LABORATOIRE D'INFORMATIQUE DE NANTES ATLANTIQUE

Reconnaissance de Mini-Messages Manuscrits

Emmanuel Prochasson

encadré par Emmanuel Morin et Christian Viard-Gaudin

Laboratoire d'Informatique de Nantes Atlantique
2, rue de la Houssinière – B.P. 92 208 – 44 322 NANTES CEDEX 3

Institut de Recherche en Communications et en Cybernétique de Nantes
1, rue de la Noë – BP 92 101 – 44 321 Nantes CEDEX 3

RAPPORT DE STAGE DE MASTER RECHERCHE

Août 2006

LINA, Faculté des Sciences & Techniques de Nantes
2, rue de la Houssinière
B.P. 92208 — F-44322 NANTES CEDEX 3

Emmanuel Prochasson (encadré par Emmanuel Morin et Christian Viard-Gaudin)

Reconnaissance de Mini-Messages Manuscrits

Reconnaissance de Mini-Messages Manuscrits

Emmanuel Prochasson (encadré par Emmanuel Morin et Christian Viard-Gaudin)

emmanuel.prochasson@etu.univ-nantes.fr

Remerciements

Je tiens à remercier mes encadrants : Christian Viard-Gaudin et Emmanuel Morin pour leurs conseils avisés, mais aussi pour avoir supporté mes fréquentes irruptions ainsi que mes avalanches de courriers électroniques. Je remercie également Freddy Perraud pour son attention et pour m'avoir fait partager son expérience.

Je remercie les locataires du bureau 212 pour avoir accepté de tolérer une présence masculine dans cet espace préservé (et puis pour avoir rigolé à mes blagues – parfois).

Je remercie Béatrice Daille pour sa sollicitude et l'ensemble de l'équipe TALN pour son accueil et sa convivialité.

Je remercie tous les employeurs qui ont accepté de m'exploiter cette année en échange de quelques euros (en particulier le groupe La Poste, qui m'a sponsorisé et Brinks, même s'ils ont récupéré mon uniforme).

Je remercie enfin mes amis (sauf deux qui se reconnaîtront), mes collègues jeunes chercheurs (sauf ceux qui ont essayé de me tuer – en particulier ceux qui ont réussi), mes parents et Anne-Céline, pour m'avoir supporté.

Table des matières

Chapitre 1

Introduction

L'écriture manuscrite est toujours omniprésente malgré l'apparition des supports numériques. Elle reste un médium privilégié par chacun et est indispensable pour la plupart des documents administratifs. Son traitement automatique est donc un enjeu très important mais reste à ce jour surtout répandu dans des domaines très spécialisés tels que la reconnaissance d'adresse pour le tri postal ou la lecture de chèque bancaire.

En effet, plus l'écriture est contrainte et prévisible et plus la reconnaissance est efficace. Dans le cadre du tri postal, les enveloppes pré-casées sont une première contrainte d'écriture ; le système sait de plus qu'il doit reconnaître un code postal, c'est-à-dire une suite de 5 chiffres, ce qui lui permet d'ignorer toutes solutions contenant autre chose. De façon plus poussée, le tri peut aller jusqu'à la préparation de la tournée des facteurs, c'est-à-dire que le processus de tri regroupe le courrier pour quelques rues d'une ville.

Le courrier, d'abord trié par quartier avec l'information du code postal, est trié par groupe de rues. Le système, ayant connaissance du quartier concerné, sait quelles voies il est susceptible de devoir reconnaître sur chaque enveloppe (voir fig. 1.1). Dans ce cas, le système n'a pas à « deviner » ce qui a été écrit, mais plutôt à faire le choix du résultat le plus probable parmi ses connaissances. Une lettre introduite dans la machine par erreur, c'est-à-dire ne correspondant pas au programme de la machine, sera donc classée comme « inconnue ». Ce sont ces informations, très précises et ingénieusement classées, qui autorisent une reconnaissance de l'écriture ultra-rapide et très fiable, pour un tri mécanique atteignant 70 000 lettres par heure et par machine de tri !

À l'inverse, une écriture non contrainte comme l'écriture cursive, ou une écriture non prévisible comme un texte libre sont très difficiles à reconnaître, sauf à s'aider d'artifices tels qu'un alphabet modifié comme c'est le cas sur les agendas électroniques. Il est nécessaire de réduire l'ensemble des solutions de la reconnaissance pour améliorer ses performances. *La reconnaissance de l'écriture manuscrite n'est donc efficace que si elle est assistée de connaissances sur les données à reconnaître.* Ceci peut se rapprocher du travail de reconnaissance « humain » : nous sommes tout à fait capables de lire des mots mal écrits en s'aidant des mots voisins, ou de comprendre une phrase alors que nous ne l'avons entendue qu'à moitié.

Le problème que nous nous proposons d'étudier ici porte sur la reconnaissance de MIni MEssages MAnuscrits (MIMEMA). Il nous faut apporter au système de reconnaissance utilisé les informations sur la forme des MIMEMA nécessaires à une reconnaissance efficace.

Nous verrons dans les deux premières parties que les difficultés sont nombreuses de

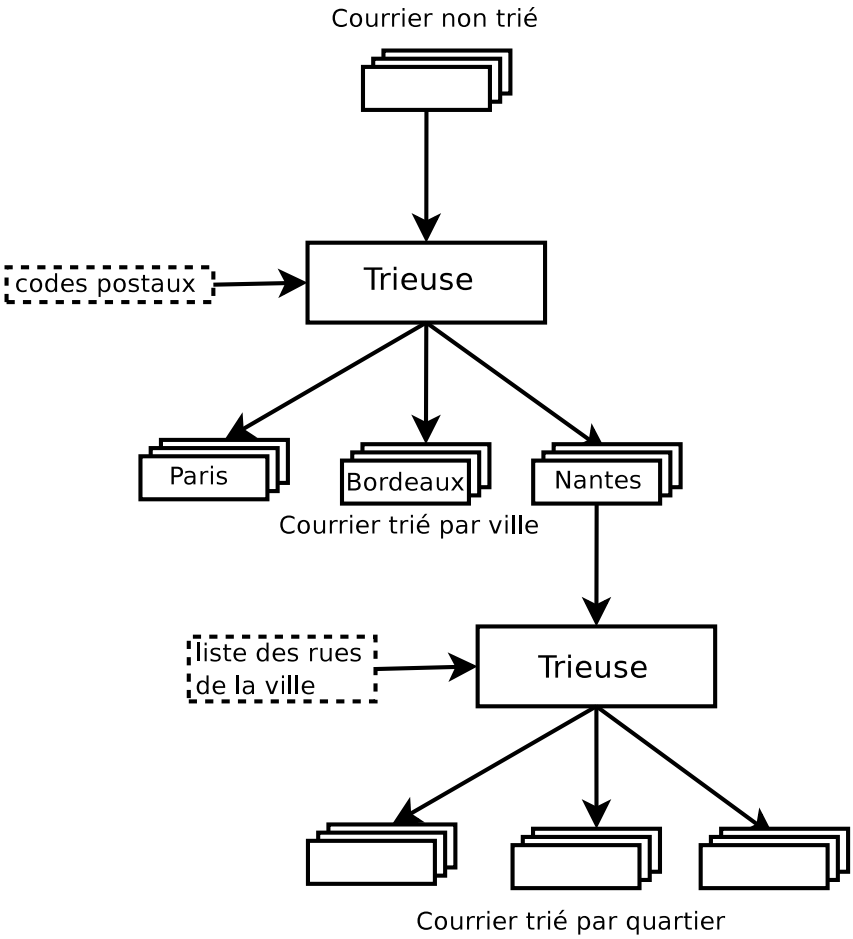

FIG. 1.1 – L'importance du contexte pour la reconnaissance de l'écriture manuscrite dans le cas du tri postal

par la nature même du langage utilisé dans les MIMEMA et des contraintes de la reconnaissance de l'écriture. Nous présenterons ensuite différentes approches du problème conduisant à une réduction du nombre d'erreurs puis nous chercherons à améliorer le taux de reconnaissance en classant notre jeu de test en fonction de la forme des mots et de leurs caractéristiques.

Cette étude s'inscrit dans le cadre du projet ATLANstic, pôle nantais en STIC regroupant les laboratoires IRCCyN[1], LINA[2] et IREENA[3]. Le projet MIMEMA est un projet labelisé entre l'équipe Traitement Automatique du Langage Naturel du LINA et l'équipe Image Vidéo Communication de l'IRCCyN.

[1]Institut de Recherche en Communications et en Cybernétique de Nantes
[2]Laboratoire d'Informatique de Nantes Atlantique
[3]Institut de Recherche en Électrotechnique et Électronique de Nantes Atlantique

Chapitre 2

À propos des Nouvelles Formes de Communication Écrites

La linguistique ne peut les ignorer plus longtemps : les Nouvelles Formes de Communication Écrite (NFCE) sont omniprésentes. Alors qu'il y a trente ans le courrier électronique était réservé à une partie de la communauté scientifique, il s'est aujourd'hui imposé dans toutes les entreprises et dans la plupart des foyers français [Ani01]. À leur tour les SMS sont massivement utilisés, en particulier par les plus jeunes qui favorisent ce médium rapide et discret. Les « chats » et les forums de discussions (et par extension les « weblogs » – voir 2.1.1) ont également connu une forte expansion grâce à la popularisation d'Internet.

2.1 SMS, Chat et forum de discussion : les NFCE

Les NFCE ont apporté avec elles leurs lots de casse-tête pour les linguistes. Nous proposons à la suite une synthèse des travaux effectués à ce jour, d'un point de vue linguistique, mais aussi sociologique.

2.1.1 Caractéristiques des NFCE

Les NFCE se caractérisent par une prise de liberté par rapport aux canons de l'écrit classique [ÉGV06]. Les phrases ci-dessous sont extraites de *skyblog*, l'hébergement de *weblog* de la station de radio *skyrock*[1].

> *slt tu vien de fere un tour sur mon blog alor tou dabor jte di merci épui ta pub sera biento ak lé otr patience... ^^*
>
> Salut, tu viens de faire un tour sur mon blog. Alors tout d'abord je te dis "merci" et puis ta pub sera bientôt avec les autres, patience...

[1] Les *skyblogs* sont un véritable phénomène puisqu'en mai 2006, on comptait 4,8 millions de blogs actifs, 244 millions d'articles et 415 millions de commentaires. En mars 2003, on estimait déjà qu'un adolescent sur deux possédait un skyblog, ce qui a conduit à une série de problèmes scolaires dues aux diffamations fréquentes de professeurs et de camarades. Voir à ce sujet deux articles du journal *Libération* disponible en ligne : *Ferme ton blog d'abord* `http://www.liberation.fr/page.php?Article=284309` et *Les blogs lycéens dans la mire des chefs d'établissement* `http://www.liberation.fr/page.php?Article=284428`

wesh bi1 ou coi sérieux ton blog il est balaiz lol

Wesh, bien ou quoi ? Sérieux, ton blog il est balaise

Les exemples choisis révèlent dans un premier temps plusieurs phénomènes communs aux différentes NFCE, à savoir :

- une orthographe inventive ;
- l'usage d'abréviations ;
- un respect approximatif des conventions typographiques et à plus forte raison de la ponctuation ;
- l'usage de « *didascalies électroniques*[2] » pour transmettre une émotion directe au delà du sens du texte ;
- l'usage de néologismes et de néographies [ÉGV06] ;

2.1.2 Orthographe et typographie

Une des premières caractéristiques des NFCE est le respect approximatif de l'orthographe et des conventions typographiques, pour plusieurs raisons. Il faut accuser en premier lieu le périphérique de saisie (généralement un clavier d'ordinateur ou pire, de téléphone portable) qui engendre régulièrement quelques erreurs par exemple des inversions :

`otrhographe`

des substitutions :

`orthogrzphe`, le "z" et le "a" étant proches sur le clavier

des redoublements :

`orthoggraphe`

voir des omissions :

`orthgraphe`

Ces phénomènes ne se retrouvent pas dans l'écrit manuscrit sauf chez les personnes atteintes de certains troubles comme la dyslexie. La seconde raison de cette déviance vient du caractère beaucoup moins formel des NFCE par rapport à d'autres formes. [Ani01] montre que le courrier électronique, même s'il est adressé à quelqu'un d'important (un supérieur hiérarchique par exemple) est généralement plus relâché qu'un courrier papier traitant du même sujet. Pour cette raison les fautes introduites par le clavier ne seront pas nécessairement corrigées.

[2] « *lol* » signifiant « *Laughing Out Loud* » l'équivalent anglophone de « *mort de rire* » et « ^^ » devant se voir comme deux yeux plissés évoquant un visage souriant à la façon des *manga*.

2.1.3 Néologisme et néographie

L'utilisation de nouveaux mots est un phénomène très fréquent en particulier dans les SMS ou l'IRC[3] où « *le plaisir de parler l'emporte parfois sur le propos lui-même* »[MDRRT04]

À titre d'exemple citons le mot *chat* que nous avons déjà utilisé, qui vient du verbe anglais *to chat*, *bavarder*. Ces néologismes sont généralement des francisations brutales de mots anglophones (l'Académie française ayant sans doute pris un peu de retard dans la francisation des mots issus des TIC[4]).

Les **néographies** sont également très fréquentes : des mots existants sont écrits de manière créative. Les « fautes » sont alors volontairement introduites par les utilisateurs. [Pié03] parle de *ludogénèse* et de phénomène de société. Les mots sont construits de façons ludiques par une communauté d'utilisateurs qui se rassemblent autour de ces « codes d'écritures ». L'auteur fait le parallèle entre la « *cyberl@ngue* » [Aur02] et certaines langues communautaires comme les créoles qui sont généralement des spécialisations d'une langue plus commune, parlée par un groupe de personnes.

