Encodage informatique multilingue : application au contexte du Niger

L'informatisation des langues nationales est une priorité pour de nombreux pays et de nombreuses organisations internationales, en particulier francophones. Un obstacle important à cet objectif est l'encodage informatique des différents alphabets ou plus souvent l'encodage des seuls caractères spéciaux pour les langues utilisant un alphabet basé sur l'alphabet latin. Dans cet article issu d'une étude réalisée au Niger, nous allons analyser ces difficultés et évaluer les mérites de la solution apportée. Dans une première partie, la pratique actuelle de l'encodage des caractères spéciaux sera comparée à la perspective offerte par le standard Unicode, et les moyens de passer de la première à la seconde seront décrits. Puis, après avoir résumé le cadre institutionnel de notre étude, à savoir la formation organisée par le Réseau international francophone d'aménagement linguistique (Rifal) en 2000 au Niger, nous évoquerons les problèmes concrets liés aux caractères propres à chaque langue, et la façon dont les outils employés ont permis d'y répondre. Quelques perspectives sur le développement de banques textuelles multilingues seront données en conclusion.

Termes-clés :
informatisation des langues ; encodage de caractères spéciaux ; Unicode ; langues du Niger (gourmantchéma, haoussa, kanouri, tamachek, zarma).

1 Introduction

On a souvent comparé le passage de la langue orale à la langue écrite avec le passage de la langue écrite à la langue « informatisée » : à chaque passage certaines langues ont su s'imposer mieux que d'autres, permettant à certaines cultures de s'épanouir alors que d'autres sont restées dans l'ombre ou parfois se sont éteintes. Plus particulièrement, l'accès à l'imprimerie et l'accès au support numérique ont aussi été comparés : car que signifie au juste l'informatisation ? Et quels sont les obstacles qui s'y opposent, les problèmes qui doivent être résolus ?

Au-delà des supports sonores ou photographiques, il nous semble que l'informatisation d'une langue peut se définir comme la possibilité de manipuler des documents stockés *sous forme alphabétique, syllabique ou idéographique* sur un support informatique, c'est-à-dire sous la forme d'une suite de caractères qui seule permet l'édition, la recherche, bref la manipulation *analytique* du contenu linguistique. C'est seulement dans une deuxième phase que l'on peut évoquer l'existence d'outils informatiques adaptés à la langue, c'est-à-dire *localisés*, par exemple les outils d'édition qui facilitent la manipulation des documents, et aussi les ressources linguistiques : correcteurs d'orthographe, de grammaire, dictionnaires électroniques. On peut alors penser à la troisième phase, qui est la diffusion d'une langue sur Internet, manifestée par le nombre de documents disponibles et consultés dans cette langue sur le réseau international, y compris les archives électroniques existant dans une langue, témoins de la culture et de la vivacité de la langue respective.

Seule la toute première étape, élémentaire mais incontournable, nous intéressera dans la présente contribution. Nous y étudierons en effet les modalités concrètes à travers lesquelles une langue accède à la représentation informatique, grâce à la *représentation de son alphabet* dans un système informatique, que nous résumerons d'abord (point 2). Nous ferons dans la partie suivante une brève introduction aux possibilités d'encodage informatique des caractères, analysées par ordre d'apparition (point 3.1–3.3) ; nous montrerons ainsi que la plus récente (point 3.3), manifestée par le standard Unicode, apporte une réponse quasi universelle au problème de l'encodage des alphabets, pourvu que ceux qui produisent des documents dans une langue donnée sachent s'y adapter (point 3.4). Nous décrirons alors un outil qui permet de convertir les documents existants vers ce format (point. 3.5). Dans la partie suivante (point 4) nous nous tournerons vers le contexte concret sur lequel se fonde notre expérience, qui est celui des deux formations organisées au Niger par le Rifal (Réseau international francophone d'aménagement linguistique) durant la session 2000-2001. Nous décrirons brièvement les cinq langues nigériennes sur lesquelles nous avons travaillé (point 4.1), puis nous montrerons en quoi les objectifs du Rifal rejoignaient ceux de l'institution nigérienne d'accueil et nous synthétiserons les problèmes soulevés par l'encodage de ces langues (point 4.2 et 4.3). Nous verrons enfin (point 5) comment ces difficultés ont été prises en compte et résolues grâce à l'outil de conversion proposé (point 5.1), avec toutefois des remarques critiques sur l'encodage actuel de ces langues (point 5.2), ouvrant

ainsi des perspectives pour les futures sessions de formation (point 6).

2 Les alphabets humains et l'informatique

Avant d'entrer dans le vif du sujet, une mise en perspective sur la nature des alphabets rencontrés s'impose. Plusieurs systèmes d'écriture sont apparus au cours de l'histoire, la plupart fondés sur un ensemble plus ou moins grand de caractères organisés en phrases sous une forme généralement linéaire. On classifie souvent ces systèmes en : (1) systèmes alphabétiques, fondés sur un petit nombre de signes à valeur souvent phonétique (le plus ancien spécimen étant l'alphabet d'Ougarit) ; (2) systèmes d'écriture syllabiques, tels le syllabaire japonais *kana* ou le syllabaire inuit ; (3) systèmes idéographiques, chronologiquement les plus anciens, exemplifiés aujourd'hui par les idéogrammes chinois. Pour cette dernière variété, le nombre de signes doit être rapproché du nombre de mots plutôt que du nombre de phonèmes ou de syllabes d'une langue, et est donc de l'ordre de plusieurs dizaines de milliers. Les problèmes informatiques posés par l'encodage de ces trois classes de systèmes d'écriture diffèrent donc par leur ordre de grandeur entre les deux premières et la dernière.

La représentation informatique de l'alphabet latin dans sa version anglaise, contemporaine des premiers ordinateurs de série, a précédé de plusieurs décennies la représentation des trois principales écritures idéographiques, celle de la Chine, du Japon et de la Corée, ou *système CJK*. Si l'encodage d'autres écritures idéographiques aujourd'hui disparues nous laisse entrevoir la nécessité de plusieurs autres dizaines de milliers de caractères, il reste que pour les langues contemporaines le champ est plus limité, et, si l'on exclut le système CJK, le problème se réduit à la représentation d'alphabets de plusieurs dizaines de signes.

