TALN 2008, Avignon, 9–13 juin 2008

Transcrire les SMS comme on reconnaît la parole

Catherine Kobus[1] François Yvon[2] Géraldine Damnati[1]
(1) Orange Labs / 2, avenue Pierre Marzin, 22300 Lannion
(2) Univ. Paris Sud 11 & LIMSI-CNRS, BP 133, 91403 Orsay Cedex

Résumé. Cet article présente une architecture inspirée des systèmes de reconnaissance vocale pour effectuer une normalisation orthographique de messages en « langage SMS ». Nous décrivons notre système de base, ainsi que diverses évolutions de ce système, qui permettent d'améliorer sensiblement la qualité des normalisations produites.

Abstract. This paper presents a system aiming at normalizing the orthography of SMS messages, using techniques that are commonly used in automatic speech recognition devices. We describe a baseline system and various evolutions, which are shown to improve significantly the quality of the output normalizations.

Mots-clés : SMS, décodage phonétique, modèles de langage, transducteurs finis.

Keywords: SMS, phonetic decoding, language models, finite-state transducers.

1 Introduction

La diffusion des outils de communication électronique (mails, SMS, blogs, forums de discussion, chats, etc) a favorisé l'émergence de nouvelles formes d'écrits (Veronis & Guimier de Neef, 2006). Destinés à des proches ou à des pairs, rédigés dans l'instant, avec des interfaces qui imposent des contraintes nouvelles (claviers d'ordinateurs, d'assistants personnels ou de téléphones portables), ces textes se caractérisent par un net relâchement vis-à-vis de la norme orthographique, ainsi que par de multiples détournements de l'usage conventionnel des caractères alphabétiques, utilisés non seulement pour encoder des formes linguistiques, mais également du méta-discours (citations), des émotions (colère, humour), des attitudes (emphase, dérision) etc. Si chaque média impose des contraintes spécifiques et se caractérise par des modes d'écriture et des codes qui lui sont propres (voir, par exemple, (Torzec *et al.*, 2001) pour les mails, (Falaise, 2005) pour les chats, ou (Anis, 2001; Anis, 2002; Fairon *et al.*, 2006) pour les SMS), ces nouvelles formes de communication écrite partagent de nombreuses similarités. Face à ces textes d'un genre nouveau, il importe de développer de nouveaux outils de traitement automatique, permettant, par exemple, de pouvoir indexer et effectuer des recherches dans des corpus de messages. Dans cette étude, nous nous intéressons plus spécifiquement aux SMS, messages courts rédigés sur les claviers de téléphones portables, qui, nous semble-t-il, condensent à l'extrême les difficultés que posent ces écrits aux systèmes de traitement des langues.

Le « langage SMS » a fait l'objet de plusieurs études linguistiques (Anis, 2001; Anis, 2002; Fairon *et al.*, 2006), qui permettent de cerner ses principales caractéristiques, notamment la très forte variabilité graphique des formes lexicales. Cette variabilité résulte, d'une part, de l'utilisation simultanée de plusieurs systèmes d'encodage : pour dire vite , l'écriture alphabé-

tique usuelle , est en compétition avec une écriture plus phonétique, ainsi qu'avec une écriture « consonantique » (seules subsistent les consonnes), enfin avec une écriture « rébus » (lettres et chiffres encodent la valeur phonétique de leur épellation). Elle découle également d'un style de communication relâché, qui autorise les plus grandes libertés par rapport à la norme orthographique (non-respect des accords, des flexions verbales, etc). En conséquence, du point de vue lexical, ces messages se caractérisent par un très fort taux de mots « hors-vocabulaire », correspondant à des néographismes, ainsi que par une forte augmentation de l'ambiguïté des formes lexicales « attestées ». Restaurer une orthographe normalisée est donc un préalable pour pouvoir leur appliquer d'autres traitements (synthèse vocale, indexation , etc.) ; elle représente également, du fait de la créativité des scripteurs, un sérieux défi.

