

FACULTÉS UNIVERSITAIRES NOTRE-DAME DE LA PAIX – NAMUR



FACULTÉ D'INFORMATIQUE

Application des Machines à Etats Finis en Synthèse de la Parole

Sélection d'unités non uniformes et Correction orthographique

Dissertation présentée par

Richard Beaufort

en vue de l'obtention du grade de

Docteur en Sciences, orientation Informatique

4 mars 2008

Composition du jury

Prof. Guy Deville, ELV, FUNDP, Namur	co-promoteur
Prof. Thierry Dutoit, TCTS, FPMs, Mons	co-promoteur
Prof. Christian Fluhr, INSTN, CEA, Saclay, France	examineur
Prof. Jean-Marie Jacquet, Faculté d'Informatique, FUNDP, Namur	promoteur
Prof. Pierre-Yves Schobbens, Faculté d'Informatique, FUNDP, Namur	examineur
Prof. Wim Vanhoof, Faculté d'Informatique, FUNDP, Namur	président du jury

Application des Machines à Etats Finis en Synthèse de la Parole

Sélection d'unités non uniformes et Correction orthographique

Richard Beaufort

Jean-Marie Jacquet Guy Deville Thierry Dutoit
promoteur *co-promoteur* *co-promoteur*

Résumé

Les machines à états finis sont des outils puissants : équivalents informatiques des langages réguliers et des relations régulières, elles s'accompagnent d'algorithmes d'optimisation et peuvent être exprimées sous la forme d'expressions régulières et de règles de réécriture. Présentes depuis l'apparition de l'informatique et abondamment employées dans les domaines les plus variés, les machines à états finis ont cependant été délaissées en traitement automatique de la langue, domaine où les chercheurs ont préféré recourir à des outils plus adaptés à la description syntaxique.

La recherche a cependant récemment réalisé un véritable bond en avant dans le domaine des machines à états finis pondérées, proposant de nouveaux algorithmes qui ont ouvert la voie à une nouvelle technologie du langage, capable de modéliser ce degré d'incertitude qui est indispensable à certains domaines du traitement automatique de la langue.

La synthèse de la parole est l'un de ces domaines. Le processus de synthèse de la parole, qui consiste à produire de la parole à partir du texte, regorge de difficultés, empreintes d'un certain degré d'incertitude, qui peuvent utilement profiter du potentiel expressif des outils à états finis dans leur ensemble.

Dans le cadre de cette thèse, deux de ces difficultés ont particulièrement retenu notre attention : l'étape de sélection d'unités non uniformes qui précède la génération de la parole, et la correction orthographique réalisée dans le contexte de l'analyse morpho-syntaxique.

Les travaux réalisés s'articulent autour de trois objectifs distincts. D'une part, nous avons désiré prouver que les machines à états finis facilitent la conception de tâches complexes sous la forme d'une succession de tâches simples. D'autre part, nous avons voulu proposer des approches nouvelles et originales en sélection d'unités non uniformes et en correction orthographique. Enfin, nous avons tâché de contribuer au domaine des machines à états finis, en définissant de nouvelles méthodes d'implémentation et de nouveaux algorithmes qui ont été intégrés dans nos propres outils : une bibliothèque de machines à états finis, et un compilateur de dictionnaires, de langages et de règles de réécriture.

Table des matières

Résumé	iii
Table des matières	xiv
Table des figures	xv
Liste des tableaux	xix
Liste des algorithmes	xxii
Liste des acronymes	xxiii
Remerciements	xxv
Dédicace	xxix
Introduction	3
1 Du contexte aux objectifs	3
2 Plan de la thèse	5
3 Postulats et hypothèses	6
3.1 Diviser pour mieux régner	6
3.2 Externalisation des données	7
4 Contributions	9
4.1 Machines à états finis	9
4.2 Sélection d'unités non uniformes	11
4.3 Correction orthographique	11
I Machines à états finis	13
1 Introduction	15
1.1 Plan de la partie	15
1.2 Notions fondamentales	16
1.3 Langages formels et hiérarchie de Chomsky	17

