases du langage, avec toutes leurs structures possibles et son interprétation doit permettre l'enchainement avec le modèle suivant L_3 (modèle sémémique).

Les problèmes posés par la réalisation pratique du modèle L_2 se présentent de la manière suivante:

Pour des raisons de commodité d'utilisation de l'algorithme d'exploitation de la grammaire, cette dernière est écrite sous forme normale; les règles sont donc soit du type (1) soit du type (2).

(1) $\quad a \rightarrowtail A\,, \qquad a \in V_T,\ A \in V_N\,,$

(2) $\quad BC \rightarrowtail A\,, \qquad A, B, C \in V_N\,.$

Avec une telle écriture le nombre des éléments de V_N devient très grand.

En effet soit K_u une catégorie syntaxique, soient $V_{u_1}, \ldots, V_{u_p}$ les variables qui lui sont associées, soit enfin $n(v_{u_i})$ le nombre de valeurs possibles de la variable v_{u_i}. Le nombre d'éléments non terminaux liés à cette catégorie est alors:

$$n_v = \prod_{i=1}^{p} n(v_{u_i})\,.$$

Avec m catégories $(u = 1, 2, \ldots, m)$ on obtient

$$n = \sum_{u=1}^{m} n_u$$

en ne faisant appel qu'aux règles de type (1). De même le nombre de règles de type (2) qu'il faut écrire dépasse la possibilité de réalisation pratique.

On est ainsi conduit à chercher un formalisme équivalent qui conduise à l'écriture d'une grammaire de volume acceptable [8], [9].

La première „contraction" envisagée est évidente et se trouve employée partout. Elle consiste à rendre équivalents tous les éléments non terminaux associés à la même catégorie syntaxique. Les variables grammaticales qui alors n'y figurent plus, apparaissent sous forme de conditions d'application d'une règle de type (2); ces conditions sont d'ailleurs soit des intersections non vide sur des variables communes soit des valeurs de variables fixées d'avance.

Exemple:

Catégories syntaxiques:	*SUBC* (substantif commun) *ADJQ* (adjectif qualificatif) *ARTC* (article)
Variables:	*GNR*: Genre (*MAS*: masculin; *FEM*: féminin) *NBR*: Nombre (*SIN*: singulier; *PLU*: pluriel)
Eléments non-terminaux:	*GNOMO*: groupe nominal *ADJGO*: adjectif général *ARTCO*: article

Au lieu de 12 règles de type (1), (4 par catégorie) on n'en écrit ainsi que 3 (une par catégorie).

Au lieu de 4 règles réalisant le groupement „substantif-adjectif" on n'en écrit qu'une avec des conditions d'accord grammatical, etc. ..

La deuxième contraction fait appel au principe de sélection d'une sous-grammaire. Soient A, B, C, A' des éléments non-terminaux. Appelons R_A l'ensemble des règles de type (2) qui contiennent A à gauche de $\rangle\!\!-$ (A en partie gauche). Soit la règle $BC \rangle\!\!- A'$ et appelons $R_{A'}$ l'ensemble des règles qu'il est nécessaire d'écrire avec A' en partie gauche.

Si on substitue A à A' dans $R_{A'}$ on obtient un ensemble de règles $R_{A'\to A}$. Supposons que $R_{A'\to A} \subset R_A$. On peut alors faire l'économie de la classe A' et des règles $R_{A'}$ en écrivant

$$BC \rangle\!\!- A$$

et en indiquant que l'élément résultant A ne peut être utilisé que dans le sous-ensemble $R_{A'\to A}$ de R_A c'est-à-dire en interdisant les règles de

$$\{R_A \to R_{A'\to A}\} = \Sigma_A .$$

Ainsi au moyen de $BC \rangle\!\!- A/\Sigma_A$ on réalise la sélection d'une sous – grammaire. Les interdictions ainsi exprimées se transmettent tant que la classe reste la même au cours d'une construction. Si l'interdiction porte sur une règle de la forme $AA \rangle\!\!- X$ on indique évidemment si elle intervient sur le premier A, sur le second ou sur les deux. Enfin dans le cas d'une règle $AA \rangle\!\!- A$ on peut aussi décider de la transmission.

La même procédé est utilisable pour les règles de type (1) $a \rangle\!\!- A/\Sigma_A$. Son emploi permet d'une part de regrouper diverses catégories syntaxiques (éléments terminaux) en une seule classe (élément non-terminal) et aussi de pouvoir définir un code syntaxique permettant de tenir compte du comportement particulier d'une unité lexicale par rapport à sa catégorie.

Cette contraction conserve la nature „context-free" du modèle [8].

Outre la réduction appréciable du nombre de classes et de règles ce procédé permet de diminuer le nombre des structures. On sait que les grammaires formelles, modèles de syntaxe, conduisent à une multiplicité de structures dont peu correspondent à des ambiguités réelles dans la langue naturelle source; d'autre part, on peut vouloir regrouper sous une même structure diverses solutions dont on sait énumérer les interprétations.