La littérature ([Ani02] et [ÉGV06]) distingue différentes néographies, citons les plus fréquentes, en nous concentrant sur celles utilisées dans les SMS :

- la **phonétisation** : `lé`, `otr`, `coi`, `fere` ;
- les **squelettes consonantiques** : `bcp`, `slt` ;
- les **rébus** : `2m1`, `bi1` ;
- les **étirements graphiques** : `supeeeeeeer` ;
- les **agglutinations** : `eske`, `jte`, `épui` ;

Ces structures productives peuvent évidemment être combinées comme dans « *1dpdte* ».

2.1.4 Didascalies électroniques

Ce terme a été introduit dans [MDC99] et évoque les didascalies du théâtre. Ces expressions servent à relater les émotions ou le ton du message. Les plus connues sont évidemment les *smileys*[5], descendant de l'« ASCII Art[6] », qui doivent se lire en tournant la tête.

- $:-)$ sourire ;
- $;-)$ clin d'oeil ;
- $:-($ mécontentement ;

Il en existe beaucoup d'autre et il n'est pas interdit d'en inventer de nouvelles au fil des besoins tels que $8-\{)$, qui représente un personnage moustachu à lunette... Citons également quelques sigles tels que *lol* (ou sa version française *mdr*[7]), *pt2r*[8] ou *rofl*[9]. Ces informations servent à pallier l'insuffisance de la ponctuation classique pour évoquer l'ironie, le mécontentement ou la connivence.

[3] « *Internet Relay Chat* », le protocole des « Chats ». Un glossaire des abréviations employées est disponible en annexe A.

[4] *Technologies de l'Information et de la Communication*

[5] aussi appelés *binettes* ou *émoticônes*

[6] Représentations graphiques à l'aide de caractères uniquement, utile pour les terminaux non graphiques

[7] « Mort De Rire »

[8] « Pété de rire » – notons l'usage du rébus au sein du sigle

[9] « Rolling On The Floor Laughing »

2.1.5 Idées reçues

[ÉGV04] et [ÉGV06] sont deux documents très complets traitant du phénomène des NFCE du point de vue du traitement du langage. Les auteurs y font tomber quelques idées reçues qui courent sur le sujet. Nous résumons ici ces travaux.

Écrit oralisé Ils montrent que les NFCE ne sont pas de l'écrit oralisé (c'est-à-dire une transcription écrite de texte oral). D'après eux, il est perçu ainsi uniquement parce qu'il est nouveau, original, donc fautif, et que, puisque le parlé est le plus souvent fautif alors les NFCE sont de l'oral écrit.

> *Il est certain que l'oral est souvent informel, et l'écrit formel [...]. Lorsque l'oral devient formel, on y retrouve les tournures caractéristiques de l'écrit [...] et à l'inverse l'écrit informel utilise des tournures fréquentes à l'oral.*

Les auteurs montrent par la suite que le vocabulaire des NFCE n'est pas propre à l'oral, pas plus qu'à l'écrit d'ailleurs. Nous n'avons donc pas à faire à un langage *particulièrement oral*, du moins, pas plus que dans le cas de beaucoup d'autres formes de communication. Il est donc plus question ici de *registre* que de *support de communication*.

Agrammaticalité Les auteurs affirment que les NFCE ne sont pas agrammaticales, pas plus que le français oralisé ou même relativement formel qui, par exemple, omet souvent « *ne* » dans les constructions négatives.

En réalité, l'apparente agrammaticalité vient plutôt du non respect de l'orthographe. Des textes issus des NFCE *traduits* sont tout à fait corrects d'un point de vue grammatical. Reprenons un des exemples du début :

> *slt tu vien de fere un tour sur mon blog alor tou dabor jte di merci épui ta pub sera biento ak lé otr patience... ^^*

```
Salut, tu viens de faire un tour sur mon blog, alors tout d'abord je
te dis merci et puis ta pub sera bientôt avec les autres, patience...
```

La forme traduite ci avant est, d'un point de vue grammatical, tout à fait anodine.

On trouve toutefois des fautes de *sens*, tel que l'inversion des verbe *être* et *savoir*, comme discuté en 2.2.1, mais ce ne sont pas des fautes de grammaire.

Dictionnaire texto Il semblerait qu'il ne soit pas possible de construire un lexique des textos[10]. En effet, les néographies étant productives, il n'est pas aussi simple que cela de les énumérer ; une rapide analyse d'un corpus de SMS ne fait ressortir que quelques rares mots fréquents sans pouvoir vraiment les considérer comme suffisamment discriminant pour le traitement du langage.

[10] « texto » est le nom choisi par la société de téléphonie mobile *SFR* pour les SMS

2.2 Traitements Automatiques du Langage et NFCE

Les NFCE sont un nouveau casse-tête pour le TALN, plus habitué à traiter des langages « standardisés » et même généralement canoniques. Les registres des textes étudiés sont souvent très spécialisés, par exemple dans le domaine scientifique ou juridique. Pourtant la normalisation du français est relativement récente et [ÉGV06] s'amuse à penser que si l'informatique était apparue au moyen-âge, le TALN aurait été confronté aux mêmes problèmes que ceux qu'il rencontre aujourd'hui avec des langages complètement libres comme ceux des NFCE.

Pour traiter efficacement les NFCE, il convient d'abord de ne pas les mélanger : le langage des SMS diffère du langage de l'IRC, ou de celui des forums de discussion, des blogs... Les vecteurs de communication sont différents : d'un point de vue physique d'abord, avec des claviers plus ou moins ergonomiques, ce qui modifie la forme du message, mais aussi en ce qui concerne la temporalité de la communication, ce qui modifie l'information du message : les échanges sur IRC sont en « temps réel » (chacun voit apparaître instantanément ce que disent les autres) alors que les SMS sont à *temporalité différée* : il y a un temps de latence entre l'émission du SMS, sa réception et sa lecture, d'autant plus si le destinataire n'est pas joignable. On aura donc plus tendance à écrire un message court et informatif par SMS, alors qu'il est fréquent de « parler pour ne rien dire » sur IRC. La table 2.2, extraite de [Ani02] résume les différences entre les différents médium.

	Lecteur	Temporalité	Écriture	Lecture
Courrier Électronique	Individu	Différé	Clavier ordi.	Écran ordi.
Messagerie Instantanée	Individu	Immédiat	Clavier ordi.	Écran ordi.
IRC	Groupe	Immédiat	Clavier ordi.	Écran ordi.
SMS	Individu	Différé	Clavier tel.	Écran tel.

TAB. 2.1 – Comparaison de différents médiums selon [Ani02]

De même, tous les médiums ne partagent pas les mêmes déviances. Les SMS, du fait du temps passé à écrire chaque lettre, sont beaucoup moins sujets aux fautes de typographie que le courrier électronique. À l'inverse, l'IRC où il faut frapper vite sur le clavier pour une discussion réactive est un fort vecteur des fautes de typographie présentées en 2.1.2. Les erreurs propres à chaque médium sont résumées dans la table 2.2.

	Typographie	Néologisme	Néographie	Didascalies
Courrier électronique	Fréquent	Rare	Rare	Fréquent
IRC	Très fréquent	Très fréquent	Très fréquent	Très fréquent
SMS	Rare	Très fréquent	Très fréquent	Fréquent

TAB. 2.2 – Les déviances de chaque médium

Au sein d'un même médium il y a aussi de lourdes différences, souvent dues aux différents usagers du médium (âge, niveau social, niveau d'étude...). Ainsi, sans rentrer dans une étude statistique, les discussions sur les salons IRC de Wikipedia, regroupant beaucoup de jeunes chercheurs ayant plus de 20 ans, avec de forts niveaux d'études, sont beau-

coup moins déviées (d'un point du vue linguistique) que les discussions sur les salons de *jeuxvideo.com* qui rassemblent des adolescents.

D'après [ÉGV04], il faudra sans doute oublier une partie des travaux du TALN pour parvenir à l'adapter au NFCE, en effet les modèles actuels semblent trop inflexibles pour s'adapter à ces formes libres. Cette flexibilité devra se retrouver dans les lexiques mais également dans les traitements spécifiques à chaque forme, formes qu'il faudra pouvoir détecter efficacement.

2.2.1 L'influence des NFCE sur l'orthographe

Beaucoup s'inquiètent de l'influence des déviances des NFCE sur l'orthographe, en particulier sur les jeunes alors qu'à l'inverse, certains y voient le retour de l'écrit grâce aux nouvelles technologies. Le « langage SMS », au delà du phénomène de mode, agace, surtout lorsqu'il est employé à tort et à travers. À titre d'anecdote citons le *Comité de lutte contre le langage SMS et les fautes volontaires sur Internet*[11] qui s'étend sur de nombreux forums de discussion à travers une série de bannières, invitant les intervenants à écrire dans un français « lisible ». Ici toutefois l'objet n'est pas directement le respect de l'orthographe et de la langue française, mais avant tout le respect des lecteurs, à qui on ne devrait pas infliger le déchiffrage du message.

[FKP06] posent la question de la nuisance des NFCE et décrivent un ensemble de déviances inquiétantes dans le corpus de SMS qu'ils ont récoltées à travers l'opération *SMS pour la science*[12]. En particulier, ils remarquent que même si certaines formes sont lourdement abrégées (ce qui est typique du SMS), certaines erreurs ne sont pas volontaires. Par exemple les inversions des sons [e] et [ϵ], ou l'emploi du verbe *savoir* au lieu du verbe *être*, comme dans « *je saurais à Paris demain* » ou « *je serais pas me passer de toi* ». Leurs inquiétudes sont appuyées par [Ani02] qui publie une conversation **écrite** entre deux collégiennes, interceptée par un professeur, où l'on retrouve l'ensemble des déviances présentées dans cette étude.

Cela ne signifie pourtant pas que les déviances des NFCE et les problèmes d'orthographe sont corrélés. En effet, en France les réformes de l'enseignement de l'orthographe et de la lecture sont accusées d'avoir contribué à ce phénomène ([Bri04]), et les conséquences se sont fait sentir bien avant l'apparition des SMS qui n'ont peut-être fait que cristalliser un problème beaucoup plus large. Pourtant la question est d'importance : le 4 mai 2006, le journal *Le Monde* a publié un article intitulé *Les fautes d'orthographe deviennent un handicap pour faire carrière* décrivant les conséquences d'une mauvaise maîtrise de l'orthographe dans le milieu professionnel. Les fautes d'orthographe y sont une arme utilisée pour rabaisser les fautifs.

2.3 Un mot sur les MIMEMA

Nous consacrerons cette étude à la reconnaissance des MIMEMA (pour *Mini-Messages Manuscrits*). Cette forme de communication est pour le moment inexistante, donc impos-

[11] `http://sms.informatiquefrance.com/faq.htm`

[12] Cette équipe a récolté un large corpus de SMS avec le concours des écoles belges, voir `http://www.smspourlascience.be/`

sible à étudier directement, mais est susceptible d'être dès demain un standard de communication comme l'est rapidement devenu l'e-mail en son temps, ou les SMS aujourd'hui.

Le succès du *FlyPen* présage un avenir brillant pour la reconnaissance de l'écriture à partir de stylos numériques. Le *FlyPen*, développé par la société *LeapFrog*, malheureusement indisponible en France, est un stylo numérique à destination des plus jeunes. Il exploite la reconnaissance de l'écriture à travers quelques outils, notamment ludiques mais aussi pédagogiques, comme des cours interactifs de langues étrangères ou de mathématiques. Il a l'avantage d'offrir des services habituellement réservés aux ordinateurs dans un stylo léger et peu volumineux, utilisable partout et beaucoup moins onéreux qu'un ordinateur complet (autour de 100$).

On imagine alors que les stylos numériques, de plus en plus petits, légers et bon marché vont se généraliser dans un avenir proche, en particulier dans le cas de la téléphonie mobile, où il est possible de concevoir un stylo « allégé » puisque le téléphone portable embarque une bonne partie des ressources nécessaires aux traitements.

Notons qu'il est déjà possible depuis quelques années d'utiliser des stylos numériques pour émettre des messages de téléphone portable à téléphone portable mais, à ce jour, les applications se contentent d'envoyer l'image du parcours du crayon sur la feuille sans chercher à l'interpréter.

La figure 2.1 présente une situation d'utilisation des MIMEMA.

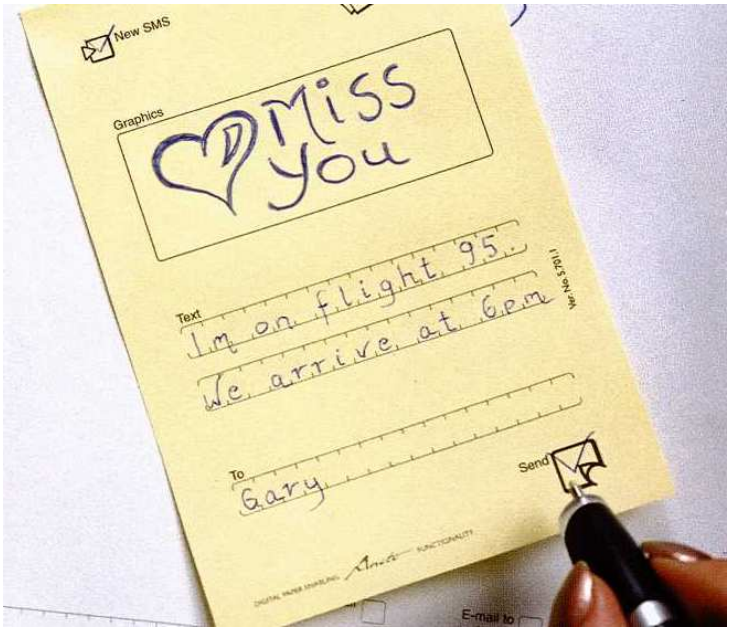

FIG. 2.1 – Un exemple d'utilisation des Mini-Messages Manuscrits

Chapitre 3

Introduction à la reconnaissance de l'écriture manuscrite

L'objet de cette étude n'étant pas directement lié à la reconnaissance de formes le sujet ne sera traité que superficiellement. De plus nous présentons ici la reconnaissance du point de vue de l'outil utilisé et dans le contexte de l'étude. Pour un document plus approfondi sur le sujet consulter [CL98].

