Informatisation de dictionnaires langues africaines-français

Mathieu Mangeot, Chantal Enguehard

Mathieu Mangeot & Chantal Enguehard (2011) Informatisation de dictionnaires langues africaines-français. Actes des journées LTT 2011, Villetaneuse, 15-16 septembre 2011.

Mathieu Mangeot(1) & Chantal Enguehard(2)
(1) GETALP-LIG, 385 rue de la bibliothèque, BP 53
F-38042 Grenoble Cedex 9,
France
& Université de Savoie
Mathieu.Mangeot@imag.fr
(2) LINA, 2 rue de la Houssinière, BP 92208
F-44322 Nantes Cedex 03,
France
Chantal.Enguehard@univ-nantes.fr

Informatisation de dictionnaires langues africaines-français

Résumé
Cet article présente des travaux réalisés dans le cadre du projet DiLAF qui vise à informatiser des dictionnaires langues africaines-français (bambara, haoussa, kanouri, tamajaq, songhai-zarma) afin de pouvoir les diffuser plus largement et étendre leur couverture. Nous présentons une méthodologie de récupération de dictionnaires au format .doc et leur conversion dans un format XML structuré suivant les standards du domaine comme Unicode et Lexical Markup Framework. Cette méthodologie se veut simple et compréhensible par un linguiste sachant manipuler les expressions régulières. Les outils nécessaires sont gratuits, en source ouverte et multi-plate-formes. La méthode décrite est suffisamment générique pour être appliquée sur des dictionnaires d'autres langues, voire toute ressource textuelle.

Mots-clés : dictionnaire, récupération, conversion, XML, LMF, LibreOffice, expressions régulières, projet DiLAF.

1 Introduction

Les travaux à l'origine de cet article ont été réalisés dans le cadre du projet DiLAF qui vise à informatiser des dictionnaires langues africaines-français (bambara, haoussa, kanouri, tamajaq, songhai-zarma) afin de pouvoir les diffuser largement et étendre leur couverture pour ensuite continuer les étapes suivantes dans l'informatisation de ces langues. Nous présentons une méthodologie de récupération de dictionnaires au format .doc de Word ou .odt de OpenOffice et leur conversion dans un format XML structuré suivant les standards du domaine comme Unicode pour les codages de caractères et Lexical Markup Framework (LMF) pour les structures de ressources lexicales. Le traitement automatique des langues africaines en est à ses débuts. Il est de notre devoir d'aider nos collègues des pays du Sud dans cette voie. Cela passe, entre autres, par la publication d'articles, qui, s'ils n'ont peut être pas un intérêt scientifique novateur évident pour le TAL européen constituent une ressource précieuse pour les langues peu dotées. Suivant les avancées de l'informatique et l'arrivée du format XML, de nombreux travaux ont été effectués par le passé dans ce domaine. Voir par exemple (Lafourcade 1996) pour la récupération du dictionnaire FeM français-anglais-malais et la thèse de Hai Doan Nguyen (Doan-Nguyen 1998). Il nous a semblé cependant intéressant de redéfinir une nouvelle méthodologie prenant en compte les récents développements comme le format Open Document Format (ODF) ou le standard LMF. D'autre part, nous voulions mettre au point une méthode la plus simple possible et fondée exclusivement sur des outils gratuits et en source ouverte (OpenSource) afin qu'elle puisse être réutilisée par le plus grand nombre. Cette méthode peut d'ailleurs être utilisée pour des dictionnaires portant sur d'autres langues et par extension, sur tout document textuel (ressource linguistique au sens large) à convertir au format XML. Nous présenterons d'abord le projet DiLAF puis la méthodologie dans son ensemble. Les parties suivantes en détailleront chacune une partie : la conversion des caractères vers Unicode, la conversion du format vers XML, le marquage explicite des informations et la correction des données puis enfin la structuration des articles.

2 Présentation du projet DiLAF

Si l'accès aux ordinateurs est considéré comme le principal indicateur de la fracture numérique en Afrique, il faut reconnaître que la disponibilité des ressources dans les langues africaines constitue un handicap dont les conséquences sont incalculables pour le développement des Technologies de l'Information et de la Communication (TIC) dans cette partie du monde. Aussi, la production, la diffusion et la vulgarisation de ressources locales portant sur ces langues nous paraissent-elles être indiquées pour une implantation durable des TIC sur le continent. Or, la plupart des langues de