Les travaux portant explicitement sur la normalisation automatique de SMS sont relativement rares : mentionnons, pour le français, (Guimier de Neef *et al.*, 2007) qui aborde le problème sous l'angle de la correction orthographique et propose une chaîne complète de traitements symboliques pour effectuer cette correction ; (Barthélemy, 2007) est plus prospectif et suggère une modélisation à base d'automates finis permettant de gérer efficacement la concurrence entre divers modes d'écritures. Pour l'anglais, signalons (Aw *et al.*, 2006), qui s'inspire des méthodes utilisées en traduction statistique, ainsi que (Choudhury *et al.*, 2007), dont le système de normalisation utilise des méthodes statistiques de correction d'orthographe.

Le système de normalisation présenté dans cet article propose une approche différente, qui cherche à tirer parti de la proximité, relevée par de nombreux auteurs, entre les formes d'écriture utilisées dans les SMS et la langue orale. Notre hypothèse est que le recensement (par exemple dans un dictionnaire) de l'ensemble des variations orthographiques est voué à l'échec. Il semble comparativement plus aisé de produire une représentation phonémique approximative et ambiguë d'un message, sous la forme d'un ensemble de phonétisations possibles, comme il est commun de le faire en correction orthographique. La reconstruction d'un message normalisé est alors très similaire au décodage phonétique, puisqu'il s'agit de retrouver, dans un treillis phonétique la séquence de mots la plus vraisemblable : il semble alors naturel d'utiliser, pour ce problème, des techniques utilisées en reconnaissance de la parole.

Cet article est organisé comme suit. Dans un premier temps, nous décrivons notre système de base (section 2), avant de présenter, à la section 3, plusieurs évolutions de ce système : amélioration du traitement des mots hors-vocabulaire ; introduction de grammaires locales pour les heures et dates ; acquisition automatisée d'un dictionnaire d'exceptions. La section 4 présente une évaluation des performances du système de base et des évolutions proposées, sur l'analyse desquelles nous nous appuyons pour esquisser quelques perspectives (section 5).

2 Architecture

2.1 Principes généraux

Notre système de normalisation repose sur un principe d'expansion/contraction :
– dans un premier temps, le message est converti en un ensemble de séquences phonétiques représentant toutes les prononciations possibles sous la forme d'un « treillis »[1] de phonèmes.
– la conversion inverse est ensuite calculée : transformation des séquences de phonèmes en

[1]Formellement, il s'agit d'un automate acyclique sur l'alphabet phonétique.

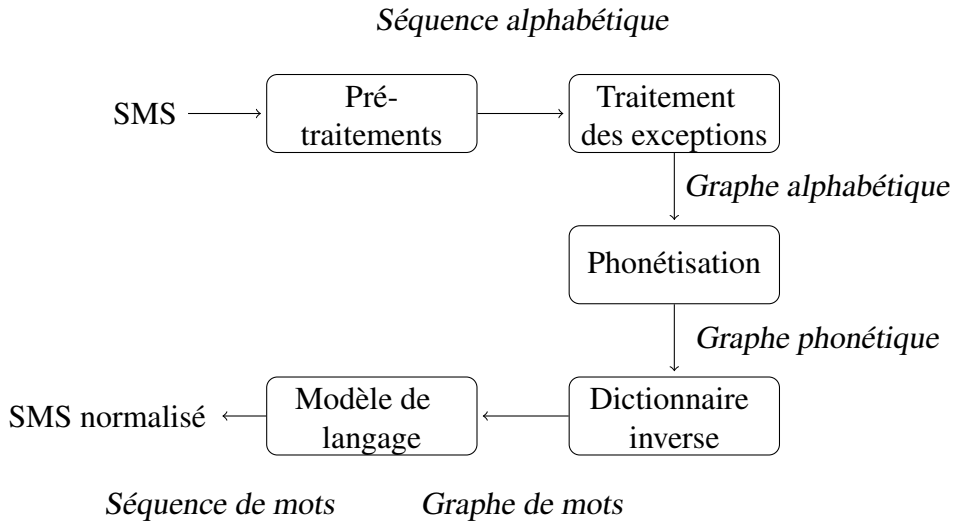

FIG. 1 – Étapes de la normalisation du SMS

séquences de mots par accès dictionnairique, puis sélection, par un modèle de langage statistique, de la meilleure séquence de mots. Cette étape est identique aux calculs effectués dans un système de reconnaissance vocale : a ceci près que dans notre cas, l'incertitude sur les phonèmes et sur les positions des frontières de mots est bien moins grande.

Une vue schématique des traitements réalisés dans le système de base est donnée Figure 1. Après prétraitement du SMS, la normalisation débute par le traitement des exceptions et des abréviations (*"pr"* pour *"pour"* ou *"par"*, *"bcp"* pour *"beaucoup"*, etc), qui sont à la fois très fréquentes en langage SMS et difficiles à modéliser autrement que par construction de listes. Durant cette étape, les mots du message sont analysés et chaque forme trouvée dans le dictionnaire d'exception est mise en compétition avec le ou les expansions associées. Notons que la forme originale est conservée, car elle n'a pas nécesssairement été utilisée comme abréviation. L'expansion des exceptions est donc non-déterministe et produit un treillis de formes.

La troisième étape est la phonétisation, qui utilise des règles de récriture contextuelles *non-déterministes* décrivant les correspondances graphème-phonème. Une règle est formalisée par :

$$\phi\,[a]\,\psi \;\rightarrow\; [b]$$

qui exprime la récriture du motif a en b dans un contexte décrit par les expressions rationnelles ϕ et ψ. Le non-déterminisme de ces règles, c.-à-d. la possibilité que le langage dénoté par b contienne plusieurs mots est inhabituel en transcription graphème-phonème : c'est toutefois un aspect crucial du système, qui assure que l'espace des prononciations possibles est complètement envisagé. Par exemple, la règle de prononciation la plus générale de la lettre 'c' lui associe les quatre prononciations : /k/, /s/, /sɛ/, /se/ : si les deux premières prononciations sont attendues, les deux suivantes expriment la possibilité que cette lettre soit utilisée phonétiquement (et doive donc être « épelée »). Les exceptions détectées lors de la première étape subissent ici un traitement particulier : dans la mesure où ces formes sont déjà normalisées, la phonétisation s'applique de façon déterministe, par accès à un dictionnaire de prononciation.

La suite du traitement utilise les ressources suivantes :

– un dictionnaire de prononciation, utilisé pour convertir des séquences de phonèmes en séquences de mots ;
– un modèle de langage statistique, qui permet d'ordonner par probabilité croissante les séquences de mots ;

L'accès au dictionnaire permet de dégager, à partir du graphe de phonèmes, l'ensemble des séquences de mots possibles ; le modèle de langage permet de pondérer l'ensemble des hypothèses de phrases possibles ; enfin, un algorithme de programmation dynamique sélectionne la séquence de mots la plus probable.

2.2 Implémentation

Chacun de ces modules peut être implanté par des automates ou transducteurs finis éventuellement pondérés. C'est le cas des deux dictionnaires décrits dans la section précédente : le dictionnaire d'exception réalise une transduction de séquences orthographiques en séquences de phonèmes (transducteur E), l'inverse du dictionnaire de prononciation (transducteur D) associe des séquences de mots à des séquences de phonèmes. C'est encore le cas du module appliquant des règles de phonétisation contextuelles (Kaplan & Kay, 1994; Mohri *et al.*, 1996), qui sont globalement compilées en un transducteur R, ainsi que du modèle de langage de type n-gramme, représenté par un accepteur pondéré L. Ces transducteurs sont construits , pour les trois premiers, par des scripts ad-hoc et par les outils de la suite GRM (Allauzen *et al.*, 2005) pour le modèle de langage. Une fois le message en entrée converti en un automate fini M par le module de prétraitement[2], l'ensemble des récritures réalisant la normalisation est prise en charge par les opérations suivantes :

– construction de l'ensemble des séquences de mots possibles pour M, pondérées par leur probabilité pour le modèle de langage. Cette opération est réalisée par composition des différents transducteurs : $T = M \circ E \circ R \circ D \circ L$, dont on ne conserve par projection que le langage de sortie $\Pi_2(T)$.
– recherche de la séquence de probabilité maximale dans $\Pi_2(T)$ par un algorithme calculant des plus courts chemins dans un graphe valué.

Il est possible d'optimiser ce traitement en précalculant $E \circ R \circ D \circ L$, ainsi qu'en optimisant (par déterminisation[3] et minimisation) préalablement $D \circ L$ selon des procédés usuellement utilisés en reconnaissance vocale. L'ensemble de ces opérations est réalisée par les outils de la suite de manipulation de transducteurs finis FSM (Mohri *et al.*, 2000).

Le passage par une représentation phonétique comporte un avantage supplémentaire : dans l'optique d'une vocalisation des SMS, il permet de produire sans calcul supplémentaire non seulement la forme orthographique normalisée, mais également la forme phonétique associée à cette normalisation.

2.3 Le traitement des frontières de mots

L'architecture décrite ci-dessus permet de traiter simplement la question des frontière de mots. Il est courant de trouver dans les messages des formes agglutinées telles que :

[2]Ce module accomplit également certaines opérations de normalisation : traitement rudimentaire des chiffres, insertion de marques de débuts et de fin de phrase, etc.

[3]Comme il est usuel en reconnaissance vocale, la déterminisation est réalisée en traitant le mot de longueur nulle ε comme un symbole à part entière.

(1) *Kestu fe ?*

(2) *... avec lbac blanc ...*

(3) *g ésayé 2tapelé pl1 2foi* (exemple tiré de (Guimier de Neef *et al.*, 2007))

Ces exemples sont notoirement difficiles à traiter par des systèmes symboliques (Guimier de Neef *et al.*, 2007). Pour autant, les messages à normaliser sont partiellement segmentés (espaces, ponctuations) ; cette information est relativement fiable et doit être utilisée. Notre architecture

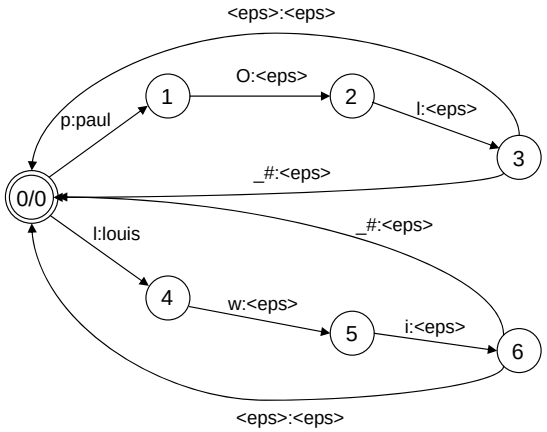

FIG. 2 – Gestion des frontières de mots dans le dictionnaire inverse de prononciation

permet à la fois d'utiliser les informations de segmentation disponibles, en tout autorisant l'insertion de nouvelles frontières de mots. Ceci est réalisé par la procédure de construction du transducteur représentant le dictionnaire inverse de prononciation D. Ce transducteur possède l'allure d'un arbre des préfixes, chaque branche correspondant à une séquence de phonèmes le long de laquelle le mot orthographique correspondant est émis. Deux transitions « rebouclent » sur l'état initial : l'une est étiquetée par le symbole /_#/, qui représente un séparateur explicite ; l'autre est une transition ε, qui permet de démarrer la reconnaissance d'un nouveau mot alors même qu'aucun séparateur ne figure dans l'entrée. En pondérant différentiellement ces deux transitions, on exprime une plus ou moins grande préférence envers une segmentation qui respecterait les séparateurs originaux. Ce mécanisme est illustré à la figure 2, qui représente un dictionnaire contenant les deux mots *louis* et *paul*. Deux transitions bouclent sur l'état 0 à partir de l'état 6 (fin de *louis*) : l'un est une transition ε. L'emprunter signifie qu'on introduit une frontière de mot qui est absente du message original ; l'autre est étiquetée $\_\#$: elle est utilisée si l'on rencontre un séparateur dans le message.

En revanche, il n'est pas possible, dans ce schéma, que deux mots soient « recollés » : tout séparateur présent dans l'entrée sera également présent dans la sortie. Ceci rend notre système incapable de traiter correctement des entrées telles que *"je ne pep a mpaC dtoi"* ou encore *"slt le zami"* dans lequel des formes sont incorrectement segmentées.

