COMPARAISON LEXICALE AUTOMATIQUE (COLEXA)

J.P. NZALI, M. DIEU[1] , M.H. NGOA
DEPARTEMENT D'INFORMATIQUE
FACULTE DES SCIENCES B.P. 812
YAOUNDE CAMEROUN

RESUME

Dans le cadre du programme "Atlas Linguistique du Cameroun"[5] mené par l'institut des Sciences Humaines du Cameroun en son Centre de Recherches et d'Etudes Anthropologiques, a été développé un système informatique de comparaison lexicale automