

# Levée d’ambiguités par grammaires locales

Éric Laporte\*

## 1 Introduction

De nombreux mots sont ambigus quant à leurs catégories grammaticales : ainsi, *montre* peut être nom ou verbe. Toutefois, lorsqu’un mot apparaît dans un texte, cette ambiguïté se réduit généralement beaucoup : dans *Une des études les plus importantes montre que l’expérience est critique*, le mot *montre* ne peut être qu’un verbe. Un système d’étiquetage lexical est un système qui attribue des catégories lexicales aux mots. Lever des ambiguïtés de catégories lexicales consiste à utiliser le contexte pour réduire le nombre de catégories lexicales associées aux mots. La levée des ambiguïtés de catégories lexicales est un des principaux défis de l’étiquetage lexical.

Le problème d’étiqueter les mots par des catégories lexicales se pose fréquemment dans le traitement des langues naturelles, par exemple pour la correction orthographique, la vérification grammaticale ou stylistique, la reconnaissance d’expressions, la phonétisation, l’analyse de corpus de textes... En analyse syntaxique, l’étiquetage correct des mots fait partie des résultats de l’analyse, mais si on a préalablement affecté les mots de leurs catégories lexicales, le reste de l’analyse en est souvent facilité (Milne, 1986 ; Hindle, 1989 ; Rimon, Herz, 1991 ; Cutting et al., 1992). Les grands corpus désambiguïsés sont une vaste source d’informations utiles dans de nombreuses applications, et sont notamment mis à contribution pour l’apprentissage des systèmes probabilistes, mais leur étiquetage manuel est lent, coûteux et source d’erreurs. Les systèmes d’étiquetage lexical sont ainsi utiles comme composant initial de nombreux systèmes de traitement de langues naturelles.

## 2 Méthodologie

### 2.1 Les étiquettes lexicales

Un système d’étiquetage lexical attribue à chaque forme d’un texte une ou plusieurs étiquettes, c’est-à-dire des codes qui renferment des informations lexicales. Si ces informations se réduisent à la catégorie grammaticale, on compte de 10 à 20 catégories lexicales. Si elles incluent d’autres données grammaticales, les catégories lexicales sont plus fines et plus nombreuses. Ces données grammaticales peuvent comporter :

- la forme canonique, par exemple *montrer* pour *montrée*, ou les informations requises pour la reconstituer à partir de la forme trouvée dans le

---

\*Institut Gaspard-Monge, Université de Marne-la-Vallée, 2, rue de la Butte-Verte, F-93166 Noisy-le-Grand CEDEX, France.

texte ;

- les traits flexionnels : temps, personne, genre, nombre... ;
- la délimitation des mots composés, c'est-à-dire des séquences figées comportant plusieurs mots simples séparés par des séparateurs graphiques, comme *bien sûr* ou *traitement de texte*. Dans ce cas, les composés sont étiquetés en tant que tels, par exemple *Adverbe* pour *bien sûr*.
- Si les étiquettes donnent explicitement les formes canoniques, et aussi les codes de flexion (les numéros de conjugaison par exemple), toute forme peut être déduite de son étiquette.

Dans la levée d'ambiguités lexicales, on prend rarement en compte les informations de plus haut niveau, telles que la relation syntaxique avec le prédicat (Koskenniemi, 1990).

Le corpus étiqueté Brown utilise un ensemble de 87 étiquettes simples (Garside, Leech, Sampson, 1987, pp. 165-183) réutilisé dans d'autres projets. Pour le français, un ensemble d'étiquettes qui donne seulement la catégorie grammaticale et les traits flexionnels a à peu près la même taille. Dans cet article, nous décrivons des expériences en français avec des catégories lexicales qui incluent la forme canonique dans les informations lexicales, à la fois pour les mots simples et pour les mots composés. La taille de l'ensemble d'étiquettes est ainsi celle d'un dictionnaire de la langue.

## 2.2 La forme du texte désambiguïsé

La plupart des systèmes de levée d'ambiguités produisent pour un texte donné une séquence de paires mot/étiquette : chaque mot reçoit une étiquette unique. Ce choix peut a priori s'appuyer sur deux présupposés :

1. qu'il est possible de mettre au point un système d'étiquetage lexical qui affecte une étiquette unique à chaque mot sans aucune erreur ;
2. ou que le fait d'attribuer à un mot dans un texte une seule étiquette fausse n'est pas un défaut sérieux d'un système d'étiquetage lexical.

L'affirmation 1 n'a pas un statut clair, sauf à considérer un analyseur syntaxique comme un module d'un système d'étiquetage lexical au lieu de l'inverse : même si l'ensemble d'étiquettes est rudimentaire, l'étiquetage lexical correct de certaines phrases naturelles met en jeu la reconnaissance de leur structure syntaxique globale ou même la compréhension de leur sens. L'affirmation 2 est également contestable, surtout dans le contexte de l'analyse syntaxique : il est parfois impossible de corriger à la main les sorties d'un système d'étiquetage lexical ; or il est naturel qu'un analyseur syntaxique élimine des hypothèses, il l'est moins qu'il en crée de nouvelles avec des catégories grammaticales différentes de celles de départ. De plus, si on étiquette chaque mot d'une façon unique, même les phrases effectivement ambiguës sont représentées comme non ambiguës.

Un certain nombre de systèmes récents d'étiquetage lexical (Silberztein, 1989 ; Koskenniemi, 1990 ; Rimon, Herz, 1991 ; Roche, 1992) produisent, au contraire, plusieurs solutions lorsque le texte est lexicalement ambigu ou que l'unique solution correcte ne peut pas être trouvée. Ces contributions visent à garantir un taux de silence nul : la ou les étiquettes correctes pour un mot ne doivent jamais être éliminées. Cet objectif est rarement évoqué en-dehors de ces auteurs, et peu réaliste pour les systèmes qui étiquettent chaque mot de façon unique, à moins de postuler le présupposé 1.

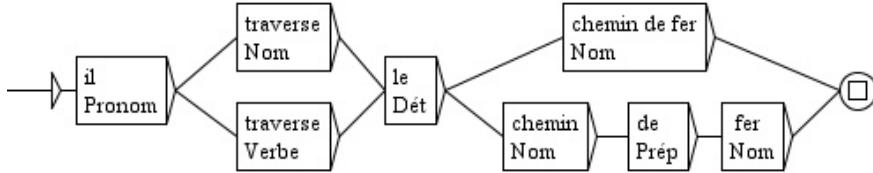


FIGURE 1 – un automate acyclique pour *Il traverse le chemin de fer*.

Les étiquettes des mots d'une même phrase ne sont pas indépendantes, c'est pourquoi le résultat d'un système à plusieurs solutions pour une séquence donnée de mots est un ensemble d'une ou plusieurs séquences d'étiquettes. L'ensemble des séquences d'étiquettes sélectionnées pour une séquence d'entrée donnée est représenté sous une forme appropriée. Comme il s'agit d'un ensemble fini de séquences qui ont généralement beaucoup en commun, cette forme est toujours celle d'un automate fini acyclique, également appelé graphe orienté acyclique (DAG ou DAWG en anglais), machine à états finis ou réseau à états finis (Koskenniemi, 1990), graphe de phrase (Rimon, Herz, 1991) ou treillis de mots (Vosse, 1992). Un des avantages des automates acycliques dans ce contexte est qu'ils permettent de systématiser la représentation des ambiguïtés lexicales. Ils sont en effet utilisables avant comme après la levée des ambiguïtés, et que celles-ci soient liées aux catégories grammaticales, aux traits flexionnels, ou à la délimitation des composés (pour représenter la distinction entre un composé et une séquence de mots simples). La figure 1 illustre l'ambiguïté entre catégories grammaticales pour *traverse* et l'ambiguïté entre composé et mots simples pour *chemin de fer*.

### 2.3 Étiquetage initial et levée d'ambiguïtés

La plupart des systèmes d'étiquetage lexical divisent la tâche en deux étapes : en premier lieu, lors d'un étiquetage initial, les formes sont considérées indépendamment de leur contexte pour dresser la liste de toutes les étiquettes pour chaque mot ; ensuite, lors de la levée d'ambiguïtés, on prend en compte le contexte pour sélectionner une partie de l'ensemble des séquences étiquetées initiales.

Dans d'autres systèmes d'étiquetage lexical (Klein, Simmons, 1963 ; Dermautas, Kokkinakis, 1989 ; Pelillo, Refice, 1991 ; Brill, 1992 ; Federici, Pirrelli, 1992), les deux sous-tâches sont effectuées en même temps et un résultat désambiguisé est construit directement, généralement pour éviter la construction d'un grand dictionnaire.

Plusieurs arguments viennent en faveur de la solution modulaire. Les deux sous-tâches sont clairement définies. Une fois choisi un ensemble d'étiquettes lexicales, les deux sous-tâches sont indépendantes. Il peut en être de même des méthodes permettant de les effectuer avec les meilleurs résultats : perfectionner l'étiquetage initial est un problème de description morphologique des mots, améliorer la levée des ambiguïtés met en jeu la description grammaticale de séquences de mots.

Cette solution est cohérente avec l'utilisation d'automates acycliques pour représenter les ambiguïtés lexicales. Après l'étiquetage initial, l'ensemble des séquences reconnues par l'automate est l'ensemble des séquences d'étiquettes

possibles a priori pour la séquence d'entrée. Lors de la levée d'ambiguïtés, l'automate est modifié. Le nombre de séquences reconnues par l'automate diminue, mais le nombre d'états et de transitions dans l'automate peut croître ou décroître.

La solution modulaire a bien sûr un intérêt particulier si l'on dispose d'un dictionnaire morphologique fiable qui donne pour chaque forme, simple ou composée, la liste des étiquettes possibles. Dans ce cas, l'étiquetage initial est simplement mené à bien par une consultation du dictionnaire. Un tel environnement pour le français a été développé au LADL<sup>1</sup> et au CERIL<sup>2</sup> avec les dictionnaires DELAF (Courtois, 1990) et DELACF (Silberztein, 1990), et avec les algorithmes de compression et de consultation mis en œuvre par Revuz (1991) et Roche (1992) pour obtenir de meilleures performances qu'avec les arbres lexicographiques (*tries*) de Knuth (1973). Il est maintenant intégré au système d'analyse lexicale INTEX (Silberztein, 1993). La taille du dictionnaire comprimé est inférieure à 900 Ko pour 700.000 formes.

## 2.4 Données élaborées à la main ou données statistiques

Les informations utilisées par les systèmes d'étiquetage lexical pour lever des ambiguïtés consistent soit en connaissances grammaticales formalisées à la main, soit en données statistiques acquises par apprentissage automatique dans un grand corpus de textes. Le système de Hindle (1989) utilise un mélange des deux. Comme exemples de systèmes d'étiquetage lexical qui utilisent des données grammaticales élaborées à la main, on peut citer ceux de Klein, Simmons (1963), Hindle (1983), Silberztein (1989), Paulussen, Martin (1992), Roche (1992). Dans le cas de Rimon, Herz (1991), les données sont produites automatiquement à partir de grammaires algébriques faites à la main. Dans toutes ces contributions, les données grammaticales sont formalisées dans des automates finis, ou pourraient facilement l'être. Nous désignons ces données par le nom de grammaires locales, car elles ne constituent jamais une grammaire complète de la langue. L'étiquetage lexical à partir de statistiques est représenté par Greene, Rubin (1971), Hindle (1989), Brill (1992), Federici, Pirrelli (1992), qui utilisent des règles produites par des processus statistiques ; et par Marshall (1983), Jeelinek (1985), DeRose (1988), Cutting et al. (1992), etc., qui utilisent des tables de statistiques, par exemple des paramètres de modèles de Markov. Toutes ces contributions donnent à chaque mot une étiquette unique.

Les arguments pour ou contre ces deux types de méthodes tiennent souvent à leur capacité à faire face à la variété qui caractérise le texte réel dans toute sa généralité. Certains doutent que des connaissances linguistiques élaborées à la main puissent tenir compte de tout ce qui peut se présenter dans du texte. D'autres pensent que des informations apprises automatiquement dans un corpus, même vaste et soigneusement composé d'échantillons variés de types de textes, ne sont pas assez précises pour de nouveaux textes. L'utilisation de connaissances linguistiques élaborées à la main nous semble en tous cas mieux adaptée pour atteindre l'objectif d'un taux de silence nul, c'est-à-dire pour garantir qu'une analyse n'est éliminée que si elle est sans aucun doute incorrecte. L'auteur des données linguistiques peut s'aider d'un corpus de textes, mais doit pouvoir créer des contre-exemples qui n'y figurent pas, de sorte que les données

---

1. Laboratoire d'automatique documentaire et linguistique, Université Paris 7, 2, place Jussieu, F-75252 Paris CEDEX 05, France.

2. Centre d'études et de recherches en informatique linguistique, Institut Gaspard-Monge.

linguistiques qu'il élabore soient indépendantes de ce corpus.

La suite de cet article concerne une méthode de levée d'ambiguïtés lexicales adaptée à l'objectif d'un taux de silence nul (Silberztein, 1989, 1993) et mise en œuvre dans le système INTEX de Silberztein (1993). Nous présentons ici une description formelle de cette méthode. Elle combine la possibilité de plusieurs solutions — le résultat produit pour une séquence de mots donnée est un automate acyclique —, le parti pris modulaire — l'étiquetage lexical et la levée des ambiguïtés sont considérés comme indépendants une fois choisi un ensemble d'étiquettes —, et l'utilisation de connaissances linguistiques élaborées à la main, pour l'étiquetage initial — de grands dictionnaires morphologiques — comme pour la levée des ambiguïtés — des données grammaticales appelées grammaires locales. La seule autre contribution qui rentre dans ce cadre est, à notre connaissance, celle de Roche (1992). Toutes les deux comprennent des algorithmes et leur mise en œuvre et elles utilisent les mêmes dictionnaires. Pour une comparaison entre ces deux systèmes, cf. Laporte (1994).

### 3 Les étiquettes lexicales dans INTEX

#### 3.1 Étiquettes grammaticales complètes

La confrontation d'un texte avec les dictionnaires produit plusieurs séquences d'étiquettes grammaticales en raison des ambiguïtés lexicales. Ainsi, pour la phrase *Je ne me le suis pas fait confirmer sur le moment*, on obtient l'étiquetage initial suivant :

(”je.” + ”je.PRO :1s”)  
(”ne.” + ”ne.XI” + ”ne.XI[+ Préd]”)  
”me.PRO :1s”  
(”le.DET :ms” + ”le.PRO :3ms”)  
(”être.V :P1s” + ”suivre.V :P1s :P2s :Y2s”)  
(”pas.ADV” + ”pas.N :ms :mp” + ”pas.XI”)  
(”faire.V :Kms :P3s” + ”fait.A :ms” + ”fait.N :ms” + ”fait.XI[+  
Préd]”)  
”confirmer.V :W”  
(  
”sur/le/moment.ADV;PDETC”  
+  
(”sur.A :ms” + ”sur.PREP”)  
(”le.DET :ms” + ”le.PRO :3ms”)  
”moment.N :ms”  
)

Les unités de base d'INTEX donnent la forme canonique (mot simple ou mot composé), la catégorie grammaticale, et les traits flexionnels s'ils sont pertinents. Nous les appellerons étiquettes grammaticales complètes et nous les noterons comme dans les exemples suivants : <*suivre V :P2s*>, <*sur/le/moment ADV;PDETC*>, <*coup/fumant N;NA :ms*>...