3 Évolutions du système

3.1 Le système *baseline*

Le système tel décrit ci-dessus (cf section 2) correspond à notre système *baseline*. Son lexique contient de plus de 23000 mots. Nous avons également utilisé un dictionnaire de plus de 900 exceptions, ainsi qu'un ensemble de 140 règles de phonétisation contextuelles. Les contextes

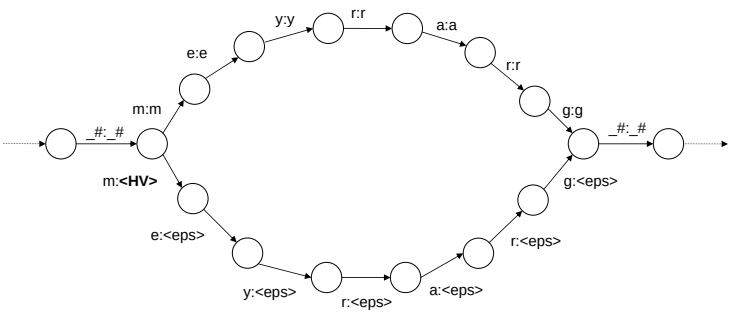

FIG. 3 – Gestion des mots hors-vocabulaire

des règles concernent principalement les débuts ou les fins de mots et aident à décrire la prononciation des finales muettes (comme *'t'*, *'s'*, *'p'*, etc.). Pour la lettre *'p'*, nous avons ainsi deux règles contextuelles distinctes : la première traite le cas d'une fin de mot (lettre muette autorisée, symbolisée par ϵ) ; la seconde règle s'applique aux autres contextes.

$$p \rightarrow \text{/p/ | /pe/ | /pɛ/ } |\epsilon \quad \text{si fin de mot,} \qquad p \rightarrow \text{/p/ | /pe/ | /pɛ/} \quad \text{sinon}$$

3.2 Traitement des mots hors-vocabulaire

Comme dans un système de reconnaissance vocale, le lexique de l'application de normalisation de SMS est fini. Avec le système *baseline*, les mots du hors-vocabulaire (HV) du SMS ne sont pas correctement traités. Dans la mesure où ils ne peuvent être restitués tels quels en sortie du système, ils sont resegmentés en mots phonétiquement proches : ainsi, *"puiske té a meyrarg"* (*"meyrarg"* est HV) produit *"puisque t' es a mis rare"*). Pour y remédier, le module de pré-traitement a été complété de façon à produire une hypothèse supplémentaire, correspondant à la recopie du mot HV dans la sortie : ces mots, qui sont potentiellement corrects, peuvent alors figurer dans la meilleure solution. La figure 3 détaille la façon dont sont gérés les mots HV dans le formalisme des FSMs, en l'illustrant sur la forme *« meyrarg"*. Lors du pré-traitement, *"meyrarg"* est reconnu comme mot HV : deux chemins alternatifs sont alors créés. Le premier segmente l'entrée en graphèmes élémentaires, qui seront phonétisés. Le second chemin est identique, à l'insertion près d'une balise `<HV>`. Cette balise rend transparentes les étapes de phonétisation et d'accès au dictionnaire. La séquence graphémique figurera dans l'ensemble des hypothèses de séquences de mots et pourra être sélectionnée par le modèle de langage. Un post-traitement permet de retrouver le mot correspondant initialement à cette balise.

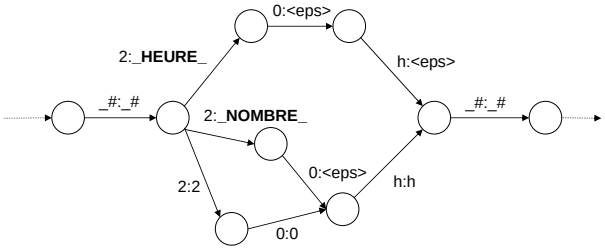

FIG. 4 – Intégration de grammaires locales : traitement de la chaîne *"... 20h ..."*