En outre, et ceci sera notre point principal, l'alphabétisation récente des langues non encore écrites fait le plus souvent appel à l'alphabet latin, enrichi de nombreux caractères phonétiques empruntés à l'Association internationale de phonétique (API). En effet, les différences entre alphabets ont en général une base historique, mais dans la mesure où l'alphabétisation actuelle des langues se fait en coopération avec des experts occidentaux, celle-ci est fondée souvent sur l'alphabet latin. Il est donc fréquent, et ce sera le cas dans notre expérience, que les langues à encoder utilisent l'alphabet latin enrichi de plusieurs « caractères spéciaux ». Mais d'autres alphabets ont pu servir de base pour l'alphabétisation, notamment l'alphabet arabe, utilisé pour le persan, pour le turc (avant 1921) et, actuellement au Niger, pour un dialecte songhay nommé *tagdal*.

La dominance de l'alphabet latin en informatique est visible jusque dans la nature de l'encodage informatique qui donne par défaut la priorité aux lettres latines utilisées en anglais. Rappelons brièvement ici les principes de l'encodage des caractères sur un ordinateur :
– l'information est toujours stockée et manipulée comme une suite de signaux binaires (bits) traditionnellement symbolisés par 0 et 1 ;
– une série de huit bits est appelée octet, avec 256 octets différents possibles, auxquels correspondent les nombres décimaux de 0 à 255 ;
– les caractères sont affichés à l'écran par une interface graphique qui traduit les codes-machine, lus en général octet par octet, vers les formes des lettres à afficher appelées *glyphes ;*
– la correspondance entre les valeurs des octets et les glyphes est définie par une *police de caractères ;*
– on distingue enfin dans la définition d'une police deux éléments : le *jeu de caractères* ou correspondance théorique entre des glyphes et des nombres (0–127 ou 0–255 ou 0–65535), et l'*encodage* concret de ces nombres en machine sous la forme d'une suite de bits.

3 « Deux octets valent mieux qu'un seul » : d'une police locale à une police universelle

La plupart des personnes utilisant un ordinateur se sont heurtées au moins une fois au problème de la saisie ou de l'affichage simultané de caractères correspondant à plusieurs langues. Ces problèmes sont inhérents à l'encodage généralement utilisé pour les caractères, encodage qui a toutefois récemment commencé à évoluer.

3.1 Limites de l'encodage d'un caractère sur un seul octet

Nous avons vu que l'unité traditionnelle de stockage de l'information est une suite de huit bits, appelées octet. Aux premiers temps de l'informatique toutefois, seuls les sept premiers bits étaient porteurs d'information, le huitième servant de contrôle pour repérer les erreurs de transmission. Il n'y avait donc alors que 128 symboles distincts possibles. Une trentaine d'entre eux, les caractères de contrôle, servaient aux commandes de gestion du texte, les autres étant affectés chacun à un caractère alphabétique latin, à un chiffre ou à un caractère de ponctuation – c'était le *standard ASCII*. La libération du huitième bit pour le codage permit d'ajouter 128 autres symboles, dont 96 furent affectés au codage des caractères latins accentués d'Europe de l'Ouest. Ce jeu de 256 caractères fut appelé un certain temps « ASCII étendu », avant d'être décliné dans les différentes variantes du standard ISO-latin, qui diffèrent par les caractères affectés aux codes de 160 à 255 – sans parler des variantes introduites par les différents formats propriétaires dans les systèmes d'exploitation.

Il ne faut pas confondre le jeu de caractères, table de correspondances entre des *nombres* et des *définitions abstraites* de caractères, avec la police, qui contient les définitions de la forme des caractères, ou glyphes. Ainsi les polices standard *Times New Roman* et *Arial*, dans le même jeu ISO-latin-1, afficheront toutes les deux un « Ù » pour le code 217 et un « ù » pour le code 249, mais le style des caractères sera différent (« ù » avec sérif dans un cas et « ù » sans sérif dans l'autre). En revanche, le jeu de caractères ISO-latin-1 contient simplement la définition « *latin capital letter u grave* » pour le code 217, et « *latin small letter u grave* » pour le code 249.

La nécessité de définir plusieurs jeux de caractères à base latine, regroupés dans le standard ISO-latin, procède de l'impossibilité à utiliser simultanément plus de 96 caractères différents (codes de 160 à 255). Ainsi, le jeu ISO-latin-1 ou ISO-8859-1 couvre la plupart des besoins de transcription des langues de l'Europe de l'Ouest : français, espagnol, allemand, basque, etc. (et incidemment des langues d'Afrique comme l'afrikaans, le sangö ou le swahili). Le jeu de caractères ISO-latin-2 ou ISO-8859-2 couvre, lui, les systèmes orthographiques de l'Europe centrale et de l'Est, et les codes 217 et 249 décrits plus haut y correspondent à des caractères différents, respectivement « *latin capital letter u ring* » et « *latin small letter u ring* », donc aux lettres « Ů » et « ů ».

3.2 Jeux de caractères sur un octet et langues « exotiques »

Si la plupart des alphabets des langues à tradition écrite d'Europe peuvent se retrouver dans un jeu de caractères standardisé, c'est loin d'être le cas pour les langues à tradition orale qui accèdent récemment à l'écriture. S'agissant souvent d'une alphabétisation à base latine, certains caractères spéciaux existent parfois mais se trouvent disséminés dans plusieurs jeux. C'est ainsi que l'on est parfois amené à créer des polices spéciales dans lesquelles plusieurs glyphes ne correspondent plus à la définition théorique du code dans un standard donné. Par exemple, dans le jeu standard ISO-latin-1, le code 198 est attribué au caractère « *latin capital letter ae* (donc « Æ »), mais une police *exotique* pourra lui substituer à l'affichage le glyphe « *epsilon* » (donc « ɛ ») pour les besoins de transcription spécifique d'une langue.

La conséquence principale de ces redéfinitions, par ailleurs indispensables, est que les caractères ne sont véritablement définis dans un texte qu'à travers une police réalisant la correspondance entre le code (sur un octet) et le glyphe. Pour qu'un document rédigé sur un ordinateur soit lisible sur une autre machine, il faudra donc que les polices spéciales utilisées dans le document y soient également installées. L'utilisation de polices locales apparaît dans ce cas comme un obstacle à la diffusion des documents dans un contexte multilingue.

D'un point de vue pratique, la Société internationale de linguistique[1] commercialise une base de plus de 1 600 glyphes représentant des caractères phonétiques et des symboles linguistiques, chacun d'entre eux étant identifié par un code à quatre chiffres appelé *SILID*. Le logiciel qui accompagne la base permet d'associer à chaque code d'un jeu de caractères un glyphe de la base, grâce à de simples opérations de *glisser-déposer* de la base vers chacune des 224 cellules d'une grille (256 moins 32 caractères de contrôle réservés aux commandes). Le logiciel permet aussi de

1. www.sil.org.

former des glyphes composites en superposant ou en décalant horizontalement ou verticalement des glyphes de la base (ce qui est utile pour les signes diacritiques). Le résultat de cet assemblage graphique produit un fichier texte de type CST définissant les correspondances entre un code jusqu'à 255 et un glyphe, puis produit également un fichier de police de type TTF (*True Type Font*). L'utilisation de ce logiciel pour créer des polices adaptées aux langues nigériennes sera discutée plus bas (*cf.* 5.2).