l'espace francophone d'Afrique de l'Ouest sont peu dotées (langues-π) (Berment 2004) : les ressources électroniques disponibles sont rares, mal distribuées, voire inexistantes. Seules sont accessibles les fonctions d'édition et d'impression rendant l'exploitation de ces langues difficile au moment où il est question de les introduire dans le système éducatif, de créer des normes d'écriture standardisées et stabilisées et surtout de développer leur usage à l'écrit dans l'administration et la vie quotidienne.
Aussi, afin de contribuer à combler ce retard, nous nous sommes engagés avec les collègues du Sud et du Nord à améliorer l'équipement de quelques langues africaines à travers, entre autres, l'informatisation de dictionnaires éditoriaux portant sur des langues africaines. A cet effet, nous avons lancé le projet DiLAF (Dictionnaires Langues Africaines Français) qui vise à convertir des dictionnaires éditoriaux bilingues en un format XML permettant leur pérennisation et leur partage (Streiter et al., 2006). Ce projet international rassemble des partenaires du Burkina Faso (Centre National de la Recherche Scientifique et Technologique), de France (Laboratoire d'Informatique de Grenoble et Laboratoire d'informatique de Nantes-Atlantique), du Mali (Centre National de Ressources de l'éducation Non Formelle) et du Niger (Institut National de Documentation de Recherche et d'Animation Pédagogiques, Ministère de l'Education Nationale, et Université Abdou Moumouni de Niamey).
En nous fondant sur un travail déjà effectué par des lexicographes nous avons constitué des équipes pluridisciplinaires constituées de linguistes, d'informaticiens et de pédagogues. Cinq dictionnaires comportant, chacun, plusieurs milliers d'entrées, devraient être convertis et intégrés à une plate-forme Jibiki de gestion de ressources lexicales (Mangeot, 2001). Les dictionnaires seront donc disponibles sur Internet d'ici la fin de l'année 2012[i] sous licence Creative Commons :

- dictionnaire bambara-français, Charles Bailleul, édition 1996 ;
- dictionnaire haoussa-français destiné à l'enseignement du cycle de base 1, 2008, Soutéba ;
- dictionnaire kanouri-français destiné pour le cycle de base 1, 2004, Soutéba ;
- dictionnaire tamajaq-français destiné à l'enseignement du cycle de base 1, 2007, Soutéba ;
- dictionnaire zarma-français destiné à l'enseignement du cycle de base 1, 2007, Soutéba.

Il s'agit de dictionnaires d'usage qui visent à vulgariser les formes écrites de l'usage quotidien des langues africaines dans la pure tradition lexicographique (Matoré 1973), (Eluerd 2000). Se démarquant des démarches dirigistes des dictionnaires normatifs (Mortureux 1997), les présents dictionnaires descriptifs restent ouverts aux contributions et leur mise en ligne devra, nous l'espérons, développer un sentiment de fierté chez les usagers des différentes langues. De même, ils participeront au développement d'un environnement lettré propice à l'alphabétisation dont le faible taux compromet les acquis des progrès réalisés dans les autres secteurs.

3 Méthodologie générale de conversion et outils utilisés

L'objectif principal de cette méthodologie est de convertir des dictionnaires au format d'un éditeur de texte adaptés à un usage humain vers le format XML et de marquer explicitement toutes les informations afin de pouvoir les utiliser de manière automatique dans des tâches de traitement automatique des langues. Les contraintes sont, d'une part, de travailler avec des outils gratuits, en source ouverte et multi-plate-formes et, d'autre part, de définir un processus simple qui puisse être compris et réalisé ensuite de manière autonome par des linguistes n'ayant pas de connaissances en informatique en dehors de la maîtrise des expressions régulières.

3.1 Méthodologie de conversion

La méthodologie de conversion des dictionnaires au format XML consiste à suivre les étapes suivantes :

- 1. conversion des caractères problématiques conséquences des polices arrangées vers Unicode;
- 2. conversion du format OpenOffice vers XML;
- 3. identification et marquage explicite de chaque partie d'information (mot-vedette, classe grammaticale, etc.);
- 4. validation XML et correction manuelle d'éventuelles erreurs dans les données (listes fermées de valeurs, renvois);
- 5. structuration des entrées suivant la philosophie du standard LMF.

3.2 Outils utilisés

Le premier outil permet d'éditer les fichiers au format d'origine puis de les convertir en XML. Pour cette étape OpenOffice (ou son évolution LibreOffice) est l'idéal. Il est gratuit et en source ouverte. D'autre part, outre le standard ISO Open Document Format (.odt), il peut ouvrir de nombreux formats de Microsoft (rtf, .doc et également .docx). Enfin, le XML produit est simple, surtout comparé au format Office Open XML de Microsoft.
OpenOffice dispose d'un moteur d'expressions régulières pour les fonctions de rechercher / remplacer. Cet outil peut

