Utilisation d'une structure arborescente pour une hiérarchisation fine des règles de transcription graphème-phonème

Michel Morel, Anne Lacheret-Dujour

Utilisation d'une structure arborescente pour une hiérarchisation fine des règles de transcription graphème-phonème

Michel MOREL, Anne LACHERET-DUJOUR
Laboratoire de Linguistique ELSAP - UPRES-A 6047
Université de Caen, Esplanade de la Paix, 14032 CAEN CEDEX
e-mail : morel@elsap.unicaen.fr, lacheret@elsap.unicaen.fr

Abstract

The grapheme-phoneme transcription module which is presented here is one element of the text-to-speech Kali project which was launched in 1995 at Caen University and which groups together four partners : the research laboratories Elsap and Greyc, the firm Electrel, present since 1981 on the voice synthesis market, and the Club Micro Son, association which unites blind users of computers and provides research, development, training and equipment.

It is pointed out in this paper that a tree-diagram structure for the writing of rules has advantages from a practical viewpoint (readability, upgrading, success rate and transcription speed) as well as from that of the theoretical validity.

1. Introduction

Dans l'état actuel des recherches en transcription graphème-phonème, se pose le problème épineux de la gestion d'un jeu de règles de plus en plus volumineux et pourtant nécessaire pour une phonétisation de haute qualité : un transcripteur aujourd'hui doit pouvoir traiter en temps réel tout type de texte incluant bien sûr titres, dates, nombres, sigles, abréviations, noms propres et mots d'emprunt.

Après un bref état des lieux, nous souhaitons mettre en lumière ici les avantages qu'offre une structure de règles arborescente, tant d'un point de vue pratique que pour la cohérence théorique des sous-ensembles créés. Une évaluation du système est enfin proposée qui renseigne sur la vitesse d'exécution et sur le taux de transcription correcte.

2. Situation du problème

Les faibles possibilités des systèmes informatiques ont orienté la transcription graphème-phonème à ses débuts vers la conception de règles peu coûteuses en mémoire : Le Cornec [Lec72], Divay et Guyomard [Div77], Léty [Lét80], Prouts [Pro80]. Les résultats étaient déjà très acceptables car, dans la plupart des langues, quelques centaines de règles suffisent à décrire 99 % des prononciations existantes. La transcription graphème-phonème est typique du phénomène de décroissance exponentielle de l'efficacité des règles, bien connu en modélisation : peu de règles suffisent à décrire une grande majorité des cas, puis de plus en plus de règles sont nécessaires pour une amélioration de moins en moins sensible. Pour la phonétisation, il semble même qu'un doublement du nombre de règles ne suffise pas à diviser par deux le taux d'échec.

Il est donc devenu nécessaire d'améliorer la stratégie de transcription : d'un côté augmenter l'efficacité des règles, par exemple en créant davantage de catégories, de l'autre, utiliser des lexiques. Parallèlement, des efforts ont été faits pour faciliter la maintenance des règles en les rendant déclaratives (Divay [Div84]) ainsi que pour structurer les données sur des critères linguistiques (Catach [Cat84]). La plupart des systèmes actuels utilisent à la fois des règles et des lexiques, en proportions variables : Aubergé [Aub85], Bailly et Alissali [Bai92], Belrhali, Libert, Aubergé et Boë [Bel92], Belrhali [Bel95], Boula de Mareuil [Bou97].

La priorité est parfois donnée aux lexiques (Laporte [Lap88], Tihoni et Pérennou [Tih91]) et des travaux de recherche récents essaient de modéliser l'information contenue dans les lexiques pour la production automatique de règles ou pour l'apprentissage par analogie (Yvon [Yvo96]).

La variabilité a fait l'objet de travaux spécifiques (Lacheret-Dujour [Lac90]) et commence à être prise en compte dans certains systèmes.

Dans tous les cas, les règles sont structurées sur une dimension, de la plus particulière à la plus générale, même si certains interpréteurs créent automatiquement des structures arborescentes dans un but fonctionnel.