#### 3.2 Étiquettes grammaticales incomplètes

Le système de levée d'ambiguïtés grammaticales d'INTEX utilise un autre ensemble d'étiquettes lexicales que nous appellerons étiquettes grammaticales

incomplètes et qui peuvent spécifier

- une forme canonique : <prendre>, <le>, <coup/fumant>...
- une catégorie grammaticale : <V>, <ADV>, <A>...
- une forme canonique et des traits flexionnels pertinents pour cette forme :  
    <prendre :P3s>, <prendre :P>, <prendre :s>, <coup/fumant :ms>...
- une catégorie grammaticale et des traits flexionnels pertinents pour cette catégorie : <V :P3s>, <V :P>, <V :s>, <N :ms>...
- une forme simple : *vient, aussitôt, fait...*
- L'étiquette <MOT> représente toutes les formes simples, à l'exclusion des formes composées.

Une étiquette grammaticale incomplète sert à représenter l'ensemble des étiquettes grammaticales complètes qui satisfont aux informations qu'elle spécifie.  
Exemples :

<prendre> représente toutes les formes du verbe *prendre*  
<A> représente tous les adjectifs à toutes les formes  
*suis* représente toutes les étiquettes lexicales possibles de *suis*, c'est-à-dire une forme du verbe *être* et trois formes du verbe *suivre*.

En d'autres termes, une étiquette grammaticale complète est ou n'est pas conforme à une étiquette grammaticale incomplète. Exemples :

<suivre V :P2s> est conforme à <suivre> et à <V>  
<être V :P1s> est conforme à <être> et à <V :P>  
<suivre V :P2s> n'est pas conforme à <être> car la forme canonique ne correspond pas  
<coup/fumant N;NA :ms> n'est pas conforme à <MOT> car c'est une forme composée  
<suivre V :P1s> et <suivre V :P2s> sont conformes à *suis*  
<être V :P1s> est conforme à *suis*

## 4 Grammaires locales de levée d'ambiguités

Dans le système INTEX, la levée d'ambiguités lexicales met en jeu des informations linguistiques élaborées à la main. Ces informations sont spécifiées sous la forme d'un ou plusieurs transducteurs appelés grammaires locales de levée d'ambiguités. On peut voir une telle grammaire locale comme un moyen formel de spécifier un ensemble de séquences grammaticales acceptées. Un algorithme de levée d'ambiguités applique la grammaire à un texte, ce qui consiste à sélectionner parmi les étiquetages grammaticaux du texte ceux qui sont conformes à la grammaire. L'intérêt de ce système est qu'il permet de spécifier des ensembles de séquences grammaticales tout à fait complexes à l'aide de grammaires relativement petites, lisibles et intuitives. Il est important que les grammaires ne rejettent jamais une séquence grammaticale correcte. En revanche, on est loin de disposer d'une grammaire qui n'accepte que les séquences correctes.

Nous utilisons l'éditeur d'automates **Editor** de Max Silberstein pour représenter graphiquement les transducteurs de levée d'ambiguités (exemple : le transducteur  $T_1$ , figure 2). Chaque transition comporte deux étiquettes grammaticales incomplètes, une dite "d'entrée" (dans la boîte) et une "de sortie" (sous la boîte). Dans un transducteur, une séquence d'entrée est une séquence

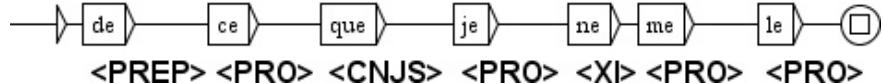


FIGURE 2 – le transducteur  $T_1$ .

d'étiquettes grammaticales d'entrée. Le transducteur sert à mettre en relation des séquences d'entrée et des séquences de sortie. Par exemple, le transducteur  $T_1$  (figure 2) met en relation la séquence d'entrée

*de ce que je ne me le*

et la séquence de sortie

*<PREP> <PRO> <CNJS> <PRO> <XI> <PRO> <PRO>*

Grosso modo, les séquences d'entrée servent à sélectionner les portions du texte auxquelles la grammaire locale va s'appliquer, et les séquences de sortie imposent des contraintes à ces portions de texte. Pour s'assurer qu'une grammaire ne rejette pas de séquences grammaticales correctes, on a besoin d'une règle plus précise pour répondre à la question : étant donné une séquence grammaticale complète et une grammaire locale de levée d'ambiguïtés, la grammaire accepte-t-elle la séquence ?

#### 4.1 Cas où les séquences d'entrée du transducteur ne sont constituées que de formes simples

Dans ce premier cas particulier, on a une règle plus simple que dans le cas général. Nous illustrerons cette règle en prenant l'exemple du transducteur  $T_1$  (figure 2) et de la séquence grammaticale complète

*<cela PRO :ms> <venir V :P3s> <de PREP> <ce PRO :3s> <que CNJS> <je PRO :1s> <ne XI> <me PRO :1s> <le PRO :3ms> <être V :P1s> <pas ADV> <faire V :Kms> <confirmer V :W> <aussitôt ADV>*

Cette séquence est l'étiquetage grammatical correct de

*Cela vient de ce que je ne me le suis pas fait confirmer aussitôt*

La grammaire accepte la séquence si et seulement si on peut diviser la séquence en portions qui sont chacune de l'un ou de l'autre des deux types suivants :

1. la portion est conforme à la fois à une séquence d'entrée et à une séquence de sortie associées par le transducteur.
2. la portion est réduite à une seule étiquette grammaticale complète, peu importe laquelle, mais le texte qui commence à ce mot n'apparaît dans aucune séquence d'entrée du transducteur.