donc être utilisé pour de nombreuses étapes avant la conversion en XML. Cependant, les fonctions de rechercher/remplacer posent problème car lors des remplacements, les frontières des styles du texte peuvent être modifiés (une partie de mot se retrouve soudain en italique). étant donné que par la suite, nous nous basons sur les styles pour convertir vers XML, nous nous limiterons donc à la conversion des caractères vers Unicode afin de garder les styles intacts.
Une fois la conversion vers XML effectuée, nous avons besoin d'un éditeur pour pouvoir modifier les fichiers. Pour ces opérations, les éditeurs XML ne sont pas d'une grande utilité car ils ne permettent pas de modifier directement le texte brut avec des expressions régulières et la plupart ne sont pas capables d'éditer des gros fichiers comme peuvent l'être des dictionnaires. Nous conseillons d'utiliser un simple éditeur de texte « brut » avec les fonctions de rechercher/remplacer supportant un langage d'expressions régulières et coloration de la syntaxe (ce dernier point n'est pas nécessaire mais nettement plus agréable). Les logiciels suivants conviennent parfaitement : NotePad++ pour Windows, gedit pour linux et TextWrangler pour MacOs.
Pour l'étape de validation XML et vérification des données, là non plus, les éditeurs XML ne sont pas indispensables. Un navigateur Web tel que FireFox fait très bien l'affaire. Il est capable de détecter et d'afficher les erreurs de validation XML et peut interpréter des feuilles de style CSS et XSLT pour améliorer l'affichage.

3.3 Sauvegardes incrémentales obligatoires

La méthodologie prévoit de faire des sauvegardes à chaque étape si minime soit-elle et de garder une trace de toutes les opérations de rechercher/remplacer effectués pour pouvoir revenir en arrière lorsque des erreurs issues des mauvaises manipulations sont identifiées. Parfois, il arrive que l'on s'aperçoive d'une erreur longtemps après l'avoir effectuée. Si une erreur ne peut être corrigée simplement par un nouveau rechercher/remplacer, il est alors possible de revenir en arrière en repartant d'une version précédente.
Malgré toutes les précautions, il arrive parfois que des erreurs soient détectées très tardivement et qu'il soit très difficile de revenir en arrière. Si l'erreur n'est pas détectable automatiquement, il faudra une correction manuelle. Si cela vous arrive, sachez que personne n'est parfait et que d'autres pourtant mieux entraînés ont dû se résoudre à ne pas pouvoir corriger automatiquement toutes les erreurs de conversion.

4 Conversion des caractères vers Unicode

Bien que les alphabets des langues sur lesquels nous avons travaillé (Enguehard 2009) soient majoritairement d'origine latine, de nouveaux caractères nécessaires pour noter des sons spécifiques à certaines langues à l'aide d'un seul caractère ont été adoptés par les linguistes lors d'une série de réunions. La première, en septembre 1978, organisée par l'UNESCO au CELTHO (Centre d'études linguistiques et historiques par tradition orale) à Niamey crée l'« Alphabet africain de référence » fondé sur les conventions de l'IPA (International Phonetic Association) et de l'IAI (International African Institute). Ainsi, chacun des alphabets que nous avons précédemment présentés comprend au moins un de ces caractères spéciaux : ɓ ɗ ə ɛ ɣ ƙ ɲ ŋ ɔ ʸ. Des caractères composés d'un caractère latin et d'un signe diacritique ont également été créés : â ê î ô û ă ã ẽ ĩ õ ũ ḍ ḷ ṣ ṭ ẓ gi̯ š .
–
Comme nombre de ces caractères étaient absents des dispositifs de saisie et des standards alors en usage, des touches de frappe de machines à écrire, des glyphes de polices d'ordinateurs ont été modifiées. Bien que la plupart de ces caractères soient depuis plusieurs années présents dans le standard Unicode (issu des travaux du comité ISO 10646 (Haralambous 2004)), les dictionnaires dont nous disposons ont été rédigés en utilisant les anciennes polices arrangées. Une méthodologie a été définie afin de repérer et remplacer les caractères inadéquats par les caractères définis dans le standard Unicode. Suivre cette méthodologie implique que l'ensemble des caractères repérés et leurs caractères de remplacement soient notés dans un fichier afin de pouvoir réitérer facilement cette opération en cas de nécessité. Le tableau 1 montre une partie de cette liste pour le zarma. Il n'y a aucune méthode automatique qui permette de détecter ces caractères problématiques. De plus, il arrive que certaines polices modifiées soient introuvables ou que deux polices différentes portent le même nom. Il est donc impératif de regarder les données. De plus, certains caractères peuvent n'apparaître qu'après avoir vérifié une grande partie du fichier. Il est donc nécessaire de stocker la liste jusqu'à ce que la conversion complète soit effectuée.