3. Choix d'une structure arborescente dès l'écriture des règles

Lorsqu'on regarde l'évolution des différents phonétiseurs, on constate que le nombre de règles n'a pas cessé de croître, et ceci malgré la percée des lexiques. Les systèmes actuellement opérationnels ont tous dépassé le millier de règles depuis une dizaine d'années. Nous souhaitons montrer ici comment les problèmes aigus de lisibilité et de maintenance posés par l'agencement de ces règles peuvent être sérieusement atténués lorsque l'on envisage une organisation arborescente de ces dernières.

3.1. Introduction d'une nouvelle règle dans une structure à une dimension

Supposons que dans l'état actuel, la règle de rang n s'applique. Un logiciel de traçage peut pointer son emplacement. La nouvelle règle (plus particulière) doit

alors être placée quelque part avant cette dernière. Le choix étant large, on recherche généralement une position pertinente ne dégradant pas la lisibilité. Exemple :

…•an•{cns}…	**✌◑▫**	**(manche, banque, grange)**
(*)…d•am•n…	**✌▫**	**(damné, condamner)**
…•a•mn…	**✌**	**(amnistie)**
…•a•mm…	**✌**	**(gamme)**
…•am•{cns}…	**✌◑▫**	**(ample, chambre)**
…•a•…	**✌**	

où : •xxx• = chaîne à transcrire
{cns} = catégorie consonne
… = contexte quelconque

Pour ajouter la règle (*) portant sur *am* dans *damné* et *condamner*, nous trouvons la règle qui s'applique : *a* suivi de *mn* → *a (amnistie)*. Dans notre exemple, la nouvelle règle (contexte *damn*) est placée juste avant.

Si l'introduction d'une nouvelle règle est fiable et relativement rapide quel que soit le nombre de règles, en revanche le choix d'une position pertinente est de plus en plus large et complexe, et par conséquent l'architecture finit par devenir confuse.

3.2. Émergence d'une structure arborescente

Revenons maintenant à notre nouvelle règle : puisqu'elle doit s'appliquer avant la règle n, c'est qu'elle constitue une exception de celle-ci. Il y a donc une connexion implicite entre ces deux règles. L'ensemble des connexions implicites entre règles n'est autre qu'un arbre. Plutôt que de concevoir un interpréteur qui crée cette structure à partir de la liste unidimensionnelle, notre approche consiste à créer nous-mêmes la structure arborescente au moment de l'écriture des règles.

Dans l'exemple que nous avons choisi, la figure 1 illustre les relations entre règles : chaque règle est un noeud de l'arbre, ses exceptions étant les noeuds qui lui sont connectés. Une telle structure optimise le parcours de l'interpréteur en éliminant tous les tests inutiles. Par conséquent elle répond parfaitement aux exigences de vitesse demandées par nos utilisateurs aveugles (réponse instantanée et lecture rapide), nous l'avons donc retenue pour l'écriture de notre phonétiseur Kali, sous la forme décrite en 3.3.

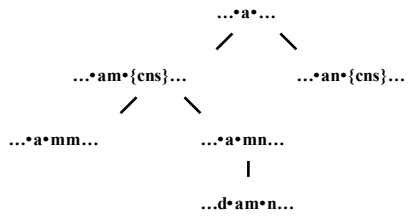

Figure 1. Arbre de dépendance des six règles choisies.

3.3. Introduction d'une nouvelle règle dans une structure arborescente

Si nous reprenons l'exemple de la nouvelle règle à insérer, la méthode est alors la suivante : nous parcourons les règles de niveau 1 (reliées à la racine) jusqu'à trouver celle dont elle dépend (règle mère), puis nous parcourons les règles de niveau 2 reliées à cette dernière, et ainsi de suite, jusqu'à un niveau où aucune règle mère n'est trouvée. Notre règle s'ajoute alors à la liste des règles de ce niveau. Exemple :

|…•a•…	**✌**	
-|…•am•{cns}…	**✌◑▫**	**(ample, chambre)**
--|…•a•mm…	**✌**	**(gamme)**
--|…•a•mn…	**✌**	**(amnistie)**
---|…d•am•n…	**✌▫**	**(damné, condamner)**
-|…•an•{cns}…	**✌◑▫**	**(manche, banque, grange)**

Pour ajouter la règle portant sur *am* dans *damné* et *condamner*, nous partons de la racine (*a* → *a*), puis nous trouvons parmi ses exceptions : *am*-consonne → ✌◑▫ *(ample, chambre)*. Cette exception comporte elle-même des exceptions (niveau 2), dont : *a* suivi de *mn* → *a (amnistie)*. Notre nouvelle règle étant incluse dans cette dernière, et aucune règle mère n'étant trouvée au niveau 3, c'est là que nous la plaçons, avec le contexte *damn*.