3.1 À propos de *MyScript Builder*

Dans le cadre de cette étude, nous disposons d'un exemplaire du SDK [1] *MyScript Builder* en version 4.0 de la société *Vision Objects*.

Cet outil permet d'intégrer la technologie développée par *Vision Objects* dans n'importe quel programme. Il propose un ensemble de bibliothèques tournées vers la reconnaissance de caractères. Cela apporte plusieurs avantages, d'abord pour le développeur qui peut facilement intégrer cette technologie dans ses projets mais également pour la société *Vision Objects* qui peut vendre sa technologie sans en dévoiler les secrets de fabrication.

Nous présenterons dans la partie 5.2.1 le fonctionnement du SDK du point de vue de l'utilisateur/développeur. Nous présentons ici son fonctionnement interne, c'est-à-dire la façon dont il traite l'information.

3.2 Reconnaissance en-ligne et hors-ligne

On distingue deux formes de reconnaissance distinctes : la reconnaissance *en-ligne* et la reconnaissance *hors-ligne*. Dans la première, l'encre est fournie sous forme de coordonnées dans l'espace, ordonnées dans le temps (nous parlerons d'*encre numérique*). Il est donc possible de retracer le caractère, coups de crayon par coups de crayon. Il est éventuellement possible, selon les modèles de stylos numériques, de connaître la pression et l'angle du stylo, propres à chaque utilisateur.

L'écriture hors-ligne en revanche ne fournit qu'une image brute, sans plus d'information que *l'aspect* de l'image. C'est le cas par exemple de l'*OCR* (*Optical Character Recognition*)

[1] *Software Development Kit* – environnement de développement

utilisé dans la reconnaissance d'adresse sur les enveloppes. Un exemple des deux types de donnée est présenté en figure 3.1.

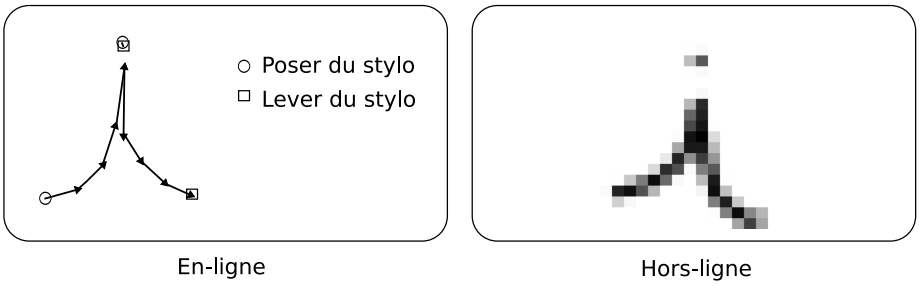

FIG. 3.1 – Écriture en-ligne et hors-ligne

La reconnaissance en-ligne apporte beaucoup plus d'informations que l'approche hors-ligne, mais n'est pas forcément accessible dans toutes les applications. Cette étude utilise exclusivement la reconnaissance en-ligne qui est la seule supportée par l'environnement de développement et qui est le support des MIMEMA.

3.3 Support de l'écriture

Pour cette étude nous avons utilisé des stylos numériques et du papier type *Anoto*. Le papier est pré-imprimé de points judicieusement placés permettant un repérage absolu et précis du stylo sur la feuille. Le stylo est muni d'une caméra infrarouge captant plusieurs points en même temps, mais aussi d'une pointe à bille classique pour aider le scripteur à suivre le fil de son écriture (ce qui n'est pas le cas avec une tablette graphique). Le processeur du stylo, en combinant les coordonnées des différents points captés est capable de calculer sa position sur la feuille. En effectuant un échantillonnage régulier, il enregistre son trajet sur la feuille. Les levés et posés de stylo sont également stockés.

3.4 Étapes de la reconnaissance avec *MyScript Builder*

Dans *MyScript Builder*, la reconnaissance de formes s'effectue à l'aide de trois experts logiciels travaillant de concert, se concentrant respectivement sur la segmentation, l'écriture et le langage. Nous présentons par la suite rapidement leur rôle sans toutefois rentrer dans les détails puisque nous n'intervenons pas dans le processus à ce niveau, mais uniquement d'un point de vue linguistique.

3.4.1 Segmentation

La segmentation est importante puisqu'une mauvaise segmentation conduira nécessairement à une reconnaissance erronée.

Pour segmenter un échantillon d'écriture manuscrite, l'expert le découpe d'abord en graphèmes (fig. 3.2), c'est-à-dire en unités graphiques minimales puis les rassemble pour

former les lettres les plus probables. Il est pour cela assisté de l'expert écriture et d'un modèle de langage.

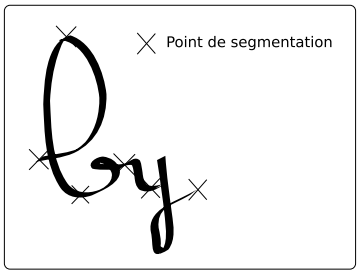

FIG. 3.2 – Segmentation d'un échantillon en graphèmes

3.4.2 Écriture

L'expert écriture va se consacrer à extraire des caractères connus à partir des groupes de graphèmes fournis par l'expert segmentation. Plus la segmentation est efficace, plus la reconnaissance l'est, et de la même façon, plus la reconnaissance est efficace et plus la segmentation l'est. C'est le paradigme *SegRec* : *Il faut segmenter pour reconnaître et il faut reconnaître pour segmenter.*

3.4.3 L'assistance du modèle de langage

L'utilisation d'un modèle de langage est très importante pour la reconnaissance manuscrite. En effet il subsiste toujours des ambiguïtés entre plusieurs candidats de la reconnaissance que l'on peut lever à l'aide d'un modèle stochastique du langage.

3.4.4 Collaboration des experts

Il est nécessaire pour les experts de collaborer, le moteur de reconnaissance étant le chef d'orchestre du processus. Les experts communiquent entre eux, le travail n'est pas séquentiel (*segmentation-écriture-langage*) mais des aller-retours sont effectués entre les différents experts pour affiner la qualité de la reconnaissance.

L'exemple de la figure 3.3 illustre la collaboration des différents experts.

Plusieurs segmentations sont possibles pour l'échantillon d'encre présenté, par exemple « lrj » ou « by ». A priori, rien dans la reconnaissance de forme ne permet de trancher, toutefois le mot « lrj » n'appartenant pas à la langue, c'est l'hypothèse « by » qui sera retenue.

Les ambiguïtés sont généralement beaucoup plus profondes et le modèle de langue pourra chercher, au delà de l'existence d'un mot dans un lexique, à retrouver des combinaisons de mots connues [Per05].

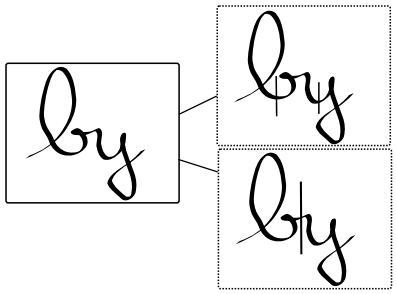

FIG. 3.3 – Plusieurs segmentations possibles pour l'échantillon d'encre numérique

Chapitre 4

Problématique

4.1 Des modèles de langage pour les MIMEMA

Nous avons vu en introduction qu'il est nécessaire d'apporter un maximum d'informations sur la forme des données à un système de reconnaissance pour espérer des performances et des résultats raisonnablement bons. Ce travail a été effectué pour l'outil que nous utilisons ici dans le cas du français, ainsi que pour la plupart des langues « courantes » en ajoutant un lexique des mots de langue, mais aussi un modèle de langage [Per05].

À notre tour, nous cherchons ici des *modèles de langage* pour les MIMEMA que nous intègrerons à l'outil de reconnaissance de l'écriture manuscrite pour augmenter la précision du système.

4.1.1 Hypothèse

Les MIMEMA n'étant pas utilisés à ce jour, nous parions sur une hypothèse en particulier. Nous devons supposer que l'utilisateur d'un stylo numérique écrira ses SMS de la même façon qu'avec le clavier de son téléphone portable.

C'est un choix risqué et cette hypothèse est vraisemblablement fausse comme tend à le montrer la littérature sur le sujet [Ani02]. Nous devons toutefois agir de cette façon puisqu'il est aujourd'hui impossible de savoir quelles formes prendront les MIMEMA s'ils viennent à se populariser. En effet, les usages n'apparaissent pas du jour au lendemain mais bien au fil des utilisations, lors de la prise en main de l'outil.

Il est donc important de garder ce point en vue puisque l'étude ne pourra se consacrer pour le moment qu'à la reconnaissance de SMS manuscrit et non pas exactement à la reconnaissance des MIMEMA.

4.1.2 Objets de l'étude

Les NFCE telles que présentées ici ne seront pas globalement traitées. Nous nous concentrerons uniquement sur l'étude des SMS. Nous avons de plus présenté précédemment beaucoup d'aspects des NFCE qui ne seront pas nécessairement représentés dans les MIMEMA, à commencer par les fautes de saisies qui, nous l'avons déjà dit, sont inhérentes à la saisie au clavier et seront donc improbables si la saisie se fait avec un stylo numérique.

Nous chercherons plutôt à capter le fonctionnement des structures productrices pour être capable de les détecter et de les interpréter efficacement. Notons bien que nous n'avons pas besoin de *comprendre* le message (comme c'est partiellement fait dans [Bov05]), mais uniquement d'être capable de le *reconnaître* le plus fidèlement possible.

Chapitre 5

Prise en main des outils et des ressources

5.1 Le corpus de MIMEMA

Les premiers travaux du projet MIMEMA [VGM05] ont permis la création d'un corpus de MIMEMA. Les scripteurs étaient invités à remplir un formulaire (fig. 5.1) avec un stylo numérique (de type Nokia). Le formulaire réclamait plusieurs types d'échantillons d'encre manuscrite :

- des entrées casées imposées (le texte à saisir était imposé) ;
- des entrées casées non imposées ;
- des entrées non casées imposées ;
- des entrées non casées non imposées ;
- de longs textes non casés non imposés.

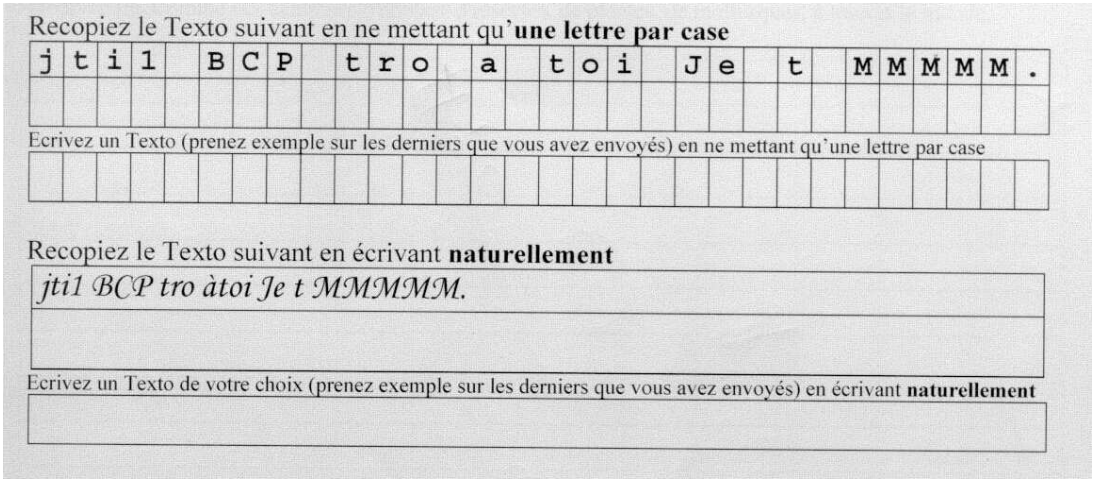

FIG. 5.1 – Une partie du formulaire ayant servi pour la construction du corpus

Il est important de se souvenir de ces différents types (en particulier il est facile de confondre *casé* et *imposé*). Par la suite, nous parlerons d'écriture **structurée** lorsque le scripteur est contraint d'écrire d'une certaine façon (par exemple, une lettre par case), comme dans la figure 5.2 et nous parlerons d'écriture **libre** lorsque le scripteur écrit naturellement (ce qui est le cas de l'écriture cursive), comme présenté figure 5.3.

FIG. 5.2 – Un exemple d'écriture structurée

FIG. 5.3 – Un exemple d'écriture libre

Cette collecte a été réalisée auprès de 150 personnes différentes. Le nombre de MIMEMA collecté après nettoyage est présenté dans la table 5.1.

	Structuré	Libre
Imposé	177	174
Non imposé	493	477
Total	670	551

TAB. 5.1 – Nombre de MIMEMA collectés

Après nettoyage, chaque échantillon d'encre contient, en plus de l'ensemble des coups de crayons, le **label**, c'est-à-dire le texte en clair, utilisé pour mesurer la qualité de la reconnaissance. Précisons enfin que pour les échantillons d'écritures **structurées** récoltés, la segmentation est imposée **dans le fichier d'encre**. Elle n'est donc pas à refaire par le système de reconnaissance.