origine	Unicode
§	ã
é	ẽ
$	ɲ
ù	ŋ
£	Ɲ

Tableau 1: vue partielle du tableau de correspondance des caractères Unicode pour le zarma

En principe, il est conseillé d'effectuer ces conversions directement dans LibreOffice car il est possible de changer de police (police standard et police arrangée) à tout moment pour voir quel est le caractère qui doit être remplacé par un caractère Unicode. Toutefois, certains cas obligent à faire la conversion dans le fichier XML une fois que les informations ont été explicitement marquées. Dans notre exemple, pour le dictionnaire tamajaq-français, le caractère « p » avait été choisi pour remplacer la lettre schwa « ə » car le « p » n'existe pas en tamajaq. Le problème est qu'il existe bien sûr en français. Donc, si tous les « p » sont transformés en « ə », les « p » des mots français seront également convertis. Il faut donc attendre d'avoir délimité les parties d'information en français pour ne convertir que les autres.

5 Conversion du format vers XML

La figure 1 montre un extrait du dictionnaire zarma-français au format original .odt (ouvert avec LibreOffice). Tous les exemples suivants seront basés sur ce dictionnaire.

abirillu [àbìríllù] *m.* • *avril* • annasaara handu taacanta kaŋ go marsu nda me game ra • *Abirillu, 15, 1974 no Sayni Kunce na hino sambu* • *f/b.* abirillo, abirilley

abiyanso [àbìyànsôo] *m.* • *aéroport* • batama kaŋ ra abiyey ga zumbu • *Tilbeeri nda Dooso sinda abiyanso kaŋ ra abiyo beeri ga zumbu* • *f/b.* abiyansa, abiyansey

abiyo [àbíyò] *m.* • *avion* • naarumay hari no kaŋ ra i ga boro nda jinay daŋ a ma deesi nd'ey • *Jidda no abiyey ga alfujaajey zumandi* • *him.* beene-hi • *f/b.* abiya, abiyey

abunaadam [àbúnăadàm] *m.* • *être humain, personne* • *f/b.* abunaadamo, abunaadamey • *di.* adamayze

Figure 1: Extrait du dictionnaire zarma-français au format original

Le format Open Document Format[ii] est un standard OASIS. La version 1.0 est également un standard ISO 26300:2006. La version 1.1 a été approuvée par OASIS le 2 février 2007. La version 1.2 est en cours de rédaction. Ce format est utilisé de manière native par les suites bureautiques de la famille de OpenOffice (StarOffice, NeoOffice, LibreOffice). Il a le précieux avantage d'être fondé sur un format XML. Le titre de cette partie et donc un peu trompeur. Au lieu d'une conversion, nous allons en fait nous contenter de récupérer le contenu XML du document puis nous le transformerons pour obtenir ce que nous souhaitons.
Un document .odt au format ODF est en fait une archive zippée de plusieurs fichiers dont le contenu du texte balisé en XML. Ce contenu est stocké dans le fichier content.xml à la racine de cette archive. Pour récupérer ce fichier, il suffit de suivre quelques manipulations astucieuses. Sur Mac, cela consiste à créer un dossier vide puis copier ensuite le fichier odt à l'intérieur. Ensuite, il faut ouvrir un terminal puis exécuter la commande unzip sur le fichier .odt. Sur Windows, il faut changer l'extension du fichier .odt en .zip puis ouvrir l'archive .zip.
Le fichier *content.xml* peut être maintenant extrait de l'archive puis renommé et placé dans un autre endroit. Il devient le fichier de base sur lequel nous allons poursuivre notre travail. L'étape suivante consiste donc à éditer ce fichier avec un éditeur de texte « brut ». Le fichier étant constitué d'une seule ligne, la première manipulation à faire et de rajouter des sauts de ligne. Si le document de départ est bien rédigé, chaque article est probablement contenu dans un paragraphe. Si l'on ajoute un saut de ligne à chaque paragraphe, cela permettra alors de visualiser chaque article sur une seule ligne. L'expression régulière suivante devrait nous tirer d'affaire : s/<text:p/\r<text:p/g
L'en-tête contenant les informations spécifiques à OpenOffice peut maintenant être supprimé. La première ligne sera constituée du premier article de notre dictionnaire.