1.3.1	Langages formels, grammaires formelles	17
1.3.2	Hiérarchie de Chomsky	18
1.3.3	Langages naturels, langages non-contextuels ?	21
2	Les automates	23
2.1	Introduction	23
2.2	Définitions	23
2.2.1	Graphes	23
2.2.2	Langages	24
2.3	Niveaux de représentation des automates	25
2.3.1	Automate déterministe	25
2.3.2	Automate non-déterministe	26
2.3.3	Automate non-déterministe avec transitions ϵ	28
2.4	Equivalence et minimisation	30
2.4.1	Tester l'équivalence des états	30
2.4.2	De l'équivalence à la minimisation	32
2.4.3	Tester l'équivalence de DFAs	35
2.5	Opérations régulières	40
2.5.1	Théorème de Kleene	40
2.5.2	Autres propriétés de clôture	41
2.6	Synthèse	42
3	Les transducteurs	45
3.1	Introduction	45
3.2	Définitions	45
3.2.1	Graphes	45
3.2.2	Relations	47
3.2.3	Transductions	48
3.3	Opérations régulières	49
3.3.1	Clôture sous l'union	49
3.3.2	Clôture sous l'inverse	49
3.3.3	Clôture sous la composition	49
3.3.4	Clôture sous l'intersection	53
3.4	Transducteurs séquentiels	53
3.4.1	Définition	53
3.4.2	Séquentialisation d'un transducteur	55
3.4.3	Minimisation d'un transducteur séquentiel	58
3.5	Synthèse	64
4	Les machines pondérées	67
4.1	Introduction	67
4.2	Fondements mathématiques	69
4.2.1	Semi-anneau	69
4.2.2	Séries entières rationnelles	70
4.3	Machines à états finis pondérées	71

4.4	Problème de la distance la plus courte	73
4.5	Recherche des n meilleurs chemins d'un graphe	74
4.5.1	Première étape	74
4.5.2	Seconde étape	78
4.6	Optimisation des machines pondérées	81
4.6.1	Suppression- ϵ pondérée	81
4.6.2	Déterminisation pondérée	84
4.6.3	Minimisation pondérée	86
4.7	Espérance-Maximisation	96
4.8	Synthèse	97
5	Expressions régulières et règles de réécriture	99
5.1	Introduction	99
5.2	Les expressions régulières	99
5.2.1	Définition	99
5.2.2	De l'expression régulière à l'automate	101
5.3	Les règles de réécriture	106
5.3.1	Introduction	106
5.3.2	Types de règles : variations sur un même thème	107
5.3.3	Algorithmes	109
5.4	Synthèse	113
6	Les outils développés	115
6.1	La bibliothèque de Machines à Etats Finis	115
6.1.1	Pourquoi une nouvelle bibliothèque ?	115
6.1.2	Principes d'implémentation	116
6.1.3	Choix du langage et application des principes d'implémentation	117
6.1.4	Représentation générale des machines	121
6.1.5	Format binaire	124
6.1.6	Principales méthodes disponibles sur les FSMs	126
6.1.7	Principales méthodes disponibles sur les alphabets	128
6.1.8	Extensions	129
6.1.9	Synthèse	137
6.2	Le compilateur Ovide	137
6.2.1	Principes	138
6.2.2	Facilités	139
6.2.3	Sections et compilation	142
6.2.4	Les marqueurs : extension des règles de réécriture	143
6.2.5	Synthèse	147
7	Conclusion	149

II Synthèse par sélection d'unités non uniformes	151
8 Introduction	153
8.1 Plan de la partie	153
8.2 Le signal acoustique de parole	154
8.2.1 Le spectre	156
8.2.2 La fréquence fondamentale	156
8.2.3 Les formants	156
8.2.4 L'énergie	157
8.2.5 La durée	157
8.2.6 Quelques méthodes d'analyse et de représentation	158
8.3 Représentation symbolique non ambiguë	159
8.3.1 Le phonème	159
8.3.2 La prosodie	166
8.4 Evolution du concept de synthèse	169
8.4.1 Vous avez dit <i>articulatoire</i> ?	170
8.4.2 Encapsulation dans des unités.	171
8.5 Les principes de la synthèse par sélection	174
8.5.1 Le corpus de parole	175
8.5.2 Le processus de sélection	179
8.5.3 Analyse	181
8.6 Conclusion	182
9 Etat de l'art en synthèse par sélection	185
9.1 Critères de sélection	185
9.1.1 Les bases	186
9.1.2 Sélection acoustique	187
9.1.3 Sélection linguistique	190
9.2 Optimisation de la recherche	194
9.2.1 Optimisation de la pré-sélection	194
9.2.2 Optimisation des coûts de concaténation	198
9.2.3 Optimisation du processus global	199
9.3 Limites des systèmes de l'état de l'art	202
10 LiONS	205
10.1 Preuve de concept	205
10.1.1 Postulat et hypothèses	206
10.1.2 Le synthétiseur	208
10.1.3 Choix des critères	210
10.1.4 Corpus et entraînement	216
10.1.5 Pondération des critères	220
10.1.6 Processus de sélection et synthèse	224
10.1.7 Evaluation et analyse	228
10.2 Révisions et optimisations	230
10.2.1 Hypothèses	231