5.2 L'environnement de développement MyScript Builder

5.2.1 Schéma du processus de reconnaissance

À notre niveau nous pouvons intervenir de deux façons dans le processus de reconnaissance. Nous pouvons faire varier les ressources attachées au moteur, voire créer nos propres ressources. Nous pouvons également sélectionner selon nos propres critères le résultat le plus probable parmi ceux proposés. La figure 5.4 présente le fonctionnement général de l'application.

5.2.2 Ressources linguistiques & alphabétiques

Ressources Linguistiques Les ressources linguistiques (*Linguistic Knowledge*) indiquent au moteur ce qu'il est supposé reconnaître d'un point de vue linguistique. Par exemple si le texte saisi est un nom de ville il faut y associer la ressource correspondante pour optimiser la reconnaissance. Dans beaucoup de cas il n'est pas forcément possible de caractériser le

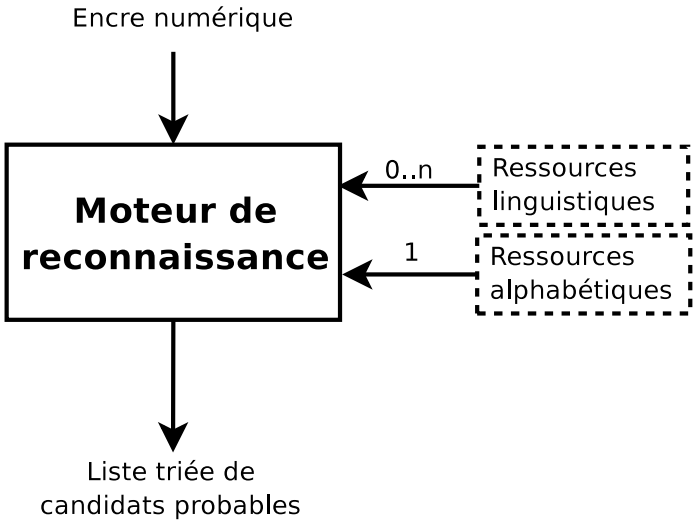

FIG. 5.4 – Schéma simplifié du processus de reconnaissance

texte de cette façon. Deux ressources génériques sont présentées dans la table 5.2. Chaque langue supportée possède un jeu de ressource linguistique propre.

Ressource	Description	Usage
`lk-free`	Possibilité de reconnaître des mots inconnus	Recommandé lorsqu'on ne dispose pas d'information sur la forme du texte.
`lk-text`	associe : – un lexique (localisé) – un modèle de langue (localisé) s'appuyant sur le contexte – la possibilité de reconnaître des mots inconnus	Recommandé pour un langage « correct » et pour la prise de notes.

TAB. 5.2 – Ressources linguistiques

La possibilité de reconnaître des mots nouveaux privilégie les combinaisons de classe, par exemple les suites de majuscules, les suites de minuscules et les suites de cardinaux. La ressource linguistique *lk-text* apporte un *a priori* beaucoup plus fort que la ressource *lk-free*. On dit que la première *ferme* le langage en le contraignant fortement, alors que la seconde l'*ouvre* puisqu'elle n'apporte que peu de contraintes.

Il est possible de *ne pas* associer de ressource linguistique au moteur de reconnaissance. Il ne dispose alors ni de lexique, ni de modèle de langage.

Ressources Alphabétiques Les ressources alphabétiques (*Alphabetic Knowledge*) servent à la reconnaissance de formes. Trois ressources sont à notre disposition (table 5.3).

Ressource	Description
`ak-cur`	écriture cursive (naturelle)
`ak-iso`	écriture isolée : la segmentation est imposée et n'est pas à refaire par le système de reconnaissance (cas de l'écriture *pré-casée*)
`ak-hpr`	écriture séparée : le scripteur doit terminer une lettre (et tous les signes diacritiques associés) et lever le stylo avant de former une nouvelle lettre (la segmentation est implicite)

TAB. 5.3 – Ressources alphabétiques

5.3 Premières expérimentations

Nous avons souhaité dans un premier temps tester le moteur fourni sans aucune adaptation, dans le but de prendre en main l'outil, mais aussi pour obtenir un étalon de comparaison utilisé dans la suite de l'étude.

5.3.1 Protocole

Traitements préliminaires

Quelques échantillons d'encre étant manifestement erronés, ils ont été écartés des jeux de tests. Nous avons toutefois veillé à supprimer uniquement les échantillons incorrects d'un point de vue méthodologique, mais certainement pas les exemples mal reconnus (fig. 5.7) en raison de la mauvaise qualité d'écriture du scripteur.

Nous avons constaté deux types d'échantillons incorrects. Dans certains cas comme présenté figure 5.5, deux échantillons d'encre sont superposés, ce qui n'est pas normal. Dans d'autre cas, la segmentation imposée (cas de l'écriture structurée) est incorrecte, c'est le cas figure 5.6 ou le point du « i » est à tort associé au « c » de la ligne du dessus *dans le fichier d'encre numérique*. La segmentation imposée induit le moteur en erreur qui reconnaît un « ç ».

FIG. 5.5 – Un exemple d'échantillons superposés, écarté

Quelques modifications sont apportées sur les chaînes à traiter. En ce qui concerne le texte original, nous supprimons les symboles indiquant un retour chariot (le moteur les reconnaît mais ne les indique pas). En ce qui concerne le texte reconnu, le moteur insère

FIG. 5.6 – Un exemple d'échantillon mal segmenté, écarté

FIG. 5.7 – Un exemple d'échantillon difficile, conservé

des espaces pour respecter les conventions typographiques. Nous les retirons pour revenir au texte d'origine.

Nous comparons une version bas de casse des deux textes, c'est-à-dire qu'ils sont passés intégralement en minuscules avant d'être comparés. C'est une simplification abusive car l'emploi des majuscules/minuscules dans les SMS n'est pas anodin (une majuscule indique généralement une écriture rébus, comme dans « KC », ou bien peut signifier que le scripteur veut « parler fort », pour marquer l'insistance ou la colère), mais n'étant pas en mesure de fournir une ligne de base indiquant au moteur la position des lettres, il lui est impossible de faire la différence entre certaines lettres et leurs versions capitales (par exemple un « c » minuscule et majuscule).

Distance entre le texte reconnu et le texte à reconnaître

Nous chercherons à mesurer la distance d'édition dite *distance de Levenshtein* (Levenshtein, 1965) entre le texte reconnu par *MyScript* et le texte original. Cette mesure donne le nombre minimum d'opérations sur les caractères (parmi suppression, substitution et insertion) nécessaires pour passer d'une chaîne à l'autre. Nous chercherons également à mesurer le nombre de mots correctement reconnus en généralisant la distance de Levenshtein.

Calcul du taux de reconnaissance

Le taux de reconnaissance indique le nombre d'éléments corrects par rapport au nombre d'éléments traités. Nous le mesurons à deux niveaux.

Au niveau "caractère" Nous mesurons combien de caractères sont correctement reconnus par rapport au nombre de caractères de la phrase d'origine. Pour ce faire, nous retirons la distance de Levenshtein calculée au nombre de caractères puisqu'elle est censée correspondre au nombre de caractères incorrects, voir l'exemple table 5.4.

$$Taux_{caractères} = \frac{|caractères\ corrects|}{|caractères\ traités|} = \frac{|caractères\ traités| - distance_{caractères}}{|caractères\ traités|}$$

Texte original :	`é k f tu dmin? Tu va o resto?`
Texte reconnu :	`é k` **`t`** `tu dmin? Tu` **`r`**`a o` **`v`**`esto?`
Distance :	3
Précision :	$21 - 3 = 18 \rightarrow 18/21 = 85\%$ de caractères correctement reconnus

TAB. 5.4 – une mesure de taux de précision correct

Toutefois, cette mesure n'est pas toujours correcte. Si la reconnaissance introduit de nouveaux caractères erronés (opération d'insertion dans la mesure de distance), la précision, qui est calculée sur la taille de la chaîne d'origine sera incorrecte puisqu'établie à partir d'une chaîne de caractère plus longue. Elle peut même être négative dans certains cas, comme présenté table 5.5.

Texte original :	`bjr`	(taille : 3)
Texte reconnu :	**`lo`**`j.`**`t`**	(taille : 5)
Distance :	4	
Précision :	$3 - 4 = -1 \rightarrow -1/3 = -33\%$ de caractères reconnus	
Précision (réelle) :	$1/3 \rightarrow 33\%$ de caractères reconnus	

TAB. 5.5 – une mesure de taux de reconnaissance incorrect

Nous avons donc pris la décision de ne pas compter le coût de l'insertion. Compte tenu du signal traité, une insertion est associée à une substitution (qui elle est pénalisée à 1). Ainsi, le taux de reconnaissance est mesuré à partir du nombre d'éléments corrects et non à partir du nombre d'éléments incorrects.

L'exemple précédent avec la version corrigée est repris table 5.6

Texte original :	`bjr`	(taille : 3)
Texte reconnu :	**`lo`**`j.`**`t`**	(taille : 5)
Distance :	2	
Précision :	$3 - 2 = 1 \rightarrow 1/3 = 33\%$ de caractères reconnus	

TAB. 5.6 – la mesure après correction

Au niveau "mot" Nous cherchons à connaître combien de mots ont été correctement reconnus sur le nombre de mots traités. Pour cela nous utilisons à nouveau la distance de Levenshtein sur une transformation des chaînes de caractères. Nous associons à chaque mot un identifiant (dans notre cas, un entier) dans une table de hachage. Nous reconstruisons deux nouvelles chaînes à partir de ces identifiants que nous comparons avec l'algorithme de distance de Levenshtein (voir table 5.7).

$$Taux_{mots} = \frac{|mots\ corrects|}{|mots\ traités|} = \frac{|mots\ traités| - distance_{mots}}{|mots\ traités|}$$

Texte original :	é	k	f	tu	dmin ?	tu	va	o	resto ?
Chaîne reconstruite :	1	2	3	4	5	6	7	8	9
Texte reconnu :	é	k	t	tu	dmin ?	tu	ra	o	vesto ?
Chaîne reconstruite :	1	2	10	4	5	6	11	8	12
Comparaison :	1	2	3	4	5	6	7	8	9
	1	2	**10**	4	5	6	**11**	8	**12**
Distance :	**3**								
Précision :	$9 - 3 = 6 \rightarrow 6/9 = 66\%$ de mots corrects								

TAB. 5.7 – Taux de reconnaissance au niveau mot

La mesure utilisée est la même qu'au niveau caractère, c'est-à-dire que la suppression et la substitution sont pénalisées à 1, mais le coût de l'insertion est nul, pour les mêmes raisons que précédemment.

Nous aurions pu également mesurer la précision au niveau phrase, c'est-à-dire le nombre de phrases intégralement correctement reconnues par rapport au nombre de phrases traitées. Cette mesure n'est toutefois pas pertinente puisqu'elle ne distingue pas les phrases très mal reconnues des phrases présentant peu d'erreurs. Les résultats sont très faibles, mais surtout très peu significatifs et l'on ne peut en tirer aucune conclusion.

Par la suite, nous parlerons indifféremment de précision ou de taux de reconnaissance.

5.3.2 Reconnaissance de l'écriture libre

Nous avons à notre disposition un ensemble de MIMEMA cursifs, imposés et non imposés. Les textes étaient écrits naturellement par les scripteurs (voir figure 5.1). Reconnaître cette écriture est très difficile puisque le moteur doit se charger de la segmentation puis de la reconnaissance, ces deux étapes nécessitant chacune la combinaison de la reconnaissance de forme et d'un modèle de langage. Une mauvaise segmentation entraîne inévitablement une mauvaise reconnaissance.

Les ressources utilisées étaient l'alphabet *ak-cur* (alphabet cursif) et les ressources linguistiques *lk-text* et *lk-free*.

Les textes imposés étaient par exemple :

– « *gspère qtu va bien a+ biz* » – j'espère que tu vas bien, à plus, bise
– « *é k f tu dmin ? Tu va o resto ?* » – et que fais-tu demain ? Tu vas au resto ?
– « *jti1 BCP tro à toi Je t'MMMM.* » – je tiens beaucoup trop à toi, je t'aime.
– « *Alors KV vous Pchez IR ?* » – Alors qu'avez-vous pêché hier ?
– « *jatend son cou 2fil Bizoo a2m1* » – J'attends son coup de fil, bisous à demain.
– « *g u 16 en fisic supppeeer* » – j'ai eu 16 en physique, super !
– « *HT du p1 D poiro et du kf stp* » – Achète du pain et des poireaux et du café stp.
– « *Ya qqun ki pourRé vnir mcherché* » – Y a quelqu'un qui pourrait venir me chercher ?
– ...

Ces résultats sont cohérents avec ce que l'on pouvait attendre des ressources, telles que présenté dans [Vis06] : la ressource *lk-text* est trop contrainte pour reconnaître efficacement le *langage texto* alors que la ressource *lk-free* est la plus adaptée puisque prévue pour les textes inconnus.

	Précision (car) sur 6967	Précision (mot) sur 1696
sans ressource	87%	36%
lk-text	84%	48%
lk-free	88%	43%

TAB. 5.8 – Mesure de la précision en fonction des ressources linguistiques

À titre de comparaison, nous avons effectué les mêmes mesures pour un texte en français « correct », libre et imposé. Les résultats sont consignés dans la table 5.9.

	Précision (car) sur 61 014	Précision (mot) sur 10 223
sans ressource	81, 4%	14, 9%
lk-text	91, 3%	60, 3%
lk-free	82, 6%	27, 6%

TAB. 5.9 – Mesure de la précision en fonction des ressources linguistiques

La ressource la plus adaptée (*lk-text*, puisque le texte est écrit en français correct) donne les meilleurs résultats avec 60% de précision au niveau mot, ce qui est à nouveau conforme à l'usage préconisé des ressources.