6 Marquage explicite des informations

Cette étape consiste à marquer explicitement toutes les parties d'information. Chaque partie d'information se distingue le plus souvent des autres dans le fichier d'origine par un style différent. Si le fichier d'origine a été bien conçu, il est possible d'identifier facilement le style correspondant à l'information. La figure 2 montre une partie de l'article « abiyanso » (aéroport) du dictionnaire zarma-français au format ODF. On peut voir que le style utilisé pour marquer la prononciation est « Phonetic_form ». Ce dictionnaire a été conçu à l'origine avec le logiciel Shoebox (Buseman et al. 2000), ce qui a permis de garder dans le nom du style, l'étiquette utilisée pour marquer l'information avec Shoebox. Pour d'autres dictionnaires, le nom du style est plus cryptique. Par exemple, pour le dictionnaire bambara-français qui a été rédigé directement avec Word, les traductions françaises sont notées avec le style « T21 ». Il est cependant possible de s'y retrouver en comparant avec le fichier d'origine.

```
<text:span text:style-name="Phonetic_20_form"><text:span text:style-
name="T7">[àbìyànsôo]</text:span></text:span>
```

Figure 2 – Extrait d'article (prononciation) au format ODF XML

Après avoir localisé les parties d'informations, il faut choisir un ensemble de balises pour les marquer. Se pose alors la question du choix de la langue utilisée pour les balises. Le choix de l'anglais, langue internationale de la recherche, peut être privilégié. Mais dans notre cas, l'anglais n'est pas une langue présente dans nos dictionnaires d'une part et d'autre part, elle n'est pas maîtrisée par tous les collègues linguistes travaillant sur le projet. L'utilisation du français résout ce problème puisque tous les partenaires présents maîtrisent cette langue. Toutefois, dans le cas de projets d'informatisation de langues peu dotées, nous pensons qu'il est important d'inciter les partenaires à utiliser les termes de leur langue pour définir le nom des balises en langue source (ou langues nationales). Cette démarche peut donner lieu éventuellement à la création de nouveaux termes qui n'existaient pas dans ces langues. D'un point de vue politique, il s'agit de s'éloigner d'une vision post-coloniale des statuts sociaux des langues africaines et contribue à leur valorisation.
Le jeu des balises étant maintenant défini, il reste à remplacer le balisage au format ODF par ce nouveau balisage « maison ». Il suffit d'effectuer des opérations de rechercher/remplacer pour chaque type d'information. Pour l'exemple de la figure 2, une première expression régulière (syntaxe perl) supprime la balise « T7 » : s/<text:span text:style-name="T7">([^<]+)<\/text:span>//g La deuxième expression remplacera la balise « Phonetic_form » par « ciiyaŋ » : s/<text:span text:style-name="Phonetic_20_form">([^<]+)<\/text:span>/ <ciiyaŋ>$1<\/ciiyaŋ>/g
Le remplacement de toutes les balises aboutit au résultat de la figure 3. L'étape de marquage des informations est maintenant terminée.

```
<sanniize>abiyanso</sanniize><ciiyaŋ>[àbìyànsôo]</ciiyaŋ>
<kanandi>m.</kanandi><bareyaŋ>aéroport</bareyaŋ> <feeriji>batama kaŋ ra abiyey ga
zumbu</feeriji> <silmaŋ>Tilbeeri nda Dooso sinda abiyanso kaŋ ra abiyo beeri ga
zumbu</silmaŋ> <f>abiyansa</f><b>abiyansey</b>
```

Figure 3 – Article converti avec des balises « maison »

7 Correction des données

Lors de cette étape, plusieurs corrections sont effectuées sur les données.

7.1 Validation XML

Pour pouvoir utiliser des outils manipulant des fichiers XML, il faut que notre fichier soit bien formé d'un point de vue de la syntaxe XML. Dans l'étape précédente, les remplacement successifs s'appliquent à un fichier au format ODF qui lui était bien formé. Si le travail est correct, le résultat devrait être théoriquement bien formé. Dans la pratique, on s'aperçoit que ce n'est jamais le cas. C'est d'ailleurs au moment de vérifier que le fichier est bien formé que certaines erreurs dues à l'étape précédente de remplacement des balises sont détectées.
Pour vérifier la syntaxe de notre document, nous avons besoin d'un analyseur (parseur) XML. Comme expliqué dans l'étape du choix des outils, il est possible d'utiliser un simple navigateur Web tel que FireFox. Celui-ci inclut un analyseur XML qui est également capable d'indiquer précisément où sont les erreurs dans le fichier. La figure 4 montre

le résultat de l'analyse d'un fichier qui n'est pas bien formé. La balise « sanniize » n'est pas correctement terminée.

Figure 4 – Affichage d'une erreur de syntaxe XML

Une fois l'erreur localisée, il faut vérifier si elle ne se répète pas ailleurs dans le fichier. Si c'est le cas, il faut il faut élaborer une expression régulière qui corrige l'erreur de manière systématique au lieu de le faire à la main. Dans le cas présent, l'expression régulière suivante réglera le problème : s/<sanniize([^<]+)<\/sanniize>/<sanniize>$1<\/sanniize>/g Le fichier XML est maintenant bien formé. Il est alors possible de le manipuler avec des outils XML. Les manipulations sur le fichier n'étant pas terminées, il faudra périodiquement vérifier la syntaxe.