Dans cet exemple (table 1), les deux premières portions sont du type 2 : *<cela PRO :ms>* et *<venir V :P3s>*. La suivante est du type 1 :

*<de PREP> <ce PRO :3s> <que CNJS> <je PRO :1s> <ne XI> <me PRO :1s> <le PRO :3ms>*

Toutes les autres sont du type 2. Finalement cette séquence est acceptée par la grammaire.

Séquences d'entrée du transducteur	Séquence gramm. complète	Séquences de sortie du transducteur
—	<cela PRO :ms>	—
—	<venir V :P3s>	—
de	<de PREP>	<PREP>
ce	<ce PRO :3s>	<PRO>
que	<que CNJS>	<CNJS>
je	<je PRO :1s>	<PRO>
ne	<ne XI>	<XI>
me	<me PRO :1s>	<PRO>
le	<le PRO :3ms>	<PRO>
—	<être V :P1s>	—
—	<pas ADV>	—
—	<faire V :Kms>	—
—	<confirmer V :W>	—
—	<aussitôt ADV>	—

TABLE 1 – une séquence grammaticale acceptée par le transducteur  $T_1$ .

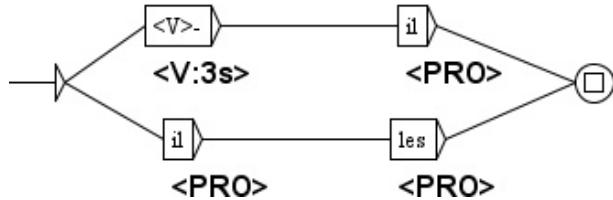


FIGURE 3 – le transducteur  $T_2$ .

On notera que la condition 2 ci-dessus met en jeu non pas seulement l'étiquette grammaticale complète qui figure dans la portion considérée, mais aussi les autres étiquettes complètes associables au même mot, et éventuellement les étiquettes des mots qui le suivent. L'écriture de grammaires locales de levée d'ambiguités s'apparente à la programmation mais nécessite aussi de l'intuition grammaticale, pour imaginer les séquences grammaticales qu'une grammaire va rejeter.

## 4.2 Deuxième cas particulier

En fait, une règle très voisine reste valable pour de nombreux autres transducteurs. Il s'agit des transducteurs dont chaque transition vérifie l'une des deux conditions suivantes :

- l'étiquette en entrée est une forme simple,
- ou bien toute étiquette grammaticale complète conforme à l'étiquette de sortie est conforme à l'étiquette d'entrée.

Par exemple, c'est le cas pour le transducteur  $T_2$  (figure 3) et pour toutes les grammaires locales de levée d'ambiguités du chapitre 8 de Silberztein (1993). La règle est alors la suivante.

La grammaire accepte la séquence si et seulement si on peut diviser la séquence en portions qui sont chacune de l'un *ou* de l'autre des deux types sui-

Séquences d'entrée du transducteur	Séquence gramm. complète	Séquences de sortie du transducteur
—	<ne XI/+ Préd/>	—
<V>- il	<faire V :P3s>- <il PRO :3ms>	<V :3s> <PRO>
—	<le DET :mp>	—
—	<compte N :mp>	—
—	<que CNJS>	—
	...	

TABLE 2 – une séquence grammaticale acceptée par le transducteur  $T_2$ .

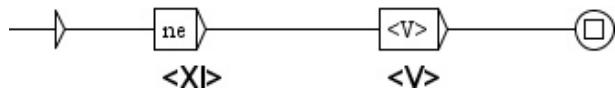


FIGURE 4 – le transducteur  $T_3$ .

vants :

1. la portion est conforme à la fois à une séquence d'entrée et à une séquence de sortie associées par le transducteur.
2. la portion est réduite à une seule étiquette grammaticale complète, peu importe laquelle, mais le texte qui commence à ce mot n'admet aucun étiquetage grammatical qui corresponde à une séquence d'entrée du transducteur.

Nous illustrons cette règle avec  $T_2$  et la séquence grammaticale

<ne XI/+ Préd/> <faire V :P3s>-<il PRO :3ms> <le DET :mp>  
<compte N :mp> <que CNJS>

qui est une partie de l'étiquetage correct de

*Ne fait-il les comptes que pour rendre service ?*

Dans cet exemple (table 2), <il PRO :3ms> <le DET :mp> n'est pas comparé aux séquences d'entrée de la grammaire, car deux portions ne peuvent pas se chevaucher.

Il n'est pas absurde d'avoir des transitions dans lesquelles l'étiquette d'entrée est la même que l'étiquette de sortie. Par exemple, le transducteur  $T_3$  (figure 4) accepte la séquence ci-dessus mais la rejette si on remplace <faire V :P3s> par <fait N :ms> (table 3).

En effet, la portion <ne XI/+ Préd/> <fait N :ms> n'est conforme ni à la séquence d'entrée ni à la séquence de sortie, alors que le texte *ne fait* admet un autre étiquetage grammatical conforme à la séquence d'entrée : <ne XI> <faire V :P3s>. Il n'existe donc aucun découpage en portions qui satisfasse aux conditions de la règle ci-dessus.