5.3.3 Reconnaissance de l'écriture structurée

Nous avons également à notre disposition un ensemble de MIMEMA pré-casés, imposés et non imposés, c'est-à-dire que le scripteur était contraint d'écrire dans les cases du formulaire (voir à nouveau figure 5.1). Ce type de reconnaissance est évidemment le plus efficace et le plus rapide puisque la segmentation est donnée par les cases et n'est pas à refaire par le moteur. Les phrases imposées étaient les mêmes que précédemment.

Nous avons testé l'efficacité de différentes combinaisons de ressources parmi les alphabets *ak-iso* (où le scripteur doit écrire une lettre par case) – table 5.10 – et *ak-hpr* (où l'on impose au scripteur de lever le stylo entre chaque caractère) – table 5.11 – et parmi les ressources linguistiques *lk-text* (français standard, lexique et modèles de langages associés) et *lk-free* (modèle de langage seulement).

	Précision (car), sur 5406	Précision (mot), sur 1232
sans ressource	89, 6%	62%
lk-text	90, 5%	62%
lk-free	91, 0%	67%

TAB. 5.10 – Mesure de la précision avec la ressource *ak-iso*

Ici encore ces résultats sont cohérents : la ressource *lk-free* est encore la plus efficace pour ces échantillons. La ressource *lk-text* est trop contrainte pour les phrases présentées et l'absence de ressource n'oriente pas suffisamment la reconnaissance pour atteindre la

	Précision (car) sur 5406	Précision (mot) sur 1232
sans ressource	94%	70%
lk-text	90%	59%
lk-free	95%	72%

TAB. 5.11 – Mesure de la précision avec la ressource *ak-hpr*

qualité obtenue avec *lk-free*. Il est toutefois surprenant de constater que, bien que l'écriture soit pré-casée, ce soit l'alphabet *ak-hpr* qui offre la meilleure qualité de reconnaissance. L'alphabet *ak-iso* traite en effet chaque caractère séparément en ce qui concerne la reconnaissance de forme, alors que l'alphabet *ak-hpr* s'appuie sur les lettres voisines pour chercher une ligne de références (en s'appuyant sur les jambages des lettres par exemple). Il est donc plus efficace pour distinguer certains caractère ambiguë sans cette information, tel que le "l" et le "e" minuscule. Ceci est une hypothèse probable pour expliquer la différence de résultat entre les deux ressources, mais nous ne l'avons pas vérifiée.

Il peut également sembler étrange de voir apparaître une si grande différence entre la précision au niveau caractère et la précision au niveau mot. Ceci s'explique très simplement en raison de la taille moyenne des mots et de la répartition des erreurs. Imaginons par exemple que nous obtenions une précision de 50% au niveau caractère, ce qui signifie qu'un caractère sur deux est erroné ; si les erreurs sont réparties équitablement, tous les mots de plus d'une lettre seront touchés (et les mots d'une lettre seront touchés une fois sur deux), la précision niveau mot chutera fortement, largement en dessous de 50%.

C'est le même phénomène qui apparaît ici : avec 90% de précision au niveau caractère, un caractère sur dix est erroné. Compte tenu de la répartition de ces erreurs et de la taille moyenne des mots, beaucoup plus d'un mot sur dix seront touchés par ces erreurs et la précision niveau mot sera de toute façon inférieure à 90%. C'est pour cette même raison, évoquée précédemment, que la précision au niveau phrase n'apporte que peu d'informations sur la qualité de la reconnaissance.

5.3.4 Utilisation d'une ressource optimale

Pour obtenir un étalon de comparaison, nous avons construit une ressource optimale : c'est un lexique qui contient l'ensemble des mots à reconnaître. Pour chaque corpus d'échantillon d'encre, nous extrayons, à partir du label, l'ensemble des mots à reconnaître. Nous pouvons ensuite construire une ressource linguistique (un simple lexique donc) qui couvrira l'ensemble des mots du corpus en apportant un minimum de bruit puisqu'il n'y aura aucun mot « en trop » dans le lexique.

Cette ressource n'est bien sûr pas utilisable dans un usage « normal », puisque apprise à partir du même corpus que celui sur lequel elle est appliquée, mais elle nous permet de mesurer la reconnaissance optimale et nous donne un encadrement de ce que nous pouvons espérer obtenir comme qualité de résultat.

Reconnaissance de l'écriture libre Les résultats des mesures de reconnaissance de l'écriture libre avec la ressource *optimale* sont présentés table 5.12.

	Précision (car) sur 6967	Précision (mot) sur 1696
optimale	96%	77%
lk-text, optimale	95%	71%
lk-free, optimale	88%	41%

TAB. 5.12 – Mesure de la précision avec la ressource *ak-cur*

Reconnaissance de l'écriture structurée Les résultats des mesures de reconnaissance de l'écriture structurée avec la ressource *optimale* sont présentés table 5.13 pour la ressource alphabétique *ak-iso* et table 5.14 pour la ressource alphabétique *ak-hpr*. La table présentant les résultats pour la ressource *ak-iso* est plus détaillée pour mettre en évidence le fait que tous les éléments n'ont pas été reconnus (voir première ligne) et que la mesure concernée ne sera pas retenue.

	Précision (car) sur 5406	Précision (mot) sur 1232
optimale	$4243/4372 = 97\%$	$881/1006 = 88\%$
lk-text, optimale	$5191/5406 = 96\%$	$1018/1232 = 82\%$
lk-free, optimale	$4930/5406 = 91\%$	$827/1232 = 67\%$

TAB. 5.13 – Mesure de la précision avec la ressource *ak-iso*

	Précision (car) sur 5406	Précision (mot) sur 1232
optimale	96%	86%
lk-text, optimale	96%	82%
lk-free, optimale	95%	73%

TAB. 5.14 – Mesure de la précision avec la ressource *ak-hpr*

Analyse

Dans le cas de la reconnaissance de l'écriture structurée comme de l'écriture libre, nous obtenons les meilleurs résultats avec la ressource linguistique *lk-free*. En y combinant la ressource optimale, nous parvenons à des résultats sensiblement identiques (en terme de précision). En revanche, toujours avec la ressource optimale, nous obtenons de bien meilleurs résultats avec la combinaison *lk-text* + *optimale* : de 7% à 4% d'erreur pour la reconnaissance structurée, et de 16% à 7% dans le cas de l'écriture libre.

Ces résultats s'expliquent facilement : sans informations supplémentaires, la ressource *lk-free* est meilleure que la ressource *lk-text* puisqu'elle est plus adaptée à une écriture inconnue, elle est plus *ouverte*[1] alors que les contraintes de langage imposées par la ressource *lk-text* privilégient les formes connues, moins efficace dans le cas des MIMEMA,

[1]la documentation de *MyScript Builder* [Vis06] la conseille lorsque la forme du texte à reconnaître est inconnue

sauf si on y ajoute un lexique adapté, ce qui est le cas avec lorsqu'on la combine avec la ressource optimale.

Nous pouvons donc espérer obtenir des résultats entre ces deux valeurs. Nous pouvons en effet améliorer le processus de reconnaissance, en l'adaptant au problème, mais nous aurons du mal à dépasser le taux obtenu avec le lexique optimal.

Notre travail par la suite sera donc de construire des ressources linguistiques que nous testerons en combinaison avec les ressources existantes et que nous comparerons à ces résultats préliminaires.

5.3.5 Séparation des différentes formes productives

Nous avons ensuite découpé le corpus en fonction des différentes formes productives. Parmi l'ensemble des formes décrit en partie 2.1.3 nous n'avons retenu que les formes suivantes :

- **rébus** : *9, c, A+, ct, 2pui, gm, IR…*
- **squelettes consonantiques** : *Slt, dvt, qd, avc, bjr…*
- **agglutinations** : *oconcert, moitu, mapler, savatrebien, qeske…*
- **phonétisations** : *koi, fé, comen, é, nouvo…*

Les *étirements graphiques* n'ont pas été séparés car ils n'apparaissaient que très rarement dans les échantillons d'encre numériques. Les tailles des différents sous-corpus sont résumées dans la table 5.15.

Corpus	Rébus	Squelettes	Agglutinations	Phonétisations	Autres
Taille (mot)	96	54	36	92	756
Taille (caractère)	222	151	199	328	2636

TAB. 5.15 – Taille des sous-corpus

Les corpus sont assez réduits, le nombre de mots dans chacun n'est pas suffisamment important pour y faire des mesures significatives telles que le nombre d'apparitions d'un mot dans le corpus et ce, d'autant plus qu'une partie de ces mots est imposée au scripteur. Le corpus *Autres* correspond à tous les mots non classés, c'est-à-dire généralement non déviés. Nous effectuons toutefois une série de reconnaissance sur chacun de ces corpus et nous les comparons avec les mesures obtenues dans le cadre générique (voir table 5.16).

	Sans		lk-text		lk-free		Optimale		Opt. + lk-text		Opt. + lk-free	
	car	mot	car	mot	car	mot	car	mot	car	mot	car	mot
Globale	94	69	95	68	90	54	96	86	96	82	96	82
Rébus	92	84	69	45	93	84	*97*	*98*	95	90	95	88
Squelette	94	85	66	15	95	85	100	100	98	96	95	87
Agglutination	90	64	77	6	91	56	100	100	96	81	91	58
Phonétisation	92	71	75	24	94	79	99	98	99	97	94	80
Autres	92	75	93	80	94	79	97	95	97	90	94	80

TAB. 5.16 – Comparaison de la précision en fonction du corpus et des ressources utilisées

Les mesures de la table 5.16 sont effectuées avec la ressource alphabétique *ak-hpr*. La ressource *Optimale* est calculée à chaque mesure, elle correspond donc précisément au lexique du corpus testé. La première mesure est la précision au niveau caractères (en pourcentage), la seconde est la précision au niveau mots (en pourcentage également). En grisé les meilleures mesures sans et avec la ressource optimale (la priorité est mise sur la mesure du taux de reconnaissance au niveau mot), correspondant à la borne inférieure et la borne supérieure de la précision espérée (cf 5.3.4). Certaines mesures ne sont pas représentatives car elles ne couvrent pas l'ensemble du corpus concerné (certains traitements ne conduisent pas à une proposition de reconnaissance), elles sont indiquées en italique.

Nous avons maintenant à notre disposition un ensemble d'étalons de comparaison qui nous donne, pour chaque forme, l'encadrement de la précision que nous pouvons espérer obtenir en améliorant les traitements. Nous allons à présent travailler, forme par forme, à améliorer ces résultats en proposant des modèles de langage au système.

Chapitre 6

Amélioration des résultats de la reconnaissance

6.1 À propos des corpus isolés précédemment

Dans un premier temps nous travaillons sur chaque forme séparément. Nous essayons donc d'améliorer les résultats de la reconnaissance pour les rébus, les squelettes consonantiques, les agglutinations et les phonétisations.

Les corpus ne sont pas suffisamment importants pour nous permettre d'y appliquer les traitements statistiques habituels. Typiquement, nous ne pouvons raisonnablement pas découper chaque corpus en un corpus d'apprentissage et un corpus de test sans biaiser les résultats.

Nous n'avons pas non plus à notre disposition un corpus plus large de SMS, qui aurait pu être utilisé puisque, étant donné que nous ne n'intervenons pas dans le processus de reconnaissance de formes, il n'est pas nécessaire de disposer des échantillons d'encre numérique pour l'apprentissage (dans ce cas, notre corpus de MIMEMA aurait constitué l'ensemble du corpus de test).

Nous allons donc extraire des propriétés connues de chacune des formes à partir de nos observations et les « faire connaître » au système de reconnaissance.

6.2 Traitement des formes isolées

6.2.1 Caractérisation des différentes formes

Rappelons ici les propriétés et surtout les différences de chaque forme présentés succintement en partie 2.1.3, tout d'abord quelques nouveaux exemples :

- **Rébus** : *C, 9, 2m1*…
- **Agglutinations** : *eske, kesketufé, sava*…
- **Squelettes consonantiques** : *dvt, avc, lgtps, slt*…
- **Phonétisations** : *é, koi, kwa*…

Rébus Les *rébus* sont généralement un mélange de lettres et de chiffres. Par opposition aux *phonétisations*, il faut lire les lettres mises en évidence par leurs noms, et non par le

son associé. À titre d'exemple avec le mot « paC » : il ne faut pas lire « pa-que » mais bien « pa-sé ». De la même façon, il faut prononcer les chiffres tels quels lorsqu'ils sont mélangés aux lettres. Ainsi « 2m1 » doit se lire « deux-m-un ».

Squelettes consonantiques Un squelette consonantique est une abréviation d'un mot commun, composé quasi-exclusivement de ses consonnes. Au contraire des autres formes il ne faut pas « lire » le mot directement mais le reconstruire avant lecture.

Agglutinations Une agglutination est un « collage » de mots juxtaposés pour former un nouveau mot unique. C'est une forme parfois difficile à caractériser car mêlant également des phonétisations, des rébus et des squelettes consonantiques. Nous classons dans cette catégorie toutes les formes qui, une fois transcrites en français correct, nécessitent plusieurs mots pour être exprimées. Par exemple « *eske* » se retranscrit en « *est-ce que* ».

Phonétisations Une phonétisation est la retranscription phonétique d'un mot, généralement à but d'abréviation. Il peut s'agir tout simplement de retirer les lettres muettes d'un mot (« *grave* » $\rightarrow$ « *grav* », « *pas* » $\rightarrow$ « *pa* »...), ou bien encore de remplacer des sons composés par la combinaison de plusieurs lettres par une seule (« *jamais* » $\rightarrow$ « *jamé* », notons ici le remplacement du son e par le son ϵ, comme présenté dans [FKP06]).