10.2.2	Révision des critères de pré-sélection	231
10.2.3	Révision des critères du coût de concaténation	235
10.2.4	Modèle d'optimisation	237
10.2.5	Entraînement adapté au modèle	239
10.2.6	Processus de sélection et synthèse	250
10.2.7	Analyse	252
11	Conclusion	261
11.1	Principe	261
11.2	Etat de l'art	261
11.3	Méthode proposée	262
11.3.1	Première version	262
11.3.2	Seconde version	263
11.4	Bien-fondé du modèle	264
11.4.1	Sur les principes de sélection	265
11.4.2	Sur les principes d'optimisation	265
11.4.3	Sur le caractère multilingue	266
11.5	Les apports des machines à états finis	266
11.6	Une synthèse cependant aléatoire...	267
III	Correction orthographique	269
12	Introduction	271
12.1	De l'importance de l'analyse linguistique en synthèse	271
12.2	Positionnement du problème	273
12.3	Plan de la partie	274
13	Présentation de l'analyse linguistique	275
13.1	Définitions	276
13.2	Choix d'implémentation d'eLite	278
13.2.1	Structure de données et unité linguistique	278
13.2.2	Données et langues	280
13.2.3	Rôle dans le développement d'eLite	280
13.3	Pré-processeur	282
13.4	Analyseur morphologique	283
13.4.1	Description de l'analyse flexionnelle	283
13.4.2	Traitement des tokens lexicaux	290
13.4.3	Traitement des tokens URI	295
13.4.4	Traitement des autres tokens	295
13.5	Analyseur syntaxique	296
13.5.1	Introduction	296
13.5.2	Etat de l'art en analyse syntaxique	297
13.5.3	Analyse syntaxique dans eLite	302
13.6	Le point sur l'analyse linguistique présentée	312

14 Etat de l'art en correction	315
14.1 Typologie des erreurs	315
14.1.1 Types d'erreurs	315
14.1.2 Causes des erreurs	316
14.2 Niveaux de complexité en correction orthographique	319
14.2.1 Mots isolés <i>vs</i> Mots en contexte	319
14.2.2 Détection <i>vs</i> Correction	319
14.2.3 Correction interactive <i>vs</i> Correction automatique	319
14.2.4 Synthèse	320
14.3 Détection des OOVs	320
14.3.1 Les dictionnaires	320
14.3.2 Les modèles <i>n</i> -gramme	322
14.3.3 Le problème des frontières de mots	323
14.3.4 Analyse	323
14.4 Correction des OOVs	323
14.4.1 Distance d'édition	324
14.4.2 Clefs de similarité	331
14.4.3 Systèmes par règles	333
14.4.4 <i>n</i> -grammes existentiels	334
14.4.5 <i>n</i> -grammes statistiques	334
14.4.6 Réseaux de neurones	335
14.4.7 Evaluation des approches présentées	337
14.5 Correction des erreurs sur IVs	338
14.5.1 Construction de règles linguistiques	340
14.5.2 Modèle de langue à orientation lexicale	342
14.5.3 Listes de confusion	343
14.5.4 Extraction de caractéristiques multi-niveaux	354
14.5.5 Exploitation du Web	357
14.5.6 Synthèse des approches présentées	358
14.6 Conclusion	359
14.6.1 Typologie des erreurs	359
14.6.2 Machines à états finis	360
15 Intégration de la correction orthographique en synthèse	363
15.1 Postulats	363
15.1.1 Typologie des erreurs	364
15.1.2 Architecture de l'analyse	364
15.2 Des postulats à l'algorithme	364
15.2.1 Au niveau de la typologie des erreurs	364
15.2.2 Au niveau de l'architecture de l'analyse	365
15.2.3 Au niveau du modèle de correction	366
15.3 L'algorithme	366
15.3.1 Algorithme complet	366
15.3.2 Algorithme d'analyse des tokens lexicaux	368
15.3.3 Analyse d'une URI	370