## 5 Combinaisons de grammaires locales

INTEX permet d'appliquer plusieurs grammaires locales de levée d'ambiguités à un même texte. Dans ce cas, tout se passe comme si on avait combiné

Séquences d'entrée du transducteur	Séquence gramm. complète	Séquences de sortie du transducteur
ne <V>	<ne XI/+ Préd/> <fait N :ms>-	<XI> <V>
-	<il PRO :3ms>	-
-	<le DET :mp>	-
-	<compte N :mp>	-
-	<que CNJS>	-
	...	

TABLE 3 – une séquence grammaticale rejetée par le transducteur  $T_3$ .

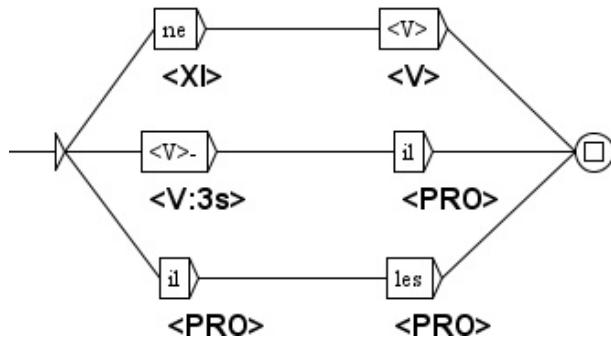


FIGURE 5 – le transducteur  $T_2|T_3$ .

les différents transducteurs en leur donnant le même état initial et le même état final. Par exemple, la combinaison de  $T_2$  (figure 3) et de  $T_3$  (figure 4), notée  $T_2|T_3$ , est représentée figure 5. Les règles ci-dessus s'appliquent alors à l'ensemble.

Si une grammaire locale rejette une séquence grammaticale, et si on combine la grammaire avec une autre, la combinaison obtenue peut accepter la séquence. Par exemple, le transducteur  $T_3$  rejette à tort la séquence

$\langle ne \; XI \rangle \; \langle lui \; PRO \; :3s \rangle \; \langle dire \; V \; :Y2s \rangle \; \langle pas \; ADV \rangle$

qui est l'étiquetage correct de *Ne lui dis pas*, à cause du fait que *lui* est aussi une forme du verbe *luire* :  $\langle luire \; V \; :Kms \rangle$ . Pour tenter de corriger ce défaut, on peut combiner  $T_3$  avec  $T_4$  (figure 6). La séquence correcte ci-dessus est acceptée par  $T_4$  et aussi par la combinaison  $T_3|T_4$  (table 4).

Inversement, la séquence incorrecte

$\langle ne \; XI \rangle \; \langle luire \; V \; :Kms \rangle \; \langle dire \; V \; :Y2s \rangle \; \langle pas \; ADV \rangle$

est rejetée par  $T_4$  mais acceptée par la combinaison  $T_3|T_4$ .

Si deux grammaires locales acceptent une même séquence grammaticale, la combinaison des deux grammaires peut rejeter la séquence. Par exemple, les

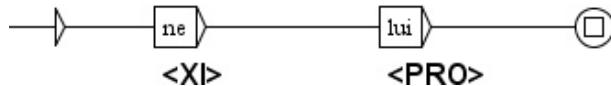


FIGURE 6 – le transducteur  $T_4$ .

Séquences d'entrée du transducteur	Séquence gramm. complète	Séquences de sortie du transducteur
ne	<ne XI>	<XI>
lui	<lui PRO :3s>	<PRO>
–	<dire V :Y2s>	–
–	<pas ADV>	–

TABLE 4 – une séquence grammaticale acceptée par le transducteur  $T_3|T_4$ .

Séquences d'entrée du transducteur	Séquence gramm. complète	Séquences de sortie du transducteur
ne <V>	<ne XI[+ Préd]> <faire V :P3s>-	<XI> <V>
il les	<il PRO :3ms> <le DET :mp>	<PRO> <PRO>
–	<compte N :mp>	–
–	<que CNJS>	–
	...	

TABLE 5 – une séquence grammaticale rejetée par le transducteur  $T_2|T_3$ .

transducteurs  $T_2$  et  $T_3$  acceptent tous les deux la séquence correcte

<ne XI[+ Préd]> <faire V :P3s>-<il PRO :3ms> <le DET :mp>  
<compte N :mp>

mais la combinaison  $T_2|T_3$  (figure 5) la rejette (table 5).

Si deux grammaires locales rejettent une même séquence grammaticale, la combinaison des deux grammaires peut accepter la séquence. Par exemple, chacun des deux transducteurs  $T_5$  (figure 7) et  $T_6$  (figure 8), utilisé séparément, rejette la séquence incorrecte

<pourquoi ADV> <me PRO :3s> <presser V :P3p>-<il PRO :3ms>  
<de PREP> <le PRO :3ms> <lire V :Kms> <dire V :W>,

qui est un étiquetage du texte incorrect

*Pourquoi me pressent-il de le lui dire ?*

Mais la combinaison  $T_5|T_6$  accepte cette même séquence (table 6).

On voit que pour vérifier une grammaire locale de levée d'ambiguités, il ne suffit pas de considérer les chemins du transducteur séparément : on a besoin de vérifier leurs interactions. De même, si on utilise une combinaison de plusieurs

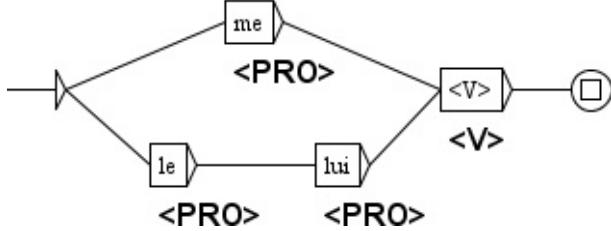


FIGURE 7 – le transducteur  $T_5$ .