3.3 Le standard Unicode ou ISO/IEC-10646 : vers un jeu de caractères universel

Afin de dépasser les limitations décrites, il est nécessaire d'utiliser une table de caractères sur plusieurs octets, au moins deux dans un premier temps. Cette nécessité était déjà manifeste pour les idéogrammes CJK, pour qui un seul octet était clairement insuffisant (256 codes). Le grand mérite de l'effort conjoint du consortium Unicode et de l'Organisation internationale de normalisation (ISO/IEC 10646) a été de proposer un jeu de caractères standard sur deux octets, donc avec 65 536 places, qui fixe un code pour la plupart des caractères alphabétiques existants (dans tous les alphabets), pour de nombreux caractères symboliques (par exemple mathématiques), et naturellement pour des dizaines de milliers d'idéogrammes CJK. Des travaux plus récents visent à étendre à quatre octets ce jeu de caractères.

Complémentaires au jeu de caractères Unicode, plusieurs encodages ont été proposés pour ces caractères sur deux octets, afin d'assurer la compatibilité avec les documents existant déjà (du moins pour les lettres latines) et pour diminuer la taille des documents qui n'utilisent pas d'idéogrammes. Le plus répandu de ces encodages est UTF-8, qui utilise un encodage de longueur variable pour les codes Unicode (parfois plus que deux octets, mais souvent moins) : les 127 premiers caractères (ASCII) sont encodés tels quels sur un octet, la plupart des caractères non-idéographiques sont encodés sur deux octets, et le reste d'Unicode sur trois octets. UTF-8 répond donc aux exigences précédentes, et grâce à une certaine redondance permet l'auto-synchronisation des caractères, c'est-à-dire qu'il permet à un logiciel de repérer dans le flux des données le début du code d'un caractère.

Le standard Unicode, affectant donc un code unique à chaque caractère nécessaire aux transcriptions des langues du monde, permet aux logiciels qui le comprennent de générer des documents lisibles sur tout autre ordinateur sans ambiguïté liée aux polices, car il n'est plus nécessaire d'utiliser des polices spéciales. Il faut toutefois disposer de polices implémentant le standard Unicode, qui ne différent donc entre elles que par leur style (forme des glyphes) et, plus important, par l'éventail de caractères Unicode qu'elles contiennent. Ainsi, une police contenant les idéogrammes CJK sera volumineuse et pourra ralentir le système, sans être particulièrement utile en Europe – il y a donc un compromis à faire entre la taille et la couverture. Citons parmi les polices implémentant plus de mille caractères : *Arial Unicode MS, Cyberbit Bitstream, Lucida Sans Unicode* (la plus compacte). Quant aux logiciels capables de comprendre l'encodage UTF-8 afin d'utiliser ces polices, citons ici les navigateurs Internet, depuis les versions 4 de *Netscape* et d'*Internet Explorer* si on les configure correctement.

À titre d'exemple, le caractère « *latin small letter epsilon* ? « ε » possède le code décimal 603 en Unicode (en hexadécimal : 025B). Ce nombre entier est encodé à travers UTF-8 sur deux octets qui valent C99B en notation hexadécimale, ce qui correspond en notation décimale aux codes 201 et 155 (C9 et 9B). Par conséquent, ce caractère s'afficherait en ISO-latin-1 simplement comme la suite de caractères « É› ». Plus généralement, le texte suivant en ISO-latin-1, « *un plug-in s'installe grâce à un fichier exécutable* », recodé en UTF-8 et lu toujours à travers le jeu ISO-latin-1 apparaîtrait comme suit : « *un plug-in s'installe grÃ¢ce Ã un fichier exÃ©cutable* ». On observe donc que les caractères ASCII n'ont pas changé, puisqu'ils restent encodés sur un octet, mais les caractères propres à ISO-latin-1 reçoivent un encodage différent, plus long, que les logiciels implémentant UTF-8 sont capables de comprendre.

3.4 Représentation des jeux de caractères par les logiciels

La plupart des logiciels d'édition n'affichent pas de façon transparente les codes des caractères, mais demandent à l'utilisateur de choisir une police parmi celles installées

sur le système et convertissent tous les codes en glyphes à travers la police choisie. Le langage HTML (Hyper-Text Markup Language) introduit toutefois une certaine transparence dans la gestion des caractères.

En effet, le langage HTML a été conçu comme un standard d'échange entre différentes plates-formes connectées à Internet. Les machines ne sont donc supposées partager que le jeu des 128 caractères ASCII. Par conséquent, l'encodage en HTML des autres caractères se fera par l'intermédiaire d'*entités* n'utilisant que les codes ASCII. Par exemple, le « é » sera encodé en HTML par l'entité « é ». On voit que cet encodage ne passe pas par une valeur numérique, mais par une description conventionnelle du glyphe lui-même. À ce principe simple s'ajoutent toutefois deux extensions importantes permettant d'afficher des caractères spéciaux. La première est de pouvoir indiquer explicitement le nom de la police avec laquelle un fragment de texte doit être affiché, en encadrant ce fragment entre les balises <FONT FACE=« *Nom de la police* »> et </FONT>. La seconde est de préciser au début du document HTML le jeu de caractères utilisé pour l'ensemble du document, grâce à une balise du type <META HTTP-EQUIV=« Content-Type » CONTENT=« text/html; charset=Nom-du-jeu »/>. Dans les deux cas, on saisit dans le document soit le caractère spécial de code correspondant (dans la mesure où le système d'exploitation et/ou l'éditeur le permettent), soit l'entité HTML correspondant au code, de la forme ⊗ ou ⊗ par exemple (décimal/hexadécimal).

On le voit, le nombre de possibilités est assez important, et en réalité les options dépendent souvent des logiciels qui utilisent le langage HTML, notamment les navigateurs Internet. De surcroît, les paramètres choisis par l'utilisateur d'un navigateur peuvent être prioritaires par rapport à ceux définis par le créateur d'un document. Remarquons enfin que HTML ayant des racines européennes, les navigateurs savent afficher correctement les caractères étendus du jeu de caractères ISO-8859-1 sans passer par les entités ; il est toutefois recommandé de se conformer au standard le plus général.