7.2 Vérification des listes de valeurs

L'étape de la vérification des informations prenant leur valeur dans une liste fermée est importante. Certaines erreurs proviennent de mauvaises manipulations lors des étapes précédentes, tandis que d'autres étaient présentes dans le fichier d'origine avant conversion. Pour un dictionnaire, figure par exemple la classe grammaticale. Pour une base terminologique, le domaine est généralement à vérifier. Faire une copie du fichier puis ne garder que les valeurs à vérifier constitue une démarche de vérification systématique. Dans l'exemple de la figure 3, nous pouvons extraire la classe grammaticale balisée par « kanandi » avec l'expression suivante : s/^.*<kanandi>([^<]+)<\/kanandi>.*$/$1/ Il faut ensuite trier la liste obtenue par ordre alphabétique. TextWrangler et Notepad++ avec son plugin TextFX ont les commandes nécessaires. Si l'éditeur choisi ne propose pas cette option, Calc, le tableur de OpenOffice peut être utilisé. Cette démarche permet ensuite de détecter rapidement les irrégularités. Si une valeur n'apparaît qu'une seule fois, il est fort probable que ce soit une erreur. Dans le dictionnaire pris en exemple, nous avons corrigé « alteeb » en « alteeb. », « dah. » en « dab. », « m/tsif. » en « m / tsif. », etc.

7.3 Corrections simples

Le fichier de travail étant bien formé, une feuille de style CSS[iii] peut être définie pour visualiser les données directement dans le navigateur. Un affichage compact et un style différent pour chaque type d'information permet de déceler les erreurs de structuration d'un article. Dans l'exemple de la figure 5, nous voyons tout de suite qu'il manque la définition (en gras) et l'exemple (en italique) pour l'entrée « abunaadam ».
Le langage XSLT[iv], langage de transformation d'arbres XML, permet de modifier les données avant l'affichage. Par exemple, une feuille de style XSL peut ajouter comme identifiant unique à chaque article, son mot-vedette puis, pour chaque renvoi ou synonyme définir un lien hypertexte pointant vers l'article correspondant. Lorsque le linguiste parcourt le fichier, il peut alors cliquer sur les liens hypertextes (signalés par un soulignement) pour vérifier que les renvois et synonymes sont également des articles du dictionnaire. Dans l'exemple de la figure 5, l'entrée « abunaadam » contient un renvoi vers l'entrée « adamayse ». Le mot est souligné, ce qui indique la présence d'un lien hypertexte.

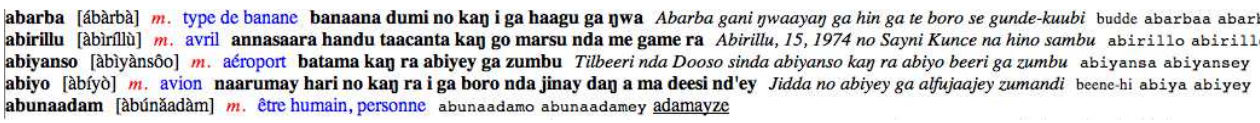

Figure 5 – Vue compacte dans un navigateur

Il est essentiel de scruter les données pour détecter certaines erreurs, même si celles-ci peuvent se régler automatiquement par la suite avec des expressions régulières. L'étape de visualisation des données est aussi très importante d'un point de vue pédagogique. Elle permet de montrer aux linguistes les avantages de l'encodage des données en XML, en particulier que différentes formes possibles (style) peuvent être associées au même fond (les

données). En apprenant des rudiments du langage CSS, ils peuvent alors modifier eux-mêmes les feuilles de style.

8 Structuration des articles

Chaque information ayant été marquée et les erreurs évidentes corrigées, les articles peuvent être restructurés. Dans les fichiers issus de logiciels de traitement de texte, il est très rare qu'une structuration des données à l'intérieur d'un article soit visible. La plupart du temps, celle-ci est implicite. Il va falloir d'abord déduire la structuration qui a été choisie au départ à partir des données puis ajouter ensuite de nouveaux éléments de structuration afin de tendre vers une structure la plus standardisée possible, ce qui permettra des réutilisations ultérieures. Concernant les standards, Lexical Markup Framework (LMF) (Romary et al. 2004), devenu norme ISO numéro 24613 :2008 en novembre 2008 (Francopoulo et al. 2009) convient parfaitement à nos objectifs. Comme il s'agit d'un méta-modèle et non un format, nous pouvons appliquer l'esprit de LMF à notre structuration et non suivre une représentation strictement définie. Le méta-modèle noyau de LMF est représenté par la figure 6. L'objet « Lexical Entry » contient un objet « Form » et un ou plusieurs objets « Sense »