3.3 Utilisation de grammaires locales

Une seconde amélioration concerne le traitement des heures et des nombres, très nombreux dans le corpus des SMS ; initialement, les chiffres sont traités comme les autres graphèmes. Ainsi, un nombre à deux chiffres est systématiquement segmenté en deux chiffres distincts. Nous avons donc introduit des grammaires régulières, compilées sous la forme de transducteurs finis ; la composition avec le SMS prétraité fournit l'ensemble des analyses possibles des heures et des nombres. Lorsqu'une heure ou un nombre est reconnu, la balise associée est émise ; les étapes de traitement des exceptions, de phonétisation et d'accès au dictionnaire restent identiques. Le modèle de langage est appliqué au graphe de mots et de balises. Ce dernier est appris au préalable sur le même corpus d'apprentissage que précédemment, après étiquetage des nombres et des montants (la phrase *'Je viens à* 20 *h'* devient *'Je viens à* $\_HEURE\_$*'*). La figure 4 illustre, pour cette même entrée, la façon dont sont définies les grammaires locales pour les heures et les nombres dans le formalisme des transducteurs finis. Lors du pré-traitement, les formes sont segmentées en graphèmes élémentaires et mises sous la forme d'un automate. Les grammaires régulières décrivant les heures et les nombres sont également mises sous la forme d'un transducteur. La composition du SMS initial avec ce dernier permet de retrouver toutes les instances d'heures et de nombres dans le SMS initial (Figure 4). Une balise associée à chacune de ces grammaires est émise. Comme pour les mots hors-vocabulaire, ces balises sont transparentes aux étapes de phonétisation et d'accès au dictionnaire. Elles figureront alors dans l'ensemble des normalisations possibles et pourront être sélectionnées par le modèle de langage.

3.4 Apprentissage automatique des exceptions

Le système *baseline* intègre un dictionnaire d'abréviations construit manuellement par analyse de corpus. Dans cette section, nous décrivons une méthode permettant d'apprendre automatiquement les abréviations les plus fréquentes à partir d'un corpus d'apprentissage contenant d'une part, les SMS originaux, pré-traités (suppression de la ponctuation et des majuscules) et leur transcription d'autre part. Cette méthode est basée sur les alignements automatiques et l'extraction de segments bilingues utilisés dans les systèmes de traduction statistique.

Des alignements automatiques sont calculés pour le corpus d'apprentissage à l'aide du logiciel *GIZA++* (Och & Ney, 2003). La technique des *refined alignments* (Koehn *et al.*, 2003) permet de déduire des alignements automatiques croisés une table de traduction, donnant pour chaque segment « source » (en langage SMS) l'ensemble des segments « cible » associés (en français standard). Pour les abréviations, nous ne conservons que les segments "source" constitués d'un seul mot et les segments "cibles" constitués d'au maximum 3 mots (par exemple, l'abréviation

"jtm" alignée avec la séquence *"je t'aime"*). Un score, $s(t, w)$ est enfin estimée pour chaque segment t apparié avec w ; $s(w) = \max_t P(t|w)$ dénote alors le meilleur score d'un segment apparié avec w. Seuls les appariements dont la forme source est suffisamment fréquente (plus de 5 occurrences) et dont le score est supérieur à un certain ratio α (fixé ici à 0.1) du meilleur score ($\{t \mid s(t, w) \geq \alpha s(w)\}$) sont finalement conservés. Ont ainsi été extraites 3264 abréviations/exceptions nouvelles, auxquelles sont associées leurs meilleures expansions. Notons que toutes ces exceptions ne sont pas utiles car il est possible qu'une abréviation et le segment associé aient la même phonétisation.

4 Expériences

4.1 Corpus et Métriques

Les expériences utilisent deux corpus : le premier a été collecté par l'université d'Aix en Provence (Hocq, 2006; Guimier de Neef *et al.*, 2007) ; il est constitué d'environ 9700 messages. Le second corpus est issu d'une collecte organisée en Belgique par l'Université Catholique de Louvain, et comprend 30000 messages (Fairon *et al.*, 2006). Un corpus d'apprentissage `App` de 36704 SMS a été constitué en mélangeant les deux corpus. Les 2998 SMS restants nous ont servi de corpus de test `Test`. Le modèle de langage utilisé dans les évaluations est un modèle de langage 3-gram lissé en utilisant un lissage de type Kneser-Ney et estimé sur le corpus `App`.