Pour résumer, il existe deux méthodes principales pour afficher des caractères spéciaux dans un fichier HTML. La première consiste à ne pas indiquer d'encodage en tête du document, ce qui fera qu'il sera lu octet par octet ; dans le document, on signalera les polices à utiliser par des balises <FONT> et on saisira les caractères spéciaux directement par leur code, ou grâce aux entités numériques de type « &#xyz; ». La seconde méthode consiste à réaliser un document « Unicode », c.-à-d. encodé en UTF-8, ce qui sera signalé en tête du document (« CHARSET=utf-8 »), et encore une fois à saisir les caractères spéciaux par leur code (sur un, deux ou trois octets) ou par une entité numérique. Dans les deux cas, il faut que les polices utilisées (« exotiques » dans le premier, Unicode dans le second) soient présentes sur le système sur lequel on consulte le document. Remarquons enfin que la gestion des polices dites *dynamiques* par les navigateurs récents autorise l'incorporation des polices dans un document HTML. Le texte peut alors être lu et imprimé correctement sans nécessiter la présence des polices de départ sur le système. Néanmoins l'absence de standardisation rend cette dernière approche encore aléatoire.

3.5 La conversion de l'encodage des documents, une nécessité pour leur partage

Il apparaît ainsi que le langage HTML est de par sa transparence un format adapté au traitement multilingue, de surcroît un format *non propriétaire*. Si la création de documents directement en HTML peut sembler incommode, puisqu'on peut préférer d'autres programmes d'édition, la conversion à HTML paraît essentielle pour l'*échange* de documents multilingues. Parmi les deux conventions d'encodage décrites ci-dessus, la première est fréquemment utilisée lors de l'exportation vers HTML des documents (par exemple sous *MS Word*), mais seule la seconde permet un échange aisé grâce au standard Unicode. Nous allons décrire ici le logiciel *conv2utf8* (« conversion à UTF-8 »), développé au LLACAN[2], permettant de convertir des documents HTML utilisant des polices vers le format universellement partageable UTF-8. La nécessité de ce logiciel était manifeste, étant donné la quantité de documents saisis sous *MS Word* à l'aide de polices « exotiques » créées avec l'outil de la SIL, et la capacité actuelle de *MS Word* à convertir ou créer des fichiers au format HTML munis de balises <FONT>.

2. *http://llacan.cnrs-bellevue.fr.*

Le principe de fonctionnement de *conv2utf8* est aisé à comprendre sur un exemple. Soit le texte suivant, comportant une partie en bisa (langue d'Afrique de l'Ouest) qui utilise une police exotique (Bisa SILDoulos) et le reste en français avec une police au standard ISO-latin-1 par défaut :

/bà:bá gyıta:rɛ/ Frère aîné du père

Ce fragment de texte sera enregistré en HTML par *MS Word* (en choisissant l'option d'exportation convenable) sous la forme suivante, qui ne contient que des caractères ASCII :

```
<FONT FACE="Bisa SILDoulos">/b&agrave; b&aacute; gy&#165;ta:r&AElig;/
</FONT>Fr&egrave;re a&icirc;n&eacute; du p&egrave;re
```

On remarquera la présence d'entités numérique (¥) et littérales (à) ainsi que la présence de balises <FONT> faisant appel à la police Bisa SILDoulos. Pour qu'un tel texte HTML s'affiche correctement sur un navigateur, il est indispensable que la police Bisa soit installée sur le système, ce qui, on le conçoit, n'est pas toujours le cas.

Le logiciel commence alors par rechercher les polices utilisées dans le texte HTML grâce aux balises <FONT>, puis il parcourt le fichier HTML et remplace les entités numériques ou littérales par les codes UTF-8 correspondants, en tenant compte pour chacune d'elles de la police en cours, et il supprime à cette occasion les balises <FONT> devenues superflues, puisque Unicode les remplace toutes. Le résultat est alors indépendant des polices exotiques (ici Bisa), et sera lisible sur n'importe quel navigateur muni d'une police Unicode pour l'encodage UTF-8 :

/bÃ:bÃ¡ gyÉ©ta:rÉ›/ FrÃ¨re aÃ®nÃ© du pÃ¨re

Naturellement, pour réaliser cette transformation, on doit donner au logiciel une table de correspondance entre les glyphes de la police Bisa et leur code Unicode (le passage d'Unicode à UTF-8 étant connu). Par exemple ici le code 165 qui est le « *yen sign* » (¥) en ISO-latin-1, représente en fait sous la police Bisa le caractère phonétique « *latin small letter iota* ». Pour que le logiciel sache qu'il doit à cet endroit mettre le code UTF-8 du « iota » et non pas celui du « yen », il utilise la table CST ayant servi à créer la police Bisa avec les outils de la SIL. Cette table met en correspondance chaque code réaffecté dans la police, avec le (s) code(s) SILID du/des glyphe(s) installé(s) à cette emplacement. Au code 165 ici correspond le SILID du « iota », à savoir 1012 ; puis, au SILID 1012 correspond par définition le code Unicode 0269 (hexa). Enfin, l'encodage UTF-8 de cette valeur est C9A9, d'où les deux octets « É© », représentés par « É© » (le premier octet « É » vaut C9 ou 201 décimal, le second A9 ou 169 décimal).

Dans sa dernière version, *conv2utf8* accepte également des polices qui n'ont pas été créées par le logiciel de la SIL et donc pour lesquelles il n'y a pas de table CST – ce qui correspond à un cas réel constaté en pratique. Pour cela, le logiciel utilise pour chacune de ces polices un fichier texte analogue à un fichier CST, mais contenant directement les correspondances entre chaque code correspondant à un glyphe exotique, et le(s) code(s) Unicode composant ce caractère.

4 Application aux langues du Niger : la formation organisée par le Rifal

Les principes et le mécanisme que nous venons de décrire n'ont bien sûr de sens que dans la perspective d'une application à un contexte multilingue réel. De fait, les travaux présentés sont étroitement liés au plan de formation du Rifal pour les années 2000 et 2001, et plus particulièrement à la formation que l'un des auteurs a assurée au Niger. En effet, dans ce contexte, les problèmes créés par la manipulation de plusieurs langues nationales, ainsi que de plusieurs polices, rendent manifeste la nécessité d'une solution globale, indépendante des polices locales. Nous décrirons dans cette partie les données de la formation et de l'institut d'accueil, et esquisserons le déroulement de la formation et l'application des techniques décrites. L'évaluation de *conv2utf8* et des polices nigériennes fait l'objet de la partie suivante.