Les exceptions à une règle sont donc placées après celle-ci, avec une indentation supplémentaire (formalisée ici par des tirets) pour marquer leur dépendance. Cette méthode, vieille comme l'écriture, permet de structurer un texte et contient de fait une structure arborescente.

La stratégie de transcription utilisée s'applique de la même façon pour les liaisons, élisions, nombres, sigles et abréviations, qui sont intégrés dans les règles.

3.4. Avantages de la structure arborescente

Dans cette structure, chaque règle a donc sa place, ce qui permet d'éviter la perte progressive de lisibilité. En parcourant l'arbre, nous retrouvons les règles de base du Français près de la racine (*ai*, *ain*, *am*, *an*, etc.), puis dans les niveaux suivants des règles plus particulières (consonnes finales muettes ou non, terminaisons en *ent*, *h* disjonctif ou non, *i*-voyelle, *s* entre voyelles, *ti*-voyelle prononcé *si* ou *ti*, etc.), et aux derniers niveaux les règles les plus particulières.

Des règles qui auraient pu appartenir à des lexiques se trouvent alors structurées. Exemple : l'ensemble des règles sur *…•t•i{voy}…* où *t* est susceptible de se prononcer *s* se décompose en *…{voy}•t•i{voy}…*, *…c•t•i{voy}…*, *…n•t•i{voy}…*, *…p•t•i{voy}…* et *…r•t•i{voy}…*, règles qui elles-mêmes se décomposent plus finement. Nous aboutissons ainsi à une hiérarchisation fine de l'ensemble des règles de transcription.

De plus, comme un arbre s'étend sur deux dimensions, le nombre total de règles est homogène à une surface, tandis que le chemin à parcourir pour atteindre l'emplacement de la règle à insérer est homogène à une distance. Il n'est donc jamais très long.

Plus généralement, la complexité de l'arbre augmente comme la racine carrée du nombre de règles, ce qui permet de gérer sans difficulté, comme nous le verrons plus loin, un nombre de règles très supérieur à 1000.

L'une des conséquences de notre approche est une modification de l'équilibre règles/lexiques, en faveur des règles : le nombre de règles atteint à ce jour est de 4700. Les lexiques (cf. 3.5) ne contiennent actuellement que quelques centaines d'entrées.

3.5. Lexiques

Dans la pratique, les listes de mots ou groupes de mots que nous constituons à partir des erreurs rencontrées sont d'abord placées dans des lexiques, de façon à être immédiatement opérationnels. Mais ce choix n'est pas définitif, et nous les considérons plutôt comme placés en file d'attente. Peu à peu, nous les intégrons dans les règles. Cette procédure nous permet souvent de généraliser : par exemple le mot "Washington" peut être placé dans un lexique. Mais son implémentation dans les règles nous permet de voir que le contexte "ington" suffit et recouvre alors tous les mots en "ington", jusqu'ici oubliés :

---|…•ington• ♓♗♦###■ (Washington, Wellington, etc.)

L'absence du joker "…" signifie que l'extrémité du contexte correspond à l'extrémité du mot.

Le fait que nous cherchions à intégrer les lexiques ne veut pas dire que ceux-ci soient voués à une existence éphémère : nous espérons les développer plus vite que notre capacité à les transformer en règles. Le problème est le suivant : un lexique de noms propres, surtout étrangers, contient beaucoup de mots d'occurrence très rare et doit être étudié avec soin pour éliminer diverses sources d'erreurs, notamment les homographes de mots plus courants. Le rapport efficacité/coût n'est pas forcément supérieur à celui de l'intégration d'une petite liste contenant des mots nettement plus courants, comme on en obtient après un test d'évaluation.