Nous n'avons pas été en mesure, dans cette étude, de traiter les agglutinations et nous ne traitons les phonétisations que superficiellement. En effet, bien qu'il soit assez évident de *reconnaître* ces formes pour un humain, elles semblent *a priori* trop créatives pour se plier aux traitements classiques du traitement automatique du langage qui se base souvent sur des caractéristiques morphologiques ou syntaxiques, à peu près inexistantes dans ces cas.

6.2.2 Traitement des squelettes consonantiques

Règles de transformations

[Ani02] écrit :

> *On sait depuis longtemps grâce à la théorie de l'information que les consonnes ont une valeur informative plus forte que les voyelles. Le mot français écrit est fortement charpenté autour des consonnes, dont certaines n'ont pas de contrepartie phonique. Les consonnes retenues comportent toujours la première et la dernière ; les consonnes en position faible dans les groupes consonantiques (<l, r, h> précédés d'une consonne en début de syllabe, <n, m> suivis d'une consonne en fin de syllabe) sont en général éliminées.*

Cela nous donne quelques règles de transformation pour transcrire un mot de la langue française en son squelette consonantique.

1. Conservation de la première et de la dernière consonne
2. Suppression des voyelles
3. Suppression de <l, r, h> lorsqu'ils sont situés après une consonne, en début de syllabe

4. Suppression de <n, m> lorsqu'ils sont situés avant une consonne, en fin de syllabe

Par exemple, pour le mot « longtemps » :
– `l` et `s` sont conservés → `l...s` ;
– `l`**`o`**`ngt`**`e`**`mps` : les deux voyelles `o` et `e` sont supprimées (règle 2) → `lngtmps` ;
– `l`**`n`**`t`**`m`**`ps` : le `n` et le `m` sont éliminés (règle 3) → `lgtps` ;
Le résultat obtenu est `lgtps`.

Toutefois les squelettes consonantiques ne sont pas toujours aussi déterminés, `toujours` est souvent également abrégé en `tjs` alors que la transformation proposée donne `tjrs`.

De même, il est parfois nécessaire de conserver certaines voyelles (en particulier les voyelles d'attaque) comme dans `avec` qui s'abrège en `avc`.

Nous modifions donc sensiblement les règles proposées par Anis :

1. Conservation de la première et de la dernière consonne **ainsi que les voyelles situées avant la première, et après la dernière** ;
2. Suppression des voyelles ;
3. Suppression de <l, r, h> lorsqu'ils sont situés après une consonne, en début de syllabe
4. Suppression de <n, m> lorsqu'ils sont situés avant une consonne, en fin de syllabe

Cela donne pour le mot « indépendance » :
– Conservation de `in...ce` (règle 1) ;
– Conservation de `indpndce` (Règle 2) ;
– Suppression du « n » → `indpdce` (Règle 4) ;
Le résultat obtenu est `indpdce`.

Constitution d'un lexique de squelettes consonantiques

À partir de ces quelques règles, nous avons construit un automate qui transforme un mot français en un de ses squelettes consonantiques. Nous appliquons cet algorithme sur un corpus français, étiqueté et nettoyé pour obtenir un lexique de squelette consonantique que nous intégrons au système de reconnaissance.

Seuls certains mots sont traités, ce sont :
– les adverbes ;
– les adjectifs ;
– les substantifs.

Il ne nous semblait pas pertinent de traiter les verbes, puisque la variation de leur morphologie en fonction de leur conjugaison enlève une grosse partie de son intérêt à ce traitement. De même, les mots « rares » ne sont pas ajoutés au lexique, de façon à ne pas générer trop de bruit. Nous extrayons finalement 3244 mots transformés en squelette consonantique à partir d'un corpus du *Monde* étiqueté grammaticalement, ainsi que d'un lexique de mots-outils mis à notre disposition.

Construction d'une expression régulière

Il est nécessaire d'associer au lexique précédemment construit une expression régulière, utilisée en tant que « soupape » lorsque le mot à reconnaître n'apparaît pas dans le lexique.

L'expression, formée à partir de l'observation des squelettes mais aussi d'une série de statistiques issues du lexique est présentée dans la table 6.1, l'automate correspondant est présenté en figure 6.1. Notons que *MyScript* permet d'utiliser des expressions régulières probabilistes. Les lignes sont numérotées pour commenter cette expression.

Classification des symboles :
1 `consonne = [bcdfghjklmnpqrstvwxz]`
2 `voyelle = [aeiouyéèàîâê]`
Mots composés uniquement de consonnes :
3 `groupe_cons = ({consonne})+`

Mots contenants des groupes de voyelles hors extrémités : (type 1)
Syllabes contenants une voyelle (« tor »)
4 `melange_1 20% = {consonne} {voyelle} {consonne}`
Syllabes contenants deux voyelles (« jour »)
5 `melange_2 80% = {consonne} {voyelle} {voyelle} {consonne}`
6 `melange = ({melange_1} | {melange_2})+`

Mots contenants des groupes de voyelles aux extrémités : (type 2)
7 `voy_deb 70% = {voyelle}+ {consonne}+`
8 `voy_fin 7% = {consonne}+ {voyelle}+`
9 `voy_deux 23% = {voyelle}+ {consonne}+ {voyelle}+`

10 `type1 20% = {voy_deb} | {voy_fin} | {voy_deux}`
11 `type2 80% = ({groupe_cons} {melange}?)+`

Point d'entrée de l'expression :
12 `squelette = {type1} | {type2}`

TAB. 6.1 – Expression régulière pour les squelettes consonantiques

Le point d'entrée de l'expression est évidemment l'expression `{squelette}`. Les lignes 1 et 2 classent les lettres entre voyelles et consonnes. La ligne 3 déclare l'expression `{groupe_cons}` qui représente un groupe d'une ou plusieurs consonnes à la suite, sans voyelles.

Les lignes 7, 8 et 9 déclarent trois types de squelettes consonantiques : parmi les squelettes contenant des voyelles aux extrémités (20% – ligne 10, correspondant au type 1), 70% débutent par une ou plusieurs voyelles, 7% terminent par une ou plusieurs voyelles et 23% débutent et terminent par une ou plusieurs voyelles.

La possibilité est enfin laissée au scripteur de ne pas composer de mot complet sous la forme de squelette. En effet, il est fréquent que seule une partie du mot soit abrégée, comme dans « *bjour* ». Les lignes 4, 5 et 6 autorisent l'insertion de groupe(s) de une ou deux voyelles dans un squelette consonantique. Ce sont les mots de *type 2* dans l'expression. Ici les probabilités ont été calibrées non pas à partir d'un corpus mais à l'usage, pour obtenir les meilleurs résultats. Ces choix ont été affinés par pallier de 10 points à partir d'une série de mesures.

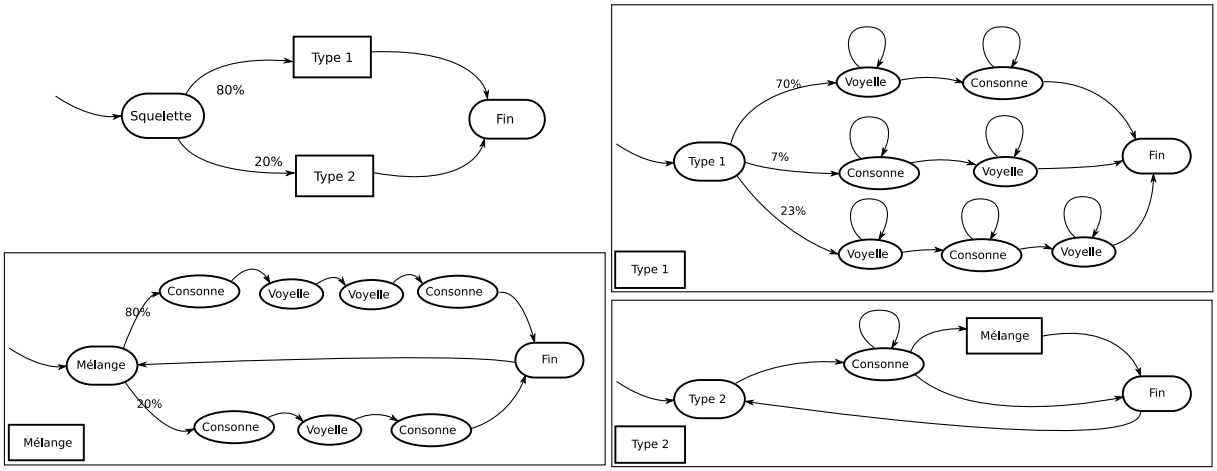

FIG. 6.1 – L'automate correspondant à l'expression régulière du tableau 6.1

Résultats

Nous avons maintenant à notre disposition deux nouvelles ressources linguistiques : un lexique de squelette consonantique et une expression régulière modélisant les squelettes consonantiques.

La reconnaissance de l'écriture est effectuée sur le corpus de 54 squelettes précédemment extraits. Les résultats sont présentés en table 6.2.

	Sans	*lk-free*	*lk-text*
Sans (car)	$142/151 = 94,0\%$	$143/151 = 94,7\%$	$100/151 = 66,2\%$
Sans (mot)	$46/54 = 85,2\%$	$46/54 = 85,2\%$	$8/54 = 14,8\%$
Expr. régulière (car)	$148/151 = 98,0\%$	$147/151 = 97,4\%$	$148/151 = 98,0\%$
Expr. régulière (mot)	$51/54 = 94,4\%$	$50/54 = 92,6\%$	$51/54 = 94,4\%$
Lexique (car)	$128/151 = 84,7\%$	$143/151 = 94,7\%$	$134/151 = 88,7\%$
Lexique (mot)	$41/54 = 75,9\%$	$46/54 = 85,18\%$	$42/54 = 77,7\%$
Expr. + Lexique (car)	$148/151 =$ **98,0%**	$147/151 = 97,4\%$	$148/151 =$ **98,0%**
Expr. + Lexique (mot)	$51/54 =$ **94,4%**	$50/54 = 92,6\%$	$51/54 =$ **94,4%**
Optimale (car)	$151/151 = 100\%$	$143/151 = 94,7\%$	$148/151 = 98,0\%$
Optimale (mot)	$54/54 = 100\%$	$47/54 = 87,0\%$	$52/54 = 96,3\%$

TAB. 6.2 – Résultats de la reconnaissance en fonction des combinaisons de ressources linguistiques

Ces résultats se situent entre la borne inférieure (94,7% au niveau caractère, 85,2% au niveau mot) et la borne supérieure (100% quelque soit le niveau de granularité), comme présenté en 5.16.

Toutefois, peu de conclusions peuvent être tirées de ces résultats, ceci pour plusieurs raisons. Tout d'abord, le corpus de test est très réduit et la simple reconnaissance d'un caractère en plus ou en moins fait énormément varier le taux de reconnaissance. De plus, la construction de l'expression régulière est contestable : elle est basée uniquement sur les

observations de l'expérimentateur et n'est pas validée par un corpus suffisamment large. Il est probable que la forme des squelettes (que l'expérimentateur a trié manuellement) a influencé la construction de l'expression régulière.

Les résultats obtenus avec le lexique uniquement ne sont pas vraiment intéressants : ils atteignent la borne inférieure et n'apportent apparemment pas suffisamment d'informations (en revanche, la reconnaissance avec la combinaison *lk-text* et le lexique donne de bien meilleurs résultats que chaque ressource utilisée séparément). Il faut noter que le lexique est construit à partir d'un corpus de textes écrits assez éloigné du registre des MIMEMA récoltés. Il y a par exemple quinze occurrences du mot « slt » dans le corpus de squelette, mais le mot « salut » à l'origine ne se trouve pas dans le lexique car c'est un mot assez rare dans le contexte de l'écrit. Ainsi, le lexique construit n'est pas suffisamment couvrant dans 12 cas sur 54 (29%) et l'introduction manuelle des formes « slt » et « bjr » change sensiblement les résultats. Combiné avec l'expression régulière, le lexique n'apporte pas plus d'informations : les résultats sont rigoureusement identiques pour les 54 formes proposées, du point de vue du taux de reconnaissance, mais également en comparant les mots résultant de la reconnaissance.

À défaut de disposer de ressources suffisamment volumineuses (plus de MIMEMA ou un corpus de SMS adéquat) nous sommes contraints de nous contenter de ces résultats, qui doivent être pris avec les réserves qui s'imposent.

6.2.3 Traitement des rébus

Nous avons effectué un traitement semblable pour améliorer le résultat de la reconnaissance des rébus. Il ne semblait toutefois pas pertinent de construire un lexique de rébus en raison de la grande créativité possible dans leur construction, mais aussi en raison de l'absence de ressources conséquentes. De plus [ÉGV06] mettent en garde contre l'utilisation d'un dictionnaire dans le cas de ces formes (voir la discussion en 2.1.5).

Expression régulière pour la reconnaissance des rébus

Nous avons toutefois construit une expression régulière à partir des propriétés des rébus, à savoir :

- la possibilité de mélanger des lettres, des chiffres et certains symboles (*a+*, *2m1*...) ;
- la forte présence de certains singletons (*c*, *g*, *9*...) ;
- la faible probabilité d'avoir deux chiffres ou plus à suivre dans un même mot ;

L'expression régulière ainsi construite est présentée dans la table 6.3.