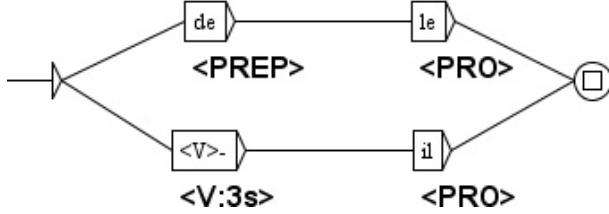


FIGURE 8 – le transducteur  $T_6$ .

Séquences d'entrée du transducteur	Séquence gramm. complète	Séquences de sortie du transducteur
–	<i>&lt;pourquoi ADV&gt;</i>	–
me <V>	<i>&lt;me PRO :3s&gt;</i> <i>&lt;presser V :P3p&gt;</i>	<i>&lt;PRO&gt;</i> <i>&lt;V&gt;</i>
–	<i>&lt;il PRO :3ms&gt;</i>	–
de le	<i>&lt;de PREP&gt;</i> <i>&lt;le PRO :3ms&gt;</i>	<i>&lt;PREP&gt;</i> <i>&lt;PRO&gt;</i>
–	<i>&lt;luire V :Kms&gt;</i>	–
–	<i>&lt;dire V :W&gt;</i>	–

TABLE 6 – une séquence grammaticale acceptée par le transducteur  $T_5|T_6$ .

transducteurs, on ne peut pas prévoir le résultat en les considérant isolément les uns des autres.

## 6 Cas général

Étant donné une séquence grammaticale complète et une grammaire locale de levée d'ambiguïtés, la grammaire accepte-t-elle la séquence ? Contrairement aux règles données plus haut, la règle ci-dessous répond à cette question dans le cas général.

Pour énoncer cette règle on a besoin de la notion suivante : deux séquences grammaticales complètes sont équivalentes si et seulement si elles décrivent le même texte avec la même délimitation des mots simples et des mots composés. Par exemple,

*<superbe N :fs> <gaulliste A :fs>*

et

*<superbe A :fs> <gaulliste N :fs>*

sont équivalents, mais

*<pomme/de/terre N;NDN :fs> <cuire V :Kfs>*

et

*<pomme N :fs> <de PREP> <terre/cuite N;NA :fs>*

ne le sont pas.

La grammaire accepte la séquence si et seulement si on peut diviser la séquence en portions qui sont chacune de l'un des deux types suivants :

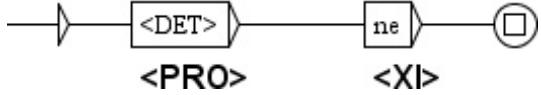


FIGURE 9 – le transducteur  $T_7$ .

Séquences gramm. compl. reconnues	Séquences d'entrée du transd. ( $u$ )	Séquence gramm. complète	Séquences de sortie du transd. ( $v$ )
–	–	<mais CNJC>	–
<aucun DET :ms> <ne XI>	<DET> ne	<aucun PRO :ms> <ne XI>	<PRO> <XI>
–	–	<pouvoir V :P3s>-	–
–	–	<dépasser V :W>	–
–	–	<ce DET :fs>	–
–	–	<limite N :fs>	–

TABLE 7 – une séquence grammaticale acceptée par le transducteur  $T_7$ .

1. la portion est conforme à une séquence de sortie  $v$  du transducteur ; la portion est équivalente à une séquence grammaticale complète conforme à une séquence d'entrée  $u$  associée à  $v$  par le transducteur.
2. la portion est réduite à une seule étiquette grammaticale complète, peu importe laquelle, mais le texte qui commence à ce mot n'admet aucun étiquetage grammatical qui corresponde à une séquence d'entrée du transducteur.

Exemple : le transducteur  $T_7$  (figure 9) accepte la séquence

<mais CNJC> <aucun PRO :ms> <ne XI> <pouvoir V :P3s>  
<dépasser V :W> <ce DET :fs> <limite N :fs>

qui est l'étiquetage correct de

*Mais aucun ne peut dépasser cette limite*

(table 7). En revanche, si on remplace <aucun PRO :ms> par <aucun DET :ms> dans la séquence grammaticale, on obtient une séquence qui est rejetée par  $T_7$ .

## 7 Conclusion

Lorsqu'on examine l'étiquetage initial d'un texte tel qu'il est produit par INTEX, comme dans la section 3.1, des idées de règles de levée d'ambiguités viennent spontanément. Ces idées se présentent parfois sous une forme telle que

Si le déterminant *du* est suivi par un verbe, ce verbe ne peut être  
qu'au participe présent,

c'est-à-dire avec une condition qui reconnaît une configuration grammaticale, et une contrainte grammaticale à imposer lorsque la condition est remplie. Il est facile de concevoir et de réaliser une première version d'une grammaire locale à partir d'une telle idée. Les règles que nous avons citées (sections 4 et 6) pour décider si une grammaire donnée accepte une séquence donnée ne sont guère utiles à ce stade, car le fonctionnement du système est plus intuitif que ces règles ne le laissent penser.