4.1 Le groupe de travail « formation » du Rifal : session 2000-2001

Le Réseau international francophone d'aménagement linguistique visait dans son programme de formation 2000-2001 le soutien au traitement informatique de la langue française et des langues partenaires, en particulier en vue de la constitution de banques de données terminologiques multilingues dans lesquelles le français joue le rôle de langue pivot – et cela conformément à la mission assignée au Rifal par l'Agence intergouvernementale de la Francophonie. Afin de progresser dans cette voie, un des objectifs des sessions de formation 2000 et 2001 était l'implémentation de techniques permettant la diffusion des ressources linguistiques grâce à l'utilisation d'un format universel Unicode/UTF-8 pour le codage des documents, et leur partage grâce à l'utilisation du réseau Internet. Les missions de formation à Niamey présentaient un intérêt particulier dans la mesure où le Niger reconnaît huit langues nationales, à côté du français comme langue officielle, et que malgré cela les efforts d'informatisation des ressources linguistiques sont restés encore limités dans cette partie de l'Afrique sub-saharienne.

4.2 Les langues nationales du Niger : alphabets et encodages

Les six plus importantes langues nationales du Niger ont fait l'objet d'une application lors des formations Rifal. Il peut être utile de situer brièvement ces langues, qui font toutes partie de familles linguistiques différentes, en adoptant ici pour les désigner l'orthographe française[3]. Le *haoussa* est une des langues les plus parlées d'Afrique noire, avec environ 35 millions de locuteurs, dont 5 millions au Niger, soit plus de la moitié de la population du pays. Cette langue est utilisée aussi comme langue d'échange parmi les commerçants d'Afrique de l'Ouest. Le *zarma* est la langue de l'ethnie homonyme qui constitue environ 20 % de la population du Niger (1,5 millions de personnes), et qui est reliée aux Songhay du Mali, dont le royaume aux XIV^{e} et XV^{e} siècles a marqué l'histoire de la région. Le *peul* (appelé aussi *fulfulde*) est une langue très répandue en Afrique de l'Ouest, totalisant environ 15 millions de locuteurs, dont 800 000 environ au Niger. Le nombre de locuteurs du *tamachek* est aussi de 800 000 environ au Niger – il s'agit d'une langue berbère parlée par les Touaregs du Sahara, apparentée par exemple au kabyle d'Algérie. Le *kanouri* est parlé par environ 4 % de la population du Niger, dans le dialecte *manga* (différent du *yerwa* du Nigeria). Enfin, le *gourmantché* est la moins représentée des langues rencontrées au cours de la formation Rifal au Niger, puisqu'elle est parlée par environ 30 000 locuteurs au Niger (mais plus d'un demi million au Burkina Faso voisin). Les trois ou quatre groupes ethniques nigériens restants, chacun avec sa langue, ne dépassent pas 1% de la population chacun. Dans ce qui suit, nous citerons ces langues par ordre alphabétique, en dépit de leur importance ou de leur difficulté d'encodage inégales : gourmantchéma, haoussa, kanouri, tamachek, et zarma.

Parmi les six langues utilisées, le peul, relativement minoritaire au Niger, ne semble pas poser des problèmes d'encodage, puisque le jeu de caractères ISO-latin-1 est suffisant pour représenter son alphabet (deux ou trois caractères supplémentaires sont parfois utilisés). En revanche, les cinq autres langues utilisent à des degrés divers des caractères spéciaux, en général construits à partir de l'alphabet latin, de quatre caractères pour le kanouri à seize caractères pour le tamachek – ce point est peu surprenant puisque le tamachek est une langue sémitique, avec un système de consonnes bien plus riche que celui des langues latines. De façon générale, nos considérations sur les caractères nécessaires à chaque langue, et sur leur disponibilité dans le catalogue de la SIL et/ou dans Unicode, sont fondées sur les informations fournies par les participants à la formation sur leurs propres langues, et sur leur usage de l'écriture – puisque nous ne parlons pas les langues du Niger. Pour certaines langues comme le zarma ou le tamachek, des recommandations officielles existent déjà.

Signalons qu'il ressort de nos recherches que la présence de ces langues sur Internet semble tout à fait occasionnelle – nous ne disposons toutefois pas d'une estimation approfondie. Ainsi, il existe une version de la *Déclaration universelle des droits de l'homme* en kanouri (dialecte yerwa) sur le site des Nations unies : cette version utilise l'encodage UTF-8 mais impose malencontreusement l'usage de la

3. Voir aussi *www.sil.org/ethnologue.*

police *Arial Unicode MS* (à travers une balise <FONT>) ce qui empêche l'utilisation d'autres polices Unicode ! À l'opposé du spectre des possibilités, il existe une version du *Pater* en haoussa donnée simplement par le cliché de la page imprimée correspondante ! On voit donc facilement combien la présence des langues sur Internet est conditionnée par la possibilité d'encoder leurs alphabets.

Afin de manipuler au format informatique des textes en ces différentes langues, les chercheurs de l'Indrap (Institut national de documentation, de recherche et d'animation pédagogiques) utilisent jusqu'à dix polices de caractères (certaines assez semblables), construites en utilisant les outils de la SIL décrits plus haut, grâce au savoir-faire acquis lors des précédentes sessions de formation du Rifal ou du RINT. Malheureusement, la construction et l'utilisation de ces polices n'obéissent pas toujours aux recommandations données, ce qui fait que sur ce point, le mécanisme théorique de conversion à Unicode/UTF-8 se heurte aux difficultés de la pratique, notamment la redéfinition de caractères essentiels, l'incomplétude des polices par langue, etc. – nous étudierons ce problème plus loin.

Nous avons recensé, pour chaque langue, les caractères spéciaux et leurs codes dans chaque police, avec les notations suivantes : A pour « Add », I pour « Indrap98 », L pour « Langues Niger SIL Doulos », N pour « Nigérienne », H pour « Hausa », T pour « TamajaqTT20.3 » (l'Annexe fournit à titre d'exemple la table du zarma). À côté de ces polices, d'autres polices encore sont utilisées. Ainsi, sous deux noms différents, « Niger3 SIL Doulos » et « Alpha3 SIL Doulos », on trouve une police qui ne diffère de L que par cinq caractères, tous utilisés dans les textes nigériens. La police « Manga SIL Doulos » redéfinit seulement quatre caractères utiles au Niger, ceux du kanouri. Enfin, « Hausa Win SIL Doulos » semble être une police inachevée, contenant des redéfinitions souvent paradoxales.