4. Intégration du module de transcription dans le système de synthèse vocale

Bien que le transcripteur puisse fonctionner de façon autonome, ce dernier est destiné à être intégré dans le synthétiseur vocal Kali. Il bénéficie par conséquent des informations fournies par une analyse syntaxique automatique, qui, outre un étiquetage du texte à synthétiser en catégories morphosyntaxiques, propose, sur les bases de règles de dépendance (Giguet et Vergne [Gig97]), une segmentation du texte en constituants intonosyntaxiques.

L'utilisation des résultats fournis par l'analyseur syntaxique en entrée de la conversion graphème-phonème permet un traitement fin de la liaison ainsi qu'une bonne résolution des problèmes posés par la transcription des homographes hétérophones. C'est ainsi qu'une liaison *a priori* associée à la finale consonantique d'un adjectif est bloquée par l'adjonction d'un marqueur pour tout adjectif postposé (*i.e.* final de constituant intonosyntaxique) :

un ancien administrateur	liaison obligatoire
ce meuble ancien attire le regard	liaison interdite

En ce qui concerne les homographes hétérophones, si l'absence de traitement spécifique donne le bon résultat dans 80 % des cas environ (ex : *président* = nom, *tous* = pronom), l'analyse syntaxique permet d'atteindre un taux de transcription correcte de 90 % (ex : *Jean et Jacques président, ils viennent tous / ils viennent tous les jours*) ; la résolution des ambiguïtés restantes nécessitant une analyse sémantique poussée non disponible à ce jour.

La mise en œuvre de l'ensemble du synthétiseur vocal pour les essais de transcription permet également de mettre à l'épreuve certains choix qui demandent une confirmation ergonomique. Par exemple le traitement des liaisons et élisions facultatives fait l'objet de tests d'écoute auprès de nos utilisateurs aveugles : l'option choisie doit contribuer à l'esthétique du texte énoncé sans dégrader son intelligibilité.

5. Maintenance des règles

5.1. Écriture et compilation

Une fois les règles correctement positionnées, écrites et agrémentées d'exemples, un compilateur vérifie la syntaxe de ces règles avant de les condenser dans un fichier destiné à l'interpréteur et d'en extraire un fichier de test ne conservant que les exemples. Un comparateur fournit ensuite une liste différentielle d'exemples entre la nouvelle version et la version précédente. Cette méthode garantit le maintien des règles considérées comme acquises : si les nouvelles règles ont été correctement implémentées, seuls les exemples correspondant à celles-ci apparaissent, dans le cas contraire, les exemples erronés pointés par le comparateur renseignent sur les corrections à effectuer.

5.2. Mise à jour

Le fichier de règles évolue en s'enrichissant d'exemples trouvés dans les ouvrages spécialisés et de listes fournies par les utilisateurs de synthèses vocales. La constitution de ces listes à partir de la lecture de journaux et revues augmente la probabilité de traiter les règles dans le bon ordre de fréquence (donc d'importance ?), car ces publications recouvrent des domaines très divers. Le traitement des noms propres d'actualité est indispensable à la bonne lecture de ces textes. Une mise à jour périodique est nécessaire, mais peu à peu, l'ensemble se consolide.

Cette variété permet également de dresser un bilan fiable du niveau de performance, dont l'évolution pourra renseigner sur la pertinence des principes mis en place.

6. Performances actuelles

Nous distinguons deux types de performances, le taux de réussite et la vitesse. Pour être acceptable, ou mieux, pour être apprécié des utilisateurs, un système de synthèse vocale doit réunir ces deux qualités : réponse immédiate et grande majorité des phrases correctes.

6.1. Vitesse de transcription

Les temps de réponse du phonétiseur sont ramenés à ceux qui seraient obtenus sur Pentium 100 MHz, considéré comme configuration minimale.

Le résultat sans analyse syntaxique est de 100 µs par caractère en moyenne. L'analyse syntaxique, plus lente, porte le délai à environ 700 µs par caractère, soit un délai total de **70 ms** pour une phrase de **100 caractères**. Ce temps de réponse presque imperceptible est jugé satisfaisant par les utilisateurs.