Toutefois, les intuitions grammaticales peuvent se révéler inexactes, à cause bien souvent d'une construction ou d'une ambiguïté à laquelle on n'aura pas pensé. Puisque nous nous sommes fixé l'objectif d'un taux de silence nul, les grammaires locales doivent être testées avec soin. C'est là qu'il est nécessaire de savoir en détail ce que fera une grammaire une fois appliquée à des textes.

## Références

- Brill, Eric. 1992. "A Simple Rule-Based Part of Speech Tagger", *3rd Applied ACL*, Trente (Italie), pp. 152–155.
- Courtois, Blandine. 1990. "Un système de dictionnaires électroniques pour les mots simples du français", in *Langue française 87, Dictionnaires électroniques du français*, Paris : Larousse, pp. 11–22.
- Cutting, Doug, Julian Kupiec, Jan Pedersen, Penelope Sibun. 1992. "A practical part-of-speech tagger", *3rd Applied ACL*, Trente (Italie), pp. 133–140.
- Dermatas, E., G. Kokkinakis. 1989. "A System for Automatic Text Labelling", *Eurospeech 89*, pp. 382–385.
- DeRose, Stephen J. 1988. "Grammatical Category Disambiguation by Statistical Optimization", *Computational Linguistics*, vol. 14, no. 1, pp. 31–39.
- Federici, Stefano, Vito Pirrelli. 1992. "A Bootstrapping Strategy for Lemmatization : Learning Through Examples", *Papers in Computational Lexicography. COMPLEX '92*, F. Kiefer, G. Kiss, J. Pajzs, éds., Institut de linguistique de l'Académie hongroise des sciences, Budapest, pp. 123–135.
- Garside, Roger, Geoffrey Leech, Geoffrey Sampson. 1987. *The Computational Analysis of English*, Londres : Longman.
- Greene, Barbara, Gerald Rubin. 1971. *Automated Grammatical Tagging of English*, Rapport technique, Département de linguistique, Brown University, Providence, Rhode Island.
- Hindle, Donald. 1983. "Deterministic parsing of syntactic non-fluencies", *21st Annual Meeting of the Association for Computational Linguistics. Proceedings of the Conference*.
- Hindle, Donald. 1989. "Acquiring disambiguation rules from text", *27th Annual Meeting of the Association for Computational Linguistics. Proceedings of the Conference*, pp. 118–125.
- Jelinek, F. 1985. "Markov source modeling of text generation", in *Impact of Processing Techniques on Communication*, J.K. Skwirzinski, éd., Dordrecht.
- Klein, S., R.F. Simmons. 1963. "A Computational Approach to Grammatical Coding of English Words", *JACM* 10, pp. 334–347.
- Knuth, Donald. 1973. *The Art of Computer Programming*, Addison-Wesley.
- Koskenniemi, Kimmo. 1990. "Finite-state parsing and disambiguation", *Proceedings of COLING 90*, H. Karlsgren, éd., Université d'Helsinki, pp. 229–232.
- Laporte, Éric. 1994. "Experiments in lexical disambiguation using local gram-

- mars”, *Papers in Computational Lexicography. COMPLEX 94*, Institut de linguistique de l’Académie hongroise des sciences, Budapest, 10 p.
- Marshall, Ian. 1983. “Choice of Grammatical Word-Class Without Global Syntactic Analysis : Tagging Words in the LOB Corpus”, *Computers in the Humanities* 17, pp. 139–150.
- Milne, Robert. 1986. ”Resolving Lexical Ambiguity in a Deterministic Parser”, *Computational Linguistics*, vol. 12, no. 1, pp. 1–12.
- Paulussen, Hans, Willy Martin. 1992. ”DILEMMA-2 : a Lemmatizer-Tagger for Medical Abstracts”, *3rd Applied ACL*, Trente (Italie), pp. 141–146.
- Pelillo, Marcello, Mario Refice. 1991. ”Syntactic disambiguation through relaxation processes”, *Eurospeech 91*, vol. 2, pp. 757–760.
- Revuz, Dominique. 1991. *Dictionnaires et lexiques, méthodes et algorithmes*, Thèse de doctorat, Publication 91-44 du LITP, Université Paris 7, 105 p.
- Rimon, Mori, Jacky Herz. 1991. ”The recognition capacity of local syntactic constraints”, *5th Conference of the European Chapter of the ACL. Proceedings of the Conference*, Berlin, pp. 155–160.
- Roche, Emmanuel. 1992. ”Text disambiguation by finite-state automata, an algorithm and experiments on corpora”, in *COLING-92, Proceedings of the Conference*, Nantes.
- Silberztein, Max. 1989. *Dictionnaires électroniques et reconnaissance lexicale automatique*, Thèse de doctorat, LADL, Université Paris 7, 176 p.
- Silberztein, Max. 1990. ”Le dictionnaire électronique des mots composés”, in *Langue française 87, Dictionnaires électroniques du français*, Paris : Larousse, pp. 71–83.
- Silberztein, Max. 1993. *Dictionnaires électroniques et analyse automatique de textes. Le système INTEX*, Paris : Masson, 233 p.
- Vosse, Theo. 1992. ”Detecting and Correcting Morpho-syntactic Errors in Real Texts”, *3rd Applied ACL*, Trente (Italie), pp. 111–118.