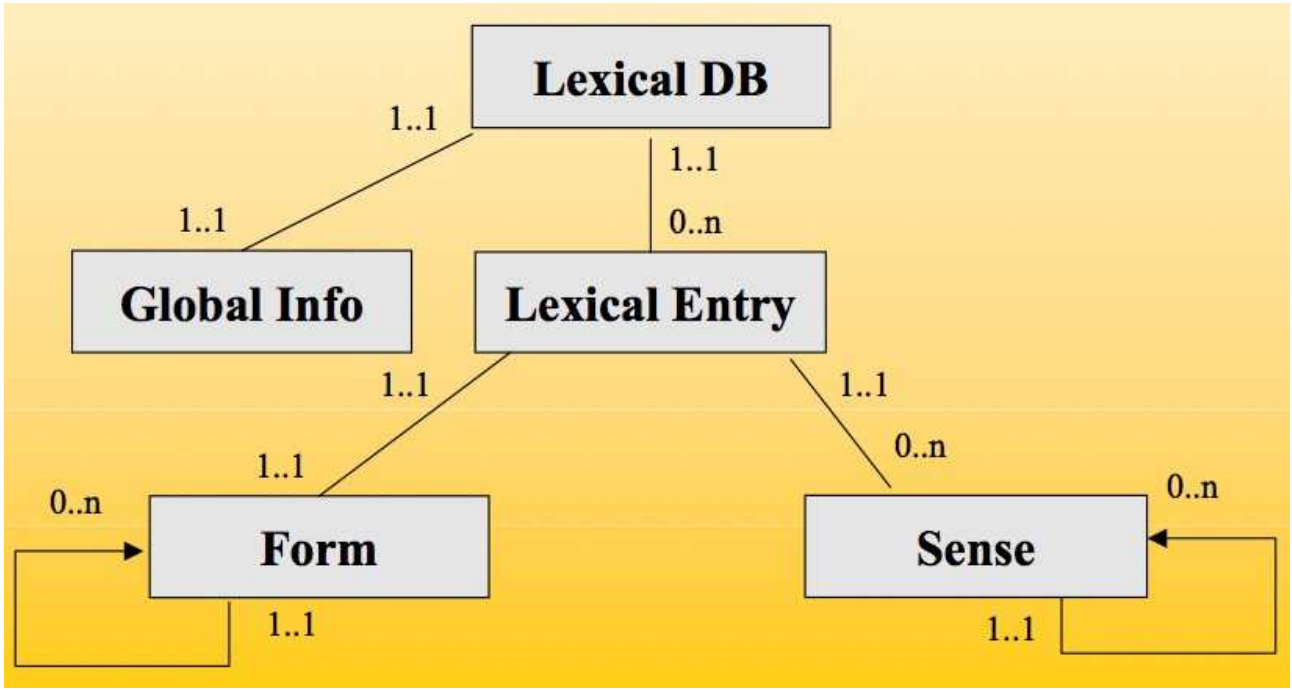

Figure 6 – Méta-modèle noyau de LMF

Il faut maintenant structurer nos articles en suivant ce méta-modèle. Les figures 7 et 8 présentent un exemple d'article respectivement avant et après l'ajout des balises de structuration. La balise « article » correspond à l'objet « Lexical Entry », la balise « bloc-vedette » correspond à l'objet « Form » et la balise « bloc-semantique » correspond à l'objet « Sense ». Le lecteur attentif aura noté ici que le nom de ces balises est en franaçais et n'a dont pas encore été traduit en langue source (zarma). La réflexion sur la structuration des articles n'est pas terminée. Elle se poursuivra dans les ateliers suivants du projet DiLAF.

Figure 7 – Article avant structuration

Si nous avons respecté l'esprit de LMF, il doit être maintenant possible de convertir automatiquement nos articles vers un format respectant la lettre de LMF avec une feuille de style XSLT. La figure 9 montre la conversion au format LMF de l'article précédent. Il apparaît que le résultat est nettement plus verbeux et que les noms des balises comme des attributs sont en anglais. On peut comprendre facilement que ce type de format n'est pas idéal pour mettre un linguiste à l'aise avec le format XML.