6.2. Taux de bonne transcription

La dernière évaluation (avril 1998) de notre phonétiseur repose sur la transcription d'échantillons du journal "Le Monde" en date du 30 mars 1998. Seize articles couvrant les diverses rubriques ont été phonétisés (total 115657 caractères soit 21000 mots soit 77000 phonèmes). Le résultat global est de **99,6 %** de phonèmes corrects. La liste ci-dessous montre le nombre et le taux relatif d'erreurs par catégorie :

Noms propres	150	49 %
Mots d'emprunt	30	10 %
Sigles	11	3 %
Liaisons obligatoires	23	8 %
Liaisons interdites	12	4 %
Élisions obligatoires	11	3 %
Élisions interdites	2	1 %
Homographes hétérophones	32	11 %
Divers	35	11 %
Total	306	100 %

Erreurs potentielles évitées grâce à l'analyse syntaxique	39	13 %

7. Conclusion

Nous avons mis au point une stratégie de transcription et un formalisme qui nous semblent cohérents et qui permettent d'assurer une progression soutenue des performances.

La construction manuelle de la structure arborescente telle que nous la concevons aujourd'hui favorise la maintenance et l'évolution du système. Cette structure offre en outre une lisibilité optimale des règles : elle met en évidence des regroupements de règles selon des critères allant des plus généraux aux plus particuliers, tout en respectant la variété des règles et leur hiérarchie.

Afin d'optimiser notre système, un programme d'automatisation de la structure a été amorcé, comportant notamment : adjonction automatique d'une règle, traçage automatique des règles, historique sur l'utilisation de chaque règle.

Bibliographie

[Aub85] Aubergé V. (1985), "Contribution à la phonétisation automatique des langues alphabétiques : le langage TOPH", Rapport de DEA, CRISS, Université de Grenoble II.

[Bai92] Bailly G., Alissali M. (1992), "Compost : a server for multilingual text-to-speech system", Traitement du Signal, vol. 9, pp. 359-366.

[Bel92] Belrhali R., Libert L., Aubergé V., Boë L.J. (1992), "Des lexiques aux règles : vers une méthode descriptive de la phonétisation du Français", 19èmes JEP, Bruxelles, pp. 225-230.

[Bel95] Belrhali R. (1995), "Phonétisation automatique d'un lexique général du Français : systémique et émergence linguistique", Thèse de l'Université Stendhal de Grenoble 3.

[Bou97] Boula de Mareuil P. (1997), "Étude linguistique appliquée à la synthèse de la parole à partir du texte", Thèse de l'Université de Paris XI, Orsay.

[Cat84] Catach N. (1984), "La phonétisation automatique du Français", Éditions CNRS, Paris.

[Div77] Divay M., Guyomard M. (1977), "Conception et réalisation sur ordinateur d'un programme de trancription graphèmo-phonétique du Français", Thèse de troisième cycle, Université de Rennes.

[Div84] Divay M. (1984), "De l'écrit vers l'oral ou contribution à l'étude des traitements des textes écrits en vue de leur prononciation sur synthétiseur de parole", Thèse d'Etat, Université de Rennes.

[Gig97] Giguet E., Vergne J. (1997), "From part of speech tagging to memory-based deep syntactic analysis", International Workshop on Parsing Technologies 1997 Proceedings, Boston.

[Lac90] Lacheret-Dujour A. (1990), "Contribution à l'analyse de la variabilité phonologique pour le traitement automatique de la parole continue multilocuteurs", Thèse de l'Université de Paris VII.

[Lap88] Laporte É. (1988), "Méthodes algorithmiques et lexicales de phonétisation de textes, applications au Français", Thèse de troisième cycle, Université de Paris VII.

[Lec72] Le Cornec A. (1972), "La transcription orthographique-phonétique du Français". Essai de formalisation, Rapport de stage, CNET Lannion.

[Lét80] Léty M. (1980), "Transcription orthographique-phonétique : un système interpréteur", Thèse de doctorat, Université Scientifique et Médicale de Grenoble.

[Pro80] Prouts B. (1980), "Contribution à la synthèse de la parole à partir du texte, transcription graphème-phonème en temps réel sur microprocesseur", Thèse de Docteur-Ingénieur, Université de Paris Sud-Orsay.

[Tih91] Tihoni J., Pérennou G. (1991), "Phonotypical transcription through the GEPH expert system", Proceedings of the European Conference on Speech Communication and Technology (Eurospeech), pp. 767-770.

[Yvo96] Yvon F. (1996), "Prononcer par analogie : motivation, formalisation et évaluation", Thèse de l'ENST.