Contrairement à (Aw *et al.*, 2006; Guimier de Neef *et al.*, 2007), qui évaluent leurs performances en termes de mesure BLEU (Papineni *et al.*, 2002), nous avons choisi d'évaluer nos systèmes en termes de taux d'erreurs mots ou WER (*Word Error Rate*), métrique qui est également utilisée en reconnaissance vocale. La mesure BLEU, qui s'appuie sur un décompte des n-grams présents dans l'hypothèse et dans une référence, ne vaut que lorsque plusieurs références sont disponibles, comme il est commun en traduction automatique. Pour notre problème, l'ambiguïté dans le choix de la transcription de référence est presque nulle justifiant le calcul de taux d'erreurs par mots et par phrases.

4.2 Résultats

Le tableau 4.2 détaille les résultats obtenus et permet d'apprécier l'impact des améliorations apportées au système. Le système baseline donne un WER de 19.79% ; la majorité des erreurs sont des erreurs de substitution, qui portent souvent sur des mots courts comme *'les'* $\leftrightarrow$ *'le'*, *'j'* $\leftrightarrow$ *'je'*, *'des'* $\leftrightarrow$ *'de'*, etc. Dans une majorité des cas, le mot est pourtant bien orthographié dans le SMS, mais l'étape de phonétisation réintroduit une ambiguïté que le modèle de langage ne parvient pas toujours à compenser. La deuxième ligne du tableau 4.2 montre l'apport du traitement des mots hors-vocabulaire, qui permet de diminuer principalement le nombre d'insertions ; en effet le système baseline avait tendance à segmenter les mots HV en plusieurs petits mots proches phonétiquement et donc à commettre plus d'insertions. Sur l'exemple de *"puisk té a meyrarg ..."*, le système baseline fournit *"puisque t'es à mes ir argh"*, sortie qui est corrigée par le traitement des mots HV.

L'utilisation des grammaires locales (pour les nombres et les heures) améliore globalement les résultats en termes de WER ; moins d'erreurs sont commises sur les nombres. L'introduction

de ces grammaires améliore également la capacité de généralisation du modèle de langage. Cet effort d'introduction de grammaires locales doit donc être poursuivi. Les deux dernières lignes

	WER	Ins.	Sub.	Del.
baseline	19.79%	4.76%	13.44%	1.59%
Traitement des mots HV	18.13%	2.51%	12.83%	2.80%
Utilisation de grammaires	17.58%	2.54%	12.68%	2.35%
Abréviations automatiques	16.96%	2.56%	12.10%	2.30%
Combinaison	16.51%	2.21%	11.94%	2.36%

TAB. 1 – Apport et évaluations des différentes améliorations apportées

du tableau 4.2 chiffrent l'apport de l'apprentissage automatique des exceptions par rapport à l'utilisation d'abréviations collectées manuellement. Les performances sont améliorées significativement en termes de WER, démontrant la validité de l'approche proposée. Les résultats sont encore améliorés en combinant les deux dictionnaires d'abréviations.

5 Bilan et Perspectives

Nous avons présenté, dans cet article, une nouvelle approche pour la normalisation des SMS, basée sur un décodage phonétique. Les différentes évolutions ont permis d'améliorer sensiblement les performances du système *baseline*, qui sont probablement sous-estimées par la métrique WER : de nombreuses erreurs correspondent à des problèmes d'accord, que le modèle de langage échoue à corriger. Ces erreurs sont pourtant sans conséquence dans une perspective de vocalisation car elles correspondent le plus souvent à la perte ou à l'ajout d'un morphème flexionnel « muet ». Ces erreurs sont également bénignes dans une optique d'indexation automatique.