Cette expression, plus complexe que la précédente, est découpée en différentes parties pour traiter les différentes formes de rébus :

- les lignes 1 à 3 classent l'ensemble des symboles (lettre, chiffre, + et -) ;
- les lignes 4 à 8 déclarent des ensembles de singletons très fréquents et à l'inverse, très rares ;
- la ligne 9 déclare les rébus composés uniquement de lettre (deux ou plus – les singletons sont traités avant) ;
- les lignes 10 à 13 présentent les rébus contenant un ou plusieurs chiffres en imposant l'absence de deux chiffres à suivre, en effet ceci ne s'est jamais présenté dans les

Classification des symboles

```
 1  lettre              =  [a-zA-Z]
 2  chiffre             =  [0-9]
 3  symbole             =  [+-]
```

Singletons rares et fréquents

```
 4  lettre_sing_f   49%  =  [cdfgjklmprtvCDFGJKLMPRTV]
 5  lettre_sing_r    1%  =  [abehunoqsuwxyzèéàôîABEHUNOQSUWXYZÈÉÀÎÔ]
 6  chiffre_sing_f  49%  =  [12679]
 7  chiffre_sing_r   1%  =  [34580]

 8  singleton       45%  =  ({lettre_sing_f}|{lettre_sing_r}|
                           {chiffre_sing_f}|{chiffre_sing_r})
```

Mots d'au moins deux lettres, sans chiffre

```
 9  rebus_lettre    10%  =  {lettre}({lettre})+
```

Mélange de lettres et de chiffres, mais pas deux chiffres à la suite

```
10  rebus_chiffre1  20%  =  {lettre}*{chiffre}({lettre}+{chiffre}?{lettre}*)?
11  rebus_chiffre2  40%  =  {chiffre}({lettre})+
12  rebus_chiffre3  40%  =  ({lettre})+{chiffre}

13  rebus_chiffre   45%  =  {rebus_chiffre1}|{rebus_chiffre2}|{rebus_chiffre3}
```

La forme rébus, en combinant les motifs déclarés précédemment

```
14  rebus               =  ({lettre}['])?({rebus_lettre}|
                           {rebus_chiffre}|{singleton})({symbole})?
```

TAB. 6.3 – Expression régulière pour les rébus

rébus que nous avons rencontrés dans cette étude, et apparaît comme fortement improbable. Citons à titre de contre-exemple le rébus « *20* » qui n'est toutefois composé que de chiffre et sera donc traité comme un cardinal ;

– la ligne 14 enfin, combine les différentes formes introduites précédemment pour définir l'expression régulière *rebus*.

Résultats

Les résultats sont consignés dans la table 6.4. Notons que, avant d'effectuer la reconnaissance, nous avons vérifié que l'expression régulière était en mesure de reconnaître chaque label.

Ces résultats ne sont pas satisfaisants en l'état puisque, dans le cas du taux de reconnaissance au niveau mot, nous obtenons des résultats moins bons que sans l'utilisation de ressource (84% sans utiliser de ressource, au mieux 80,9% en utilisant l'expression régulière). Toutefois, comme dans la partie précédente, il est difficile de tirer des conclusions de ces résultats et nous ne pouvons pas affirmer que cette expression régulière n'est pas efficace dans un cadre plus général. À nouveau nous manquons de ressources d'apprentissage

	Sans	*lk-free*	*lk-text*
Sans (car)	$188/204 = 92,2\%$	$189/204 = 92,6\%$	$141/204 = 69,1\%$
Sans (mot)	$79/94 = 84,0\%$	$79/94 = 84,0\%$	$42/94 = 44,7\%$
Expr. régulière (car)	$189/204 = 92,6\%$	$189/204 =$ **92,6%**	$188/204 = 92,1\%$
Expr. régulière (mot)	$76/94 = 80,9\%$	$76/94 =$ **80,9%**	$75/94 = 79,8\%$
Optimale (car)	$193/$***199*** $=$ ***96,9%***	$193/204 = 94,6\%$	$193/204 = 94,6\%$
Optimale (mot)	$91/$***93*** $=$ ***97,8%***	$83/94 = 88,3\%$	$85/94 = 90,4\%$

TAB. 6.4 – Résultats de la reconnaissance en fonction des combinaisons de ressources linguistiques

et de test, ainsi que d'un plus grand controle du SDK, pour pouvoir évaluer objectivement ces résultats.

6.2.4 Traitement des phonétisations

Nous l'avons déjà évoqué, la phonétisation est un phénomène difficile à traiter en l'état puisqu'elle n'a pas de caractéristique morphologique particulière, à l'inverse des rébus qui mélangent lettre et chiffre, ou des squelettes consonantiques qui sont principalement constitués de consonnes. Pour les phonétisations, la seule règle est que la forme, lue à voix haute, soit compréhensible, ce qui ne nous donne aucun indice pour la caractériser globalement.

Toutefois, l'objet de la phonétisation nous renseigne sur le processus de transformation. En effet, un mot est phonétisé pour plusieurs raisons :

- l'**abréviation**, dans le cas des SMS, pour utiliser un minimum de caractère mais aussi pour réduire le nombre de frappe du clavier (par exemple en abrégeant *trop* en *tro* ou *grave* en *grav*) ;
- le **jeu**, en construisant des néographies amusantes comme *bocou* ;
- la **méconnaissance de l'orthographe**, notamment de la conjugaison des verbes, qui va conduire à des simplifications (en transformant par exemple toutes les formes en $[e]$ ou $[\epsilon]$ – ai, ais, ait, é, è... – par la lettre *é*, *j'aurai* devient *j'auré*).

Il ne serait sans doute pas pertinent de construire une expression régulière modélisant les phonétisations comme cela a été fait auparavant. En effet, il n'y a pas un schéma directeur ni une caractéristique morphologique que nous pouvons décrire comme pour les squelettes consonantiques ou les rébus, mais il peut être intéressant de construire un lexique de phonétisations pour *aider* le système de reconnaissance. Ce n'est certainement pas suffisant car il est impossible de construire automatiquement un lexique suffisamment couvrant sans apporter beaucoup plus de bruits que d'information (voir à nouveau la mise en garde de [ÉGV06] à propos du nombre combinatoire de formes déviés, évoquée en partie 2.1.5).

Construction d'un lexique de phonétisations

Nous allons donc nous attacher à construire un lexique de phonétisations pour tenter d'obtenir les formes les plus évidentes sans prétendre les obtenir toutes, ce qui paraît improbable compte tenu de la nature même de la forme et de son caractère flexible.

À partir du corpus du *Monde* déjà utilisé en partie 6.2.2 mais en conservant cette fois ci la majorité des formes grammaticales (seuls les mots étrangers, les onomatopées et les cardinaux ont été écartés) nous extrayons une liste de mots auxquels nous appliquons les transformations présentées dans la table 6.5. Nous effectuons le même travail sur le corpus de mots-outils utilisés précédemment. Il semble en effet que la phonétisation ne se limite pas qu'à une catégorie grammaticale mais qu'elle peut s'appliquer à tous les mots qui s'y prêtent d'un point de vue morphologique. Cela concerne souvent les verbes conjugués mais ne s'y limite pas. Ces formes transformées sont intégrées à un lexique pour former une ressource linguistique pour *MyScript*.

La table 6.5 mérite quelques éclaircissements. Tout d'abord, en ce qui concerne les symboles utilisés pour les expressions régulières : ^ représente le début de la ligne, sauf s'il est situé à l'intérieur de crochet comme dans `c([^hei]`, où il indique la négation. L'expression `{1}` indique que le premier bloc entre parenthèse est rétablis sans modification. La règle `/^([dst])es$/ → ['{1}é', '{1}è']` indique que, si une expression commençant par un "d", un "s" ou un "t" est rencontrée, suivie de "es" alors "es" doit être remplacé par "é" et "è" mais la première lettre doit être conservée. Le symbole "$" indique la fin de la ligne (ici, la fin du mot, le fichier étant constitué d'un mot par ligne) et les crochets indiquent un intervalle ou un choix : [abc] représente "a" ou "b" ou "c".

En ce qui concerne la suppression des "e" muets en fin de mot, le traitement n'est pas trivial et nécessite une vaste connaissance linguistique pour être modélisé parfaitement. Nous proposons ici une modélisation simplifiée :

- Lorsque le "e" est situé après une voyelle, en fin de mot, il est supprimé (comme dans "joue", "boue", "joie") ;
- lorsque le "e" est situé après une consonne, il est généralement supprimé sauf :
  - si la lettre précédant la consonne est une voyelle (comme *française*, *brise*...) ;
  - si la lettre précédant la consonne est un "n" ou un "m", indiquant une voyelle nasalisée (comme *danse* ou *pompe*...) ;
- Si la consonne avant le "e" final est "k", "m", "b", "v", ou "l" alors le "e" sera supprimé quand même (comme *homme*, *époke* ou *grave*...) ;
- Si la consonne avant le "e" final est un "r", alors le "e" sera supprimé si la voyelle devant le "r" n'est pas un "e" (car la combinaison "er" ne se prononce pas de la même façon que "ere", comme *piere* – phonétisation de *pierre*) ;
- Si la consonne avant le "e" final est un "s" précédé d'une voyelle, alors le couple "se" est remplacé par "z" (*cause* devient *cauz*).

Nous appliquons ces transformations sur les mots les plus fréquents du corpus et générons *l'ensemble des combinaisons* de transformations possibles pour un mot. La table 6.6 présente quelques exemples de mots phonétisés.

Finalement, 1202 mots sont retenus donnant 3171 formes transformées.

Résultats

Le lexique ainsi construit est loin de couvrir l'ensemble des échantillons d'encre numérique du corpus de phonétisations puisque seules 55% des formes y sont présentes. Ceci s'explique encore une fois par la différence de registre entre le corpus du *Monde* et le langage utilisé pour les communications par SMS, mais aussi bien sûr par la grande créativité

Expression régulière → Transformation(s)
Suppression des "e" muets en fin de mots
Après une voyelle
`/([aeiouyéèàêîâô])(e)$/ → ['{1}']`
Après les consonnes, ce n'est pas automatique (ex : françaisE)
`/([bcdfghjklpqrstvwxz] [bcdfghjklmnpqrstvwxz])e$/ → ['{1}']`
`/([aiou]r)e$/ → ['{1}']`
`/([kmbvl])e$/ → ['{1}']`
[voyelle]se → [voyelle]z
`/([aeiouy])se$/ → ['{1}z']`
Retrait des consonnes muettes en fin de mot
`/[tsdp]$/ → [”]`
Transformations en milieu de mot :
Suppression des doubles consonnes
`/ll/ → ["l"]`
`/mm/ → ["m"]`
`/nn/ → ["n"]`
`/pp/ → ["p"]`
`/rr/ → ["r"]`
`/ff/ → ["f"]`
Retrait des "h" lorsqu'ils ne sont pas combinés avec c, p ou s
`/([^pcs])h/ → ['{1}']`
Remplacement de "qu" par "k"
`/qu/ → ["k"]`
Remplacement des "c" par "k" lorsqu'ils ne sont pas devant un "e", un "h" ou un "i"
`/c([^hei])/ → ['k{1}']`
"au" → "o"
`/(e)?au(x)?/ → ["o"]`
"oi" → "oa"
`/oi/ → ["oa"]`
[voyelle] s [voyelle] → [voyelle] z [voyelle]
`/([aeiouy])s([aeiouy])/ → ['{1}z{2}']`
ai, ais, é, è → é, è
`/ai|é|è|ais$\|ait$/ → ["é", "è"]`
Retrait des signes diacritiques
`/ç/ → ["c"]`
`/î/ → ["i"]`
`... ...`
Traitement des exceptions
tes, ses, des → sé, sè, té, tè, dé, dè
`/^([dst])es$/ → ['{1}é', '{1}è']`
est → é, è
`/^est$/ → ["é","è"]`

TAB. 6.5 – Liste des transformations pour la phonétisation

autres	autre otres autr otre otr
âge	age
nombreuses	nombreuzes nombreuse nombreuze nombreuz
robert	rober
combat	comba kombat komba
tous	tou
raisons	raizons raison résons rèsons raizon rézons rèzons réson rèson rézon rèzon
aujourd'hui	aujourd'ui ojourd'hui ojourd'ui
quoi	koi quoa koa
cause	kause cose kose koz coz
musique	muzique musiqu musike muziqu muzike musik muzik

TAB. 6.6 – Quelques exemples de phonétisations de mots fréquents dans le corpus du *Monde*

des scripteurs qui ne limitent pas la transformation aux quelques règles que nous avons proposées.

	Sans	*lk-free*	*lk-text*
Sans (car)	$300/327 = 91,7\%$	$308/327 = 94,1\%$	$246/327 = 75,2\%$
Sans (mot)	$65/91 = 71,4\%$	$72/91 = 79,1\%$	$22/91 = 24,1\%$
Lexique (car)	$256/327 = 78,3\%$	$308/327 =$ **94,1%**	$296/327 = 90,5\%$
Lexique (mot)	$54/91 = 59,3\%$	$73/91 =$ **80,2%**	$64/91 = 70,3\%$
Optimale (car)	$325/327 = 99,3\%$	$308/327 = 94,1\%$	$324/327 = 99,0\%$
Optimale (mot)	$89/91 = 97,8\%$	$73/91 = 80,2\%$	$88/91 = 96,7\%$

TAB. 6.7 – Résultats de la reconnaissance en fonction des combinaisons de ressources linguistiques

Nous pouvons constater une légère amélioration de la reconnaissance puisque, grâce au lexique, nous parvenons à reconnaître un mot en plus (73/91 avec *lk-free* + lexique) que la meilleure combinaison sans lexique (72/91 avec *lk-free* seule). En revanche, le lexique augmente considérablement le taux de reconnaissance dans le cas des combinaisons avec la ressource *lk-text* puisque le taux passe de 22 à 64 mots sur 91 reconnus correctement, mais ce résultat reste inférieur au taux de reconnaissance avec la ressource *lk-free* seule.