4.3 La mission de l'Indrap de Niamey et les impératifs du multilinguisme

Les formations Rifal 2000 et 2001 ont été accueillies à Niamey par l'Indrap. La mission première de cet institut est l'établissement des programmes d'enseignement pour les écoles nigériennes, accompagnés de nombreux outils pédagogiques : manuels scolaires, guides pour enseignants, méthodes d'évaluation, voire émissions de radio pour enseignants. L'institut est structuré en sections selon les différentes disciplines d'enseignement, chaque section regroupant des locuteurs des différentes langues nationales, souvent en proportion de leur importance pour le pays. La langue de communication et souvent aussi celle des documents de synthèse est le français. La mission de l'Indrap présente de nombreux défis : du point de vue multilingue, l'une des principales difficultés est la traduction dans les langues nationales des termes français propres à chaque discipline, termes qui sont inévitables dans la rédaction des programmes d'enseignement. On note ainsi le rôle central du français et le potentiel terminologique existant pour la recherche en linguistique.

Outre les sections par discipline, on compte une cellule dédiée aux langues nationales, une cellule audiovisuelle et une cellule de saisie et publication assistée par ordinateur. Cette dernière est responsable de la saisie et de la mise en page des ouvrages publiés par l'Indrap, et par conséquent, elle est le principal utilisateur des polices de caractères mentionnées. L'institut est autonome pour ce qui est de la rédaction des manuels grâce à cette cellule, chargée de toutes les opérations allant de la saisie des documents jusqu'à la livraison à l'imprimeur des pages définitives. Signalons que les publications vont et viennent entre les auteurs et la cellule de saisie, sous la forme de versions successives corrigées à chaque fois par l'auteur. L'enrichissement du parc informatique de l'Indrap et le développement d'un réseau local aideraient certainement à diminuer ces allers et retours et donc à réduire l'effort exigé pour chaque publication.

4.4 Déroulement des formations à l'Indrap : textes et résultats

Les participants à la formation ont apporté divers textes dans les quatre langues citées pour servir de matériel aux exercices de conversion de l'encodage, et les polices utilisées ont été installées sur les ordinateurs des différents services. L'objectif ultime était la conversion au format

Unicode/UTF-8, et le partage sur le site Internet du Rifal de différents documents, et l'on peut dire qu'il a été atteint en cinq jours de stage par la plupart des participants, même si la qualité des documents Unicode produits pouvait être nettement améliorée.

Le groupe de documents le plus attendu était un ensemble de cinq lexiques bilingues des mathématiques (haoussa, kanouri, peul/fulfulde, tamachek, zarma); toutefois, ayant été rédigés avec un logiciel qui n'était plus disponible, ces lexiques ont été seulement convertis *a posteriori* par les formateurs. Tels quels, ils pourraient servir de base à des travaux terminologiques plus avancés. Les autres documents traitaient de questions variées : deux textes ont été retenus pour chaque langue, chacun encodé avec une police de départ différente. Il s'agissait de documents officiels (constitution du Niger), de documents pédagogiques (programmes des cours de lecture, de calcul, etc.), de textes littéraires.

Les difficultés rencontrées dans l'application du mécanisme de conversion décrit ont surtout été dues à la nature des polices « exotiques » utilisées, et seront analysées dans la partie suivante. Les formateurs ont dû réagir en temps réel à ces difficultés, et il n'a pas été toujours possible de proposer une solution, puisque celle-ci impliquait à la fois un travail fastidieux (réécriture des tables de correspondance CST) et une modification du programme *conv2utf8* (impossible sur place). En effet, les polices utilisées à l'Indrap ne possédaient plus le fichier de codes CST, et certaines redéfinissaient des codes problématiques.

Toutefois, la formation a pu produire une base multilingue de documents dans les langues du Niger, consultable à l'aide de n'importe quel navigateur utilisant Unicode/UTF-8. Dans la mesure où l'Indrap ne possède pas encore de connexion Internet fiable, le résultat a été hébergé sur le site universitaire de l'un des formateurs[4]. À cette base ont été ajoutés des documents explicatifs. En attendant d'éclaircir les questions de propriété intellectuelle, seul le début de chaque document est accessible sur Internet. Enfin, un cédérom a été gravé avec l'intégralité des documents : une deuxième édition est en cours, qui améliore l'encodage des documents grâce aux perfectionnements vers lesquels nous nous tournons.

4. *www.issco.unige.ch/staff/andrei/formRifal2000/*

5 Évaluation du logiciel de conversion et des polices

5.1 Procédure

Devant le foisonnement de textes, de caractères spéciaux, et de polices utilisées dans le contexte nigérien, il a été nécessaire d'établir une méthode permettant à la fois d'atteindre le but recherché – la conversion à Unicode/UTF-8 des documents – et de tester le programme de conversion muni des tables de correspondance CST. Nombre de polices ne disposant plus des tables CST ayant servi à les créer, la première étape consiste à afficher les caractères de la police, soit à l'aide d'un tableau écrit en langage HTML avec la notation &#XXX; où XXX varie de 0 à 255, soit à l'aide d'un tableau *Excel* (la Table 2 en Annexe montre une possibilité particulièrement explicite). On examine alors les caractères qui diffèrent entre la police étudiée et le jeu ISO-latin-1, et on cherche pour chacun d'eux le code hexadécimal du caractère Unicode correspondant. Pour les signes diacritiques, il faut parfois fusionner deux caractères Unicode, ou plus. On est alors en mesure de reconstruire la table de correspondance CST, en associant à chaque code entre 0 et 255 redéfini dans la police « exotique » le ou les codes Unicode appropriés.

À ce stade, il faut tester la conversion en appliquant le programme *conv2utf8*, accompagné de la table, au tableau HTML (ou au tableau *Excel* converti à HTML) répertoriant l'ensemble des caractères de la police. On compare alors le résultat, visualisé dans un navigateur capable d'afficher les caractères Unicode, avec le tableau de départ, pour s'assurer que tous les caractères sont correctement convertis et affichés à l'écran. Si tel n'est pas le cas, on modifie le fichier CST et on répète le test. Il faut néanmoins prendre en considération certains caractères spéciaux qui ne sont pas rendus par l'un ou l'autre des navigateurs ou des systèmes d'exploitation utilisés, avant de conclure sur la qualité du programme *conv2utf8* ou sur la cohérence des redéfinitions opérées par la police étudiée.