Le système actuel peut toutefois être amélioré de multiples façons :
– les SMS contiennent de nombreuses formes qui sont correctement orthographiées : après phonétisation, cette information est perdue. Une approche qui semble meilleure consiste à chercher celles qui existent dans le dictionnaire D et à les phonétiser par accès direct ; il faudra ensuite exprimer, par des pondérations, que l'on préfère utiliser une suite phonémique extraite du dictionnaire plutôt qu'une suite produite par des règles.
– les règles de conversion graphème-phonème (module E) sont exagérément libérales. Si le non-déterminisme doit être préservé, il importerait de le modérer en pondérant les différentes sorties des règles de récriture : s'il est correct d'autoriser la lettre '*é*' à valoir /e/ ou /ɛ/, il est probable que l'on gagnerait à rendre une des deux options plus probable que l'autre.
– nous avons pour l'instant supprimé toute information liée à la ponctuation, aux majuscules ; cette information pourrait nous être utile pour segmenter le SMS et ainsi améliorer le pouvoir prédictif du modèle de langage.

Remerciements

Les auteurs remercient Émilie Guimier de Neef (Orange Labs) pour avoir mis à disposition la liste d'abréviations ainsi que les différents corpus.

Références

ALLAUZEN C., MOHRI M. & ROARK B. (2005). The design principles and algorithms of a weighted grammar library. *International Journal of Foundations of Computer Science*, **16**(3), 403–421.

ANIS J. (2001). *Parlez-vous texto ? Guide des nouveaux langages du réseau*. Éditions du Cherche Midi.

ANIS J. (2002). Communication électronique scipturale et formes langagières : chats et SMS. Actes des journées "S'écrire avec les outils d'aujourd'hui".

AW A., ZHANG M., XIAO J. & SU J. (2006). A phrase-based statistical model for SMS text normalization. In *Proc. COLING/ACL*, p. 33–40.

BARTHÉLEMY F. (2007). Cunéiforme et SMS : analyse graphémique de systèmes d'écriture hétérogènes. In *Colloque Lexique et grammaire*, Bonifacio.

CHOUDHURY M., SARAF R., JAIN V., SARKAR S. & BASU A. (2007). Investigation and modeling of the structure of texting language. In *Proceedings of the IJCAI Workshop on "Analytics for Noisy Unstructured Text Data"*, p. 63–70, Hyderabad, India.

FAIRON C., KLEIN J. R. & PAUMIER S. (2006). *Le langage SMS*. UCL Presses Universitaires de Louvain.

FALAISE A. (2005). Constitution d'un corpus de français tchaté. In *Actes de TALN*, p. 615–624, Dourdan.

GUIMIER DE NEEF E., DEBEURME A. & PARK J. (2007). TILT correcteur de SMS : évaluation et bilan quantitatif. In *Actes de TALN*, p. 123–132, Toulouse.

HOCQ S. (2006). *Étude des SMS en français : constitution et exploitation d'un corpus aligné SMS-langue standard*. Rapport interne, Université Aix-Marseille.

KAPLAN R. & KAY M. (1994). Regular models of phonological rule systems. *Computational Linguistics*, **20**(3), 331–378.

KOEHN P., OCH F. J. & MARCU D. (2003). Statistical phrase-based translation. In *Proc. NAACL-HLT*, p. 127–133, Edmondton, Canada.

MOHRI M., PEREIRA F. & RILEY M. (1996). An efficient compiler for weighted rewrite rules. In *Proceedings of the annual Meeting of the ACL*, p. 231–238.

MOHRI M., PEREIRA F. & RILEY M. (2000). The design principles of a weighted finite-state transducer library. *Theoretical Computer Science*, **231**, 17–32.

OCH F. J. & NEY H. (2003). A systematic comparison of various statistical alignment models. *Computational Linguistics*, **29**(1), 19–51.

PAPINENI K., ROUKOS S., WARD T. & ZHU W.-J. (2002). Bleu : a method for automatic evaluation of machine translation. In *Proc. ACL*, p. 311–318, Philadelphia, PA.

TORZEC N., MOUDENC T. & EMERARD F. (2001). Prétraitement et analyse linguistique dans le système de synthèse tts cvox : Application à la vocalisation automatique d'e-mails. In *Actes de TALN*, Nancy.

VERONIS J. & GUIMIER DE NEEF E. (2006). Le traitement des nouvelles formes de communication écrite. In G. SABAH, Ed., *Compréhension automatique des langues et interaction*, p. 227–248 : Paris : Hermès Science.