Ces résultats sont toutefois encourageants et invitent à refaire cette expérience en utilisant comme socle un corpus de textes dans un registre plus proche des SMS que ne l'est le corpus du *Monde* qui, comme dans le cas des squelettes consonantiques, apporte certainement beaucoup de bruit en intégrant de nombreux mots très spécifiques au journalisme, mais qui ne sont pas utilisés dans les SMS.

Chapitre 7

Conclusions & perspectives

Nous avons réfléchi dans cette étude au fonctionnement d'un système de reconnaissance de l'écriture *en-ligne* dans le cadre d'un langage libre et par essence très créatif, rendant difficile toute approche rigide basée sur une connaissance du langage, par exemple sur la morphologie des mots. Une large partie du travail a consisté à prendre en main les outils et les ressources à notre disposition, mais aussi à proposer une synthèse des (rares) publications concernants les Nouvelles Formes de Communication Écrite. Nous avons finalement proposé quelques idées pour adapter la reconnaissance de l'écriture au « langage SMS » en se concentrant sur certaines formes productives reconnues. Nous n'avons toutefois fait qu'effleurer le sujet du traitement automatique des Nouvelles Formes de Communication Écrite, tant il est vaste et nouveau dans le cadre du traitement automatique du langage.

Bien que certains résultats soient encourageants, il est important de conserver quelques réserves étant donnée la difficulté de valider ou d'invalider ces résultats en raison de la rareté des ressources linguistiques à notre disposition. Nous regrettons en particulier l'absence d'un corpus de SMS français conséquent qui aurait pu nous donner de précieux indices sur les traitements à effectuer et nous aurait permis de mesurer l'étendue des phénomènes dans l'usage courant : en effet, une rapide analyse de corpus anglophones[1] révèle que les déviances sur lesquelles nous nous sommes concentrés ici sont tout à fait marginales et ne sont pas caractéristiques du medium pour ce langage.

La suite de ce travail passera donc nécessairement d'abord par une nouvelle collecte de Mini-Messages Manuscrits (voir l'enquête proposée en annexe) dans le but d'abord de les positionner par rapport aux autres NFCE – rappelons que l'étude des SMS manuscrits n'était qu'une simplification du problème et il y a fort à parier que les MIMEMA n'auront pas la forme des SMS – mais aussi bien sûr, de pouvoir les étudier plus largement et plus efficacement.

Il faut toutefois noter que, en l'état, le système est déjà très performant dans le cadre de l'écriture pré-casée puisque les premières mesures montrent que, sans adaptation, 95% des caractères sont bien reconnus (voir table 5.11), ce qui est très correct et tout à fait utilisable[2] à condition que l'utilisateur soit attentif à la qualité de son écriture et que des

[1] Min-Yen Kan de l'Université de Singapour nous a gracieusement fourni un corpus de SMS utilisé dans l'étude de l'article [HK], voir `http://www.comp.nus.edu.sg/~rpnlpir/downloads/corpora/smsCorpus/`

[2] et ce, d'autant plus qu'il est possible d'améliorer la reconnaissance en adaptant le système à l'écriture – à

systèmes de corrections simples des erreurs soient intégrés[3]. Notons que ce résultat se rapproche du taux de reconnaissance atteint dans le cadre d'une écriture libre en français « correct » (91%, voir table 5.9).

Ces résultats peuvent être améliorés, notamment à partir des travaux présentés ici, mais l'intégration est loin d'être évidente. L'utilisation des différentes ressources construites en partie 6 pour reconnaître les MIMEMA non catégorisés donne de moins bons résultats que le système sans adaptation. Rappelons en effet que nous avons d'abord trié les formes pour les traiter et que, appliquer l'expression régulière construite dans le cas des squelettes consonantiques sur des phrases qui n'en contiennent que peu ne fait qu'ajouter du bruit et diminue conséquemment le taux de reconnaissance. Une intégration « naïve » des ressources utilisées pour les squelettes, les rébus et les phonétisations au système global n'est pas efficace, au contraire[4].

Ce n'est pas un constat d'échec. La densité, la complexité des phénomènes étudiés et la nouveauté de ce sujet d'étude invite à appronfondir la reflexion et il sera nécessaire de travailler à cette intégration. Deux solutions se présentent. La première consiste simplement à prioriser certaines ressources par rapport aux autres : il peut sembler étonnant que l'expression régulière pour les squelettes consonantiques (rares) ait la même priorité dans le traitement que le lexique de français « correct » intégré à la ressource *lk-text*. En calquant la priorité d'une ressource à la fréquence d'apparition de la forme reconnue il sera sans doute possible d'améliorer sensiblement les résultats.

Cela reste assez naïf, une meilleure solution serait d'être capable de détecter la forme à reconnaître pour y appliquer le traitement correct. Nous retombons alors sur un problème évoqué dans le chapitre 3 : il faut connaître pour reconnaître et dans ce cas, il faudra reconnaître pour connaître. Ces deux solutions nécessitent un accès plus profond au système de reconnaissance qui n'était malheureusement pas à notre portée, n'ayant accès qu'à l'environnement de développement et pas aux sources du logiciel ni des ressources linguistiques proposées par la société *Vision Object*.

Ajoutons que ce problème de la reconnaissance des différentes déviances est un enjeu bien plus grand que la simple reconnaissance de l'écriture manuscrite puisqu'il oriente grandement le traitement à effectuer. Ainsi donc, le traitement automatique du langage sur les NFCE ne pourra certainement plus se contenter des méthodes éprouvées qu'il a l'habitude d'appliquer, mais ne pourra pas faire non plus le choix de les ignorer tant ces *formes scripturales langagières* [Ani02] tendent à se répandre.

---

la forme mais aussi au vocabulaire – du scripteur, adaptation que nous n'avons pas été en mesure d'utiliser ici mais qui est très prometteuse

[3]La version de démonstration que nous avons pu manipuler intégrait la possibilité de corriger simplement une lettre erronée en la réécrivant, sans avoir à reprendre l'intégralité du message

[4]À une exception près : dans le cadre de l'écriture libre, la combinaison du lexique de phonétisations avec la ressource *lk-text* donne les meilleurs résultats avec un taux de reconnaissance de 53,4% au niveau mot, à comparer avec le taux de reconnaissance sans le lexique, qui est de 48,6%

Bibliographie

[Ani01] Jacques ANIS : *Parlez vous texto ?* Le Cherche Midi éditeur, 2001.

[Ani02] Jacques ANIS : Communication électronique scripturale et formes langagières : chats et SMS. juin 2002.

[Aur02] Dejond AURELIA : *La cyberl@ngue française*. La renaissance du livre, 2002.

[Bov05] Rémi BOVE : Étude de quelques problèmes de phonétisation dans un système de synthèse de la parole à partir de SMS. *RÉCITAL 2005*, juin 2005.

[Bri04] Marc Le BRIS : *Et vos enfants ne sauront pas lire... ni compter !* Stock, 2004.

[CL98] Jean-Pierre CRETTEZ et Guy LORETTE : Reconnaissance de l'écriture manuscrite. 1998.

[FKP06] Cédrick FAIRON, Jean René KLEIN et Sébastien PAUMIER : Le langage sms : révélateur d'1compétence. 2006.

[HK] Yijue HOW et Min-Yen KAN : Optimizing predictive text entry for short message service on mobile phones.

[MDC99] F. MOURLHON-DALLIES et J.-Y COLIN : Des didascalies sur internet ? 1999.

[MDRRT04] Florence MOURLHON-DALLIES, Florimond RAKOTONOELINA et Sandrine REBOUL-TOURÉ : Les discours de l'internet, quels enjeux pour la recherche ? *In* Florence MOURLHON-DALLIES, Florimond RAKOTONOELINA et Sandrine REBOUL-TOURÉ, éditeurs : *Les discours de l'internet, nouveaux corpus, nouveaux modèles ?*, les Carnets du Cediscor. Presse Sorbonne Nouvelle, Paris, 2004.

[Per05] Freddy PERRAUD : *Modélisation du Langage Naturel Appliquée à la Reconnaissance de l'Écriture Manuscrite En-Ligne*. Thèse de doctorat, Université de Nantes, dec 2005.

[Pié03] Isabelle PIÉROZAK : Le français tchaté, un objet à géométrie variable ? *Langage & Société*, 104:123–144, 2003.

[VGM05] Christian VIARD-GAUDIN et Emmanuel MORIN : Projet ATLANSTIC - MIMEMA (Mini Message Manuscrits). dec 2005.

[Vis06] VISION OBJECT : *MyScript Builder Help – documentation de MyScript Builder*, 2006.

[ÉGV04] Émilie GUIMIER DE NEEF et Jean VÉRONIS : 1 pw1 sr la kestion. Le traitement automatique des nouvelles formes de communication écrite (e-mails, forums, chats, SMS, etc.). Journée d'étude de l'ATALA, 2004.

[ÉGV06] Émilie GUIMIER DE NEEF et Jean VÉRONIS : Le traitement des nouvelles formes de communication écrite. *In* Gérard SABAH, éditeur : *Compréhension des langues et interaction*, Cognition et Traitement de l'Information. Lavoisier, Paris, 2006.

Liste des tableaux

Table des figures

Annexe A

Glossaire

Écriture structurée Une écriture est dite *structurée* si des contraintes sont imposées au scripteur, par exemple en lui proposant des cases dans lesquelles effectuer une saisie lettre par lettre, ou en lui demandant de lever le stylo entre chaque symbole, signes diacritiques compris.

Écriture libre Une écriture est dite *libre* lorsque le scripteur peut écrire de façon naturelle, sans nécessairement délier chaque caractère.

IM *Instant Messenging* : messagerie instantanée. À mi-chemin entre l'IRC et le courrier électronique, les interlocuteurs se rendent disponibles pour discuter, il est alors possible de leur envoyer un message qu'ils reçoivent instantanément (accompagné généralement d'une alerte pour les avertir) auquel ils peuvent choisir de prêter attention ou non. Les discussions se font généralement entre deux interlocuteurs seulement. *MSN* (pour *Micro-Soft Network*) est le client de messagerie instantanée le plus utilisé, ce pourquoi on parle souvent de « MSN » pour parler de IM, comme on parle de *frigidaire* pour un réfrigérateur.

IRC *Internet Relay Chat* : service de discussions instantanées en ligne. Aussi connu sous le nom de *chat* (de l'anglais « bavarder ») et francisé en « tchat »[1] que nous n'utiliserons pas ici, ou en « clavardage ». Les utilisateurs sont connectés à des salons de discussions, lorsque que l'un d'eux parle, tous les autres reçoivent le message instantanément rendant possible des discussions interactives et rapides.

Label D'après le *Trésor de la Langue Française informatisé*[2] :

> ***II. INFORMAT.*** *Groupe de caractères servant à identifier et décrire un article, un enregistrement, un message, un fichier ou un volume d'information (d'apr. Informat. 1972 et LE GARFF 1975).*

Dans notre étude, le *label* correspond au texte clair enregistré dans le fichier d'encre numérique dans le but de le comparer avec le texte reconnu pour évaluer le *taux de reconnaissance*.

[1] ce terme à été introduit lors d'une campagne publicitaire pour M6 net par la chanteuse Tessa Martin, voir : `http://fr.wikipedia.org/wiki/Tessa_Martin`

[2] `http://atilf.atilf.fr/`

NFCE *Nouvelles Formes de Communication Écrites* : formes de communications électroniques comme les SMS, l'IRC, le courrier électronique ou la messagerie instantanée.

Taux de reconnaissance Le *taux de reconnaissance* (ou *précision*) indique la qualité de la reconnaissance en évaluant le nombre de caractères correctement reconnus par rapport au nombre de caractères à reconnaître.

SMS *Short Message Service* : messages courts envoyés généralement à partir d'un téléphone portable vers un autre téléphone portable. Aussi appelé *texto*.

Annexe B

Nouvelle enquête dans le but de récolter des MIMEMA

INSTITUT DE RECHERCHE EN INFORMATIQUE DE NANTES

Reconnaissance de Mini-Messages Manuscrits

Emmanuel Prochasson
(encadré par Emmanuel Morin et Christian Viard-Gaudin)

Résumé

D'un côté les outils de saisie de l'écriture manuscrite (PDA, stylo numérique) se généralisant, il est nécessaire d'améliorer les systèmes de reconnaissance associés. De l'autre, les Nouvelles Formes de Communication Écrites (SMS, IRC, Forums, Weblogs...) s'imposent peu à peu dans tous les vecteurs de communications.
Forte de ce constat, cette étude se place au croisement de ces deux thèmes de recherche et tente d'adapter la reconnaissance de l'écriture *en-ligne* à la reconnaissance des Nouvelles Formes de Communication Écrite.

dun côté lé outils 2 sézi 2 l'écritur manuscrite (PDA, stylo numérik) se jénéralizant y est néceSaire d'améli0ré ls systèm 2 rekonnaissance aSocié. 2 l'autr, le Nouvelle F0rmes 2 komunication Écrites (sms, IRC, Forums, Webl0gs...) s'imposent peu à peu ds ts lé vecteur 2 komunications.
F0rt 2 c constat, 7 étud ce place o croiseman de ses deu thèmes 2 recherche & tent d'adapté la rekonnaiSanse 2 l'écritur *en-lign* à la rekonaissance des nvel F0rmes 2 komunication Écrite.

Mots-clés additionnels et phrases : reconnaissance de l'écriture manuscrite, SMS, Nouvelles Formes de Communication Écrites

LINA, Faculté des Sciences & Techniques de Nantes
2, rue de la Houssinière
B.P. 92208 — F-44322 NANTES CEDEX 3