5.2 Utilité de l'expérience de terrain pour l'amélioration de conv2utf8

Si le principe de conversion à Unicode/UTF-8 est relativement simple, la mise en œuvre du logiciel *conv2utf8* en situation réelle a révélé un certain nombre de carences de la première version, ce qui a conduit à la réalisation d'une version améliorée. Il est apparu en effet, outre l'absence des tables CST génératrices des polices, que certains des caractères spéciaux n'étaient pas disponibles tels quels en Unicode mais devaient être composés, que certains caractères de base avaient été redéfinis, etc. C'est pourquoi, dans la deuxième version, les améliorations suivantes ont été réalisées :
– possibilité d'utiliser un table de correspondances directe « octet à Unicode » sans passer par les codes de la SIL, et donc possibilité d'afficher des caractères non prévus par la SIL mais gérés par le système d'exploitation et par Unicode ;
– possibilité d'utiliser les caractères composés : à un octet de la police peuvent correspondre plusieurs codes Unicode à superposer ;
– augmentation du nombre de correspondances « SIL à Unicode » prédéfinies ;
– augmentation du nombre d'entités HTML (du type « é ») reconnues et converties à leur valeur numérique, ainsi que la possibilité d'intercepter des occurrences de certaines entités ayant une signification particulière (au cas où elles auraient été redéfinies par une police locale) ;
– « nettoyage » du code HTML produit par *Microsoft Word 2000*, qui se révèle particulièrement prolixe et par endroits éloigné du standard HTML.

On le voit, l'évaluation en contexte réel a entraîné de nombreuses modifications significatives, et l'on peut à présent affirmer que la nouvelle version pourra tout au plus subir quelques modifications « ergonomiques », ou liées au traitement particulier de tel ou tel caractère par tel ou tel système. La résolution des problèmes posés aux formateurs par les polices locales est à présent facilitée.

5.3 Polices de caractères : entre recommandations et réalité

Le paquetage *SIL Encore Font* largement diffusé auprès des institutions partenaires dans les formations Rifal a permis à celles-ci de se doter de polices adaptées à leurs besoins du moment. Il semble toutefois que les recommandations concernant l'affectation des codes aux caractères spéciaux n'aient pas toujours été bien comprises. En effet, pour des raisons de compatibilité entre les systèmes d'exploitation (*Macintosh vs Windows*), certains codes ne devraient pas être utilisés. Traditionnellement, il est impératif de conserver les 128 premiers caractères (ASCII), communs à toutes les plates-formes. Sur les 128 restants, neuf caractères ne sont pas définis sous *Windows*, seize n'ont pas de correspondance sous *Macintosh*, et seize autres posent des problèmes de conversions liés à l'éditeur *Microsoft Word* (sauvegarde en mode texte). Par ailleurs, dans la mesure où ces polices devraient servir à transcrire aussi bien du français que des langues locales, il est recommandé, dans la mesure du possible, de conserver intacts les codes supportant les caractères accentués nécessaires au français, à savoir quatorze caractères (y compris « œ », « ç ») si on exclut les majuscules accentuées, et quatorze autres dans le cas contraire. Si on observe toutes ces restrictions, on en arrive à 59 codes redéfinissables. Une police suivant ces recommandations permettra de saisir du texte sous n'importe quelle plate-forme et de le visualiser, avec le même aspect, sous une autre. Suivant les besoins de la langue à transcrire, on pourra être amené à réutiliser aussi certains des quatorze caractères majuscules du français.

L'analyse minutieuse entreprise par les formateurs a montré que les polices utilisées à l'Indrap transgressaient à des degrés variés ces recommandations. Les cinq tableaux élaborés, dont l'un figure en Annexe, montrent pour chacune des langues comment les polices encodent chaque caractère spécial. Les cas contraires aux recommandations sont marqués d'un point d'exclamation, voire de deux si le problème est sérieux (redéfinition d'un caractère essentiel). On constate ainsi de *nombreux* problèmes, qui fort heureusement peuvent en général être corrigés lors du passage à Unicode. Les conclusions sont les suivantes pour chaque langue (rappelons les notations : A pour « Add », I pour « Indrap98 », L pour « Langues Niger SIL Doulos », N pour « Nigérienne », H pour « Hausa », T pour « TamajaqTT20.3 ») :
– gourmantchéma : avec deux caractères spéciaux, le « eng » et le « n tilde », majuscules et minuscules, cette langue est la plus simple à codifier. Si le « n tilde » fait partie de ISO-

latin-1, il disparaît parfois des polices exotiques, ce qui fait que seule la police N contient les quatre glyphes nécessaires. I ne contient pas le « n tilde majuscule », et A, L, H, T ne contiennent que l'un des caractères ;
– haoussa : la police la mieux adaptée est L, suivie de la police I (un code problématique). Les polices N et H contiennent aussi les caractères nécessaires, mais presque tous les codes sont contraires aux recommandations (et certains glyphes y figurent deux fois). Les polices A et T ne contiennent aucun des caractères nécessaires ;
– kanouri : la police la mieux adaptée est L, suivie de I. La polices A peut convenir, mais elle redéfinit trois caractères essentiels, et associe à chacun des caractères du kanouri deux codes (les codes 141, 142, 144 n'étant pas affichables, ni utilisés en réalité). Les polices N, H, T ne contiennent pas tous les caractères nécessaires ;
– tamachek : les seules polices qui contiennent tous les caractères nécessaires sont I et T ; les codes utilisés pour le tamachek sont en général convenables (pas de redéfinition de caractères importants), surtout pour la police T. Dans I, deux caractères sont définis deux fois ;
– zarma : la police la mieux adaptée est encore une fois L, bien qu'elle ne possède pas le caractère « N majuscule avec une branche gauche vers la gauche », ni le « u tilde majuscule » ; on peut toutefois utiliser la minuscule avec une taille plus grande pour les rares cas où l'on utilise ces caractères ; aucune des cinq autres polices ne rassemble tous les caractères nécessaires. Notons que nous n'avons pas considéré que les caractères « e, i, o tilde » étaient nécessaires, ce point étant encore débattu par les locuteurs du zarma.

On constate donc que la police ayant la meilleure facture est L, « Langues Niger SIL Doulos », malgré quelques imperfections et la non couverture du tamachek ; cette langue est en revanche bien couverte par le police T. La police I, indrap98, contient beaucoup de caractères nécessaires, mais à des emplacements parfois peu recommandables ; de surcroît, elle semble donner des résultats parfois inattendus avec le navigateur *Internet Explorer*.

De façon plus générale, aucune des polices examinées ne respecte toutes les recommandations, ce qui inciterait un réformateur zélé à proposer leur remplacement par une seule police contenant l'ensemble des caractères nigériens, environ 30, à des emplacements corrects, réalisée par un expert. Parallèlement, un système raisonnablement ergonomique de raccourcis clavier devra être mis en place. Toutefois, il n'est pas facile de remplacer une police déjà longtemps utilisée, donc en l'absence d'un tel changement, on ne peut que recommander chaleureusement la conversion systématique à Unicode/UTF-8 des documents achevés, et leur stockage sur un serveur multilingue – l'un des objectifs des prochaines formations du Rifal.