Examinons quelques exemples simples des propriétés énumérées plus haut.

$$a \rangle\!\!- A/\Sigma_A .$$

Ainsi

$SUBC \rangle\!\!- GNOMO$ (avec transmission des variables grammaticales),

alors

$$PROJ \rangle\!\!- GNOMO/\Sigma$$

où $PROJ$: pronom personnel et Σ indique les numéros de règles (1) et (2).

(1) *ARTCO GNOMO* $\rangle\!-$ *GNOMO*/Σ' (accord en genre et nombre),

(2) *GNOMO ADJGO* $\rangle\!-$ *GNOMO* (accord en genre et nombre),

(3) *GNOMO VERBO* $\rangle\!-$ *VERBO*/Σ'' (accord en personne et nombre).

De même Σ' indique la règle (1) puisque le substantif ne peut être précédé que d'un seul article; Σ' indique aussi la règle (2) pour éviter deux structures à la chaîne

ARTC SUBC ADJQ

qui alors n'en a plus qu'une.

Le modèle syntaxique du russe a ainsi conduit à une grammaire comprenant 71 éléments terminaux, 40 éléments non terminaux, 77 règles de type (1) et 270 règles de type (2); [10].

Enfin la réalisation pratique a posé encore un problème: celui des accords grammaticaux à distance entre deux occurrences (par exemple: pronom relatif avec son antécédent, le pronom relatif étant déjà entré dans la construction de la proposition relative) et d'une manière générale le problème des constituants discontinus.

Du fait du choix d'un modèle ,,context-free" il fallait bien s'attendre à trouver là quelque difficulté.

La notion de variable grammaticale, déjà utilisée pour servir de condition d'application d'une règle, peut alors être étendue. Aux variables déjà existantes, on peut ajouter des variables dites ,,véhiculaires" qui se transfèrent vers l'élément de partie droite jusqu'à ce qu'une règle y fasse appel. Si le nombre de ces variables véhiculaires qu'on peut ainsi trouver au cours d'une construction est limité, on réalise encore un moyen de contraction du modèle ,,context-free". Si ce nombre n'est pas borné (ouverture d'une liste de variables véhiculaires) le modèle devient-il type ,,context-sensitive"? La réponse semble affirmative mais une démonstration formelle reste à faire.

(Reçu le 13 novembre 1964.)

BIBLIOGRAPHIE

[1] Lamb S.: The nature of the machine translation problem. Conference on M.T. Washington D.C. (1962).

[2] И. И: Ревзин: Модели языка: Издательство Академии наук СССР, 1962.

[3] Vauquois B.: Langages artificiels — Systèmes formels et traduction automatique. Ecole d'été de Venise — juillet 1962.

[4] Vauquois B., Veyrunes J.: Présentation de l'analyse morphologique du russe. Document CETA: G-100-C (1962).

[5] Veillon G.: Présentation de l'analyse morphologique et du programme de dictionnaire allemand. Document CETA: G-500-A (1963).

[6] Rabin M. O., Scott D.: Finite automata and their decision problems. IBM Journal, 3 (1959), 115—125.

[7] И. А. Мельчук: Проблемы кибернетики, Вып. 6, 1961.

[8] Veillon G., Veyrunes J.: Etude de la réalisation pratique d'une grammaire „Context-Free" et de l'algorithme associé. Document CETA: G-001-1 (1964).
[9] Colombaud J.: Langages artificiels en analyse syntaxique. Thèse de 3e cycle. Université de Grenoble 1964.
[10] Maksimenko, Torre, De Crousnilhon: Modèle de la syntaxe russe: Structures abstraites dans une grammaire „contrext-free". Document CETA: G-201-1 (1964).

VÝTAH

Aplikace formálních gramatik na lingvistické modely v strojovém překladu

B. VAUQUOIS, G. VEILLON, J. VEYRUNES

Autoři nejprve určují místo zkoumaného problému v rámci systému strojového překladu založeného na sepětí logicko-lingvistických modelů. Dále ukazují jak mohou být typy konečně stavové, resp. nekontextové gramatiky užity v morfologii, resp. v syntaxi některých jazyků (ruštiny, francouzštiny apod.). Teoretické výsledky studia těchto gramatik nejsou přímo aplikovatelné na sestavování modelů, a to kvůli příliš velkému počtu tříd a pravidel, které by bylo třeba zavést; mimo to by u nekontextových gramatik došlo k příliš velké dvojznačnosti struktur.

Zavedením transformačních principů se ukazuje, jak realizovat model, v němž se počítá s lexikálními zvláštnostmi, který generuje podgramatiky tak, že se značně redukuje počet tříd a pravidel, jakož se i omezuje počet dvojznačností předchozím výběrem ekvivalencí struktur.

B. Vauquois, G. Veillon, J. Veyrunes, CETAG, Faculté des Sciences, Grenoble, France.