Figure 8 – Article après structuration

```
<LexicalEntry> <feat att="partOfSpeech" val="commonNoun"/> <Lemma> <feat att="writtenForm" val="abiyanso"/> <feat att="phoneticForm" val="àbìyànsôo"/> </Lemma> <WordForm><feat att="writtenForm" val="abiyansa"/></WordForm>
```

```
<WordForm><feat att="writtenForm" val="abiyansey"/></WordForm> <Sense>
<Equivalent> <feat att="language" val="fra"/> <feat att="writtenForm" val="aéroport"/>
</Equivalent> <Context><feat att="writtenForm" val="Tilbeeri nda Dooso sinda
abiyanso kaŋ ra abiyo beeri ga zumbu"/> </Context> <Definition> <feat
att="writtenForm" val="batama kaŋ ra abiyey ga zumbu"/> </Definition> </Sense>
</LexicalEntry>
```

Figure 9 – Article au format LMF

9 Conclusion

Nous avons présenté une méthodologie de récupération de dictionnaires issus de traitements de texte et leur conversion au format XML. Le projet DiLAF ne s'arrête pas en si bon chemin. Avant de distribuer les dictionnaires, il reste des étapes de correction manuelle et éventuellement d'ajout de données. Par exemple, les exemples du dictionnaire zarma-français seront traduits en français. Ensuite, une fois les dictionnaires convertis, nous pourrons alors étendre leur couverture via un système de contribution/révision/validation qui peut se faire en ligne en direct sur la plate-forme Jibiki (Mangeot & Chalvin 2006). Le faible accès en ligne en Afrique nous obligera à mettre en place des méthodes alternatives. Nous pourrons ensuite les utiliser comme matière première pour avancer dans l'informatisation de ces langues : analyseurs morphologiques, correcteurs orthographiques, systèmes statistiques de traduction automatique, etc.

Remerciements

Le projet DiLAF est financé par le Fonds Francophone des Inforoutes de l'Organisation Internationale de la Francophonie[v]. Nous remercions également tous les linguistes de l'équipe sans qui ce projet n'aurait pas pu voir le jour : Soumana Kané, Issouf Modi, Michel, Radji, Rakia, Mamadou Lamine Sanogo.

Bibliographie

Berment V. (2004). *Méthodes pour informatiser des langues et des groupes de langues « peu dotées »*. Thèse de nouveau doctorat, spécialité informatique, Université Joseph Fourier Grenoble I, Grenoble, France.

Buseman A., Buseman K., Jordan D. & Coward D. (2000). *The linguist's shoebox : tutorial and user's guide : integrated data management and analysis for the field linguist*, volume viii. Waxhaw, North Carolina : SIL International.

Doan-Nguyen H. (1998). *Techniques génériques d'accumulation d'ensembles lexicaux structurés à partir de ressources dictionnairiques informatisées multilingues hétérogènes.* Thèse de nouveau doctorat, spécialité informatique, Institut National Polytechnique de Grenoble, Grenoble, France.

Eluerd R. (2000). *La Lexicologie.* Paris : PUF, Que sais-je ?

Enguehard C. (2009). *Les langues d'afrique de l'ouest : de l'imprimante au traitement automatique des langues.* Sciences et Techniques du Langage, 6, 29–50.

Francopoulo G., Bel N., George M., Calzolari N., Monachini M., Pet M. & Soria C. (2009). Multilingual resources for nlp in the lexical markup framework (lmf). *Language Resources and Evaluation*, 43, 57–70. 10.1007/s10579-008-9077-5.

Haralambous Y. (2004). *Fontes et codages*. O'Reilly France.

Lafourcade M. (1996). Serveurs de dictionnaires - étude de cas avec l'outil alex et le projet de dictionnaire français-anglais-malais. In *TALN'96*, volume 1, p. 162–168, Grenoble, France.

Mangeot M. (2001). *Environnements centralisés et distribués pour lexicographes et lexicologues en contexte multilingue.* Thèse de nouveau doctorat, spécialité informatique, Université Joseph Fourier Grenoble I, Grenoble, France.

Mangeot M. & Chalvin A. (2006). Dictionary building with the jibiki platform : the GDEF case. In *LREC* 2006, p. 1666–1669, Genova, Italy. Matoré G. (1973). *La Méthode en lexicologie*. Paris, France : Didier.

Mortureux M.-F. (1997). *La lexicologie entre langue et discours*. Paris, France : SEDES.

Romary L., Salmon-Alt S. & Francopoulo G. (2004). Standards going concrete : from LMF to Morphalou. In Proceedings of the COLING Workshop on Enhancing and Using Electronic Dictionaries, *ElectricDict*'04, p. 22–28, Stroudsburg, PA, USA : Association for Computational Linguistics.

Streiter O., Scannell K. & Stuflesser M. (2006). Implementing NLP projects for non-central languages : Instructions for funding bodies, strategies for developers. In *Machine Translation*, volume 20.

i http://dilaf.org
ii http://docs.oasis-open.org/office/v1.1/OS/OpenDocument-v1.1.pdf
iii http://www.w3.org/TR/CSS21/
iv http://www.w3.org/TR/xslt
v http://www.inforoutes.francophonie.org/projets/projet.cfm?der_id=262