6 Conclusion et perspectives

Pour conclure, nous devons souligner l'importance de l'interaction entre la théorie et la pratique dans le domaine de l'informatisation des langues. Dans notre cas, la formulation théorique d'un problème concret (la représentation informatique des alphabets utilisés en Afrique de l'Ouest) a débouché sur un algorithme et le logiciel qui l'implémente, qui n'ont pu être mis au point que dans le contexte de l'expérimentation.

Les textes utilisés durant la formation Rifal à l'Indrap doivent subir encore des révisions avant leur conversion définitive à Unicode/UTF-8 et leur partage sur un site du Rifal. Plus important encore, les lexiques bilingues pourront s'encadrer dans une base propre au Rifal, intégrant de nombreux travaux de ce genre. Le potentiel pour des bases documentaires ou terminologiques multilingues semble important au Niger, grâce aux huit langues nationales et au français comme langue officielle. Nombre de textes au format électronique existent dans les différentes langues, mais la proportion de lexiques est encore réduite, selon nos connaissances.

La constitution d'une base de textes et d'une banque de données terminologique multilingue au Niger, en priorité dans le domaine de la pédagogie des différentes disciplines, serait une perspective prometteuse pour de nouvelles actions du Rifal à Niamey. En effet, une telle banque permettrait de répondre à des besoins réels en matière d'instruction scolaire au Niger, mais serait aussi un instrument de recherche précieux pour les linguistes africanistes. Cette banque permettrait également de consolider le rôle de la langue française dans l'aménagement

linguistique en Afrique de l'Ouest, ce qui correspond à la mission du Rifal dans le cadre de l'Agence Intergouvernementale de la Francophonie.

Christian Chanard,
Langage, langues et cultures d'Afrique noire (Llacan),
UMR 7594 du CNRS, Villejuif, France.
chanard@vjf.cnrs.fr

Andrei Popescu-Belis,
Institut pour les études sémantiques et cognitives (Issco),
École de traduction et d'interprétation,
Université de Genève, Genève, Suisse.
andrei.popescu-belis@issco.unige.ch

Remerciements

L'étude présentée dans cet article n'aurait pas été possible sans le soutien de nombreuses personnes. Les auteurs voudraient plus particulièrement remercier Messieurs Marcel Diki-Kidiri (CNRS et GTF/Rifal, Paris), Marcel Grangier (Chancellerie fédérale, Berne) et Louis-Jean Rousseau (Rifal et Office de la langue française, Québec). L'un des auteurs (Andrei Popescu-Belis) souhaite exprimer sa reconnaissance aux Professeurs Bruno de Bessé et Margaret King, ainsi qu'à Florian Simmen (tous de l'Université de Genève). Les auteurs remercient également pour leur accueil et leur intérêt les participants nigériens à la formation Rifal, ainsi que la direction de l'Indrap et tout particulièrement son directeur général, Monsieur Djibo Seybou Kalilou. Enfin, nous remercions les éditeurs de ce numéro des « Cahiers du Rifal » pour leurs bienveillants conseils.

Bibliographie

Consortium Unicode, 2000 : *The Unicode Standard Version 3.0*, Redding, MA, Addison Wesley. Voir aussi www.unicode.org.

Czyborra (R.), 1998 : *The global character set Unicode in the Unix operating system*, mémoire de diplôme, Berlin, Technische Universität. Voir aussi www.czyborra.com.

Grimes (B.F.) et Grimes (J.E.), éd., 2000 : *Ethnologue 14th Edition*, Dallas TX, SIL International / Academic Bookstore. Voir aussi www.sil.org/ethnologue/.

ISO/IEC 10646-1, 2000 : *Information technology – Universal Multiple-Octet Coded Character Set (UCS) – Part 1 : Architecture and Basic Multilingual Plane,* Genève, Organisation Internationale de Normalisation. Voir aussi www.iso.ch.

Organisation des Nations unies, 1948 : « Déclaration universelle des droits de l'homme », Résolution 217 A (III) de l'Assemblée générale du 10.12.1948. Voir aussi www.unhchr.ch/udhr/index.htm.

Annexe : tables

Nous donnons dans la présente annexe, à titre d'exemple, un tableau qui énumère les caractères spéciaux du zarma (Table 1), suivi d'un extrait d'une grille récapitulant les recommandations du LLACAN pour la création de polices de caractères sous *Windows* (Table 2). L'ensemble des tableaux pour les cinq langues traitées dans cet article est disponible sur simple demande auprès des auteurs. Dans chaque tableau figurent les caractères spéciaux tels qu'ils sont utilisés sur place (Indrap, Niamey) – nous avons donc privilégié l'usage par rapport à d'éventuelles méthodes pour l'apprentissage des langues respectives. Pour chacun des caractères, nous donnons sa forme ou glyphe (dans l'une des polices qui le contiennent), le code Unicode (hexadécimal et décimal), le code SILID, et le code entre 0 et 255 du caractère dans les principales polices utilisées à l'Indrap (A pour « Add », I pour « Indrap98 », L pour « Langues Niger SIL Doulos », N pour « Nigérienne », H pour « Hausa », T pour « TamajaqTT20.3 »). Lorsqu'un code est absent, cela signifie que la police en question n'encode pas le caractère figurant en début de ligne. Nous donnons également une appréciation de la cohérence de chaque affectation de code pour chaque caractère dans chaque police. L'absence de marque dénote la cohérence, un point d'exclamation signifie que le choix du code respectif est contraire aux recommandations (en général parce qu'il redéfinit un caractère courant, souvent une lettre accentuée du français), deux points d'exclamation dénotent un choix dangereux (souvent la redéfinition d'un code ASCII entre 0 et 128).

Table 1. Disponibilité des caractères spéciaux du zarma (il n'est pas encore établi si les voyelles e, i, o avec tilde sont également nécessaires).

Glyphe	Code Unicode hexa.	déc.	SILID	Code dans la police A	I	L	N	H	T
ə	00E3	0227	4501	227	! 134	227	227	227	-
Ə	00C3	0196	3501	195	195	-	195	195	-
ŋ	014B	0331	1418	-	! 238	191	! 249	-	245
Ŋ / Ɲ	014A	0330	2418	-	237	186	‼ 037 167	-	213
ɲ	0272	0626	1618	-	-	192	-	-	-
Ɲ	019D	0413	2818	-	-	-	-	-	-
ũ	0169	0361	0117 +6001	-	-	255	-	-	-
Ũ	0168	0360	0085 +6101	-	-	-	-	-	-

Table 6. Grille des caractères ISO-latin-1 avec les possibilités de redéfinition (extrait).