15.3.4	Analyse des autres tokens	372
15.4	Lignes de façade de l'algorithme	372
15.4.1	Interface de communication	372
15.4.2	Les méthodes	375
15.5	Les modèles de l'analyse morpho-syntaxique	385
15.5.1	Modèle de langue	385
15.5.2	Gestion de la casse	391
15.5.3	Analyse morphologique	392
15.5.4	Génération des unités linguistiques	399
15.5.5	Les points-clefs de l'analyse morpho-syntaxique	401
15.6	Les modèles de la correction orthographique	402
15.6.1	Correction des OOVs	402
15.6.2	Correction flexionnelle	412
15.6.3	Les points-clefs des modèles de correction	419
15.7	Evaluation	419
15.7.1	Remarques préliminaires	420
15.7.2	Evaluation de l'analyse morpho-syntaxique	421
15.7.3	Evaluation de la correction des OOVs	428
15.7.4	Evaluation de la correction flexionnelle	443
15.8	Avant de conclure	449
16	Correction en scènes naturelles	451
16.1	Introduction	451
16.2	Erreurs et postulat	452
16.3	Survol de la correction en scènes naturelles	453
16.4	Le système de correction proposé	454
16.4.1	Le filtre de composition	455
16.4.2	Le dictionnaire	459
16.4.3	L'algorithme	459
16.5	Evaluation	463
16.6	Synthèse	465
16.6.1	Une approche efficace	465
16.6.2	Des limites à dépasser	467
17	Conclusion	469
17.1	Positionnement du problème	469
17.2	Objectifs	469
17.3	Etat de l'art	470
17.3.1	Correction des OOVs	470
17.3.2	Correction des IVs	470
17.4	Correction des textes entrés au clavier	471
17.4.1	Nouvelle analyse morpho-syntaxique	472
17.4.2	La correction	473
17.5	Correction des textes extraits de scènes naturelles	474
17.6	Les apports des machines à états finis	475

17.6.1	Au niveau de l'analyse morpho-syntaxique	475
17.6.2	Au niveau des modèles de correction	475
17.7	Des hypothèses confirmées	477
17.8	Une hypothèse non confirmée...	478
17.9	Un accueil encourageant et révélateur	478
Conclusion		481
1	Du contexte aux objectifs	481
2	Quelques contributions aux machines à états finis	482
2.1	Extension des modes de représentation	482
2.2	Extension des règles de réécriture	483
3	Les originalités des approches proposées	483
3.1	Sélection d'unités non uniformes	483
3.2	Correction orthographique	485
4	Les apports des machines à états finis	488
5	Perspectives	489
5.1	Analyse syntaxique	490
5.2	Gestion des mots hors-vocabulaire	508
Bibliographie		517
Annexes		541
A Bibliothèques et outils à états finis		541
1	Les bibliothèques	542
1.1	Automates classiques	542
1.2	Machines classiques	543
1.3	Machines classiques et pondérées	544
2	Les outils	546
B Ovide : documentation		549
1	A propos des expressions régulières	550
1.1	Quelques définitions	550
1.2	Opérateurs définis dans Ovide	551
1.3	Précédence des opérateurs dans Ovide	551
1.4	Strings prédéfinies dans Ovide	552
1.5	Exemples	552
2	A propos des règles de réécriture	553
2.1	Règles classiques	553
2.2	Règles pondérées	554
2.3	Règles optionnelles	554
2.4	La string vide	554

3	Les sections d'un fichier	555
3.1	Informations générales	556
3.2	Les classes d'entrée	560
3.3	Les classes de sortie	562
3.4	Les langages d'entrée	563
3.5	Les règles de réécriture	563
3.6	Les langages de sortie	564
3.7	L'inclusion	565
3.8	La compilation particulière	567
4	Quelques opérateurs particuliers	568
4.1	L'opérateur <i>strict</i>	568
4.2	L'opérateur <i>complément</i>	569
4.3	L'opérateur <i>composition</i>	569
4.4	L'opérateur <i>projection</i>	570
5	Ligne de commande	571
5.1	Comportement standard	571
5.2	Options	572
6	Exemples	574
6.1	Un exemple simple	574
6.2	Avec un fichier inclus	575
6.3	Avec un filtre	577
C Sélection d'unités non uniformes		579
1	Les groupes rythmiques	579
2	Exemple de table d'étiquetage	580
3	LiONS 1 : pondération moyenne des critères du coût cible	584
D Correction orthographique		585
1	Catégories syntaxiques	585
1.1	Catégories valables pour les unités linguistiques et les natures	585
1.2	Catégories valables exclusivement pour les unités linguistiques	586
1.3	Réflexions qui ont mené à la constitution de cette liste de catégories	587
2	Composés à traits d'union	589
2.1	2 parties	589
2.2	3 parties	589
3	Traits grammaticaux et classes flexionnelles	590
3.1	Traits grammaticaux	590
3.2	Classes flexionnelles	590
4	Pseudocodes de l'analyse morpho-syntaxique	592
5	Segmentation dans le cas de la dichotomie par alignement	600
E Postulats et hypothèses		601
1	Démarche globale	601
2	Sélection d'unités non uniformes	602
2.1	Postulats linguistiques	602

2.2	Principes de sélection	602
2.3	Principes d'optimisation	603
3	Correction orthographique	603
3.1	Architecture du système	603
3.2	Analyse morphologique	604
3.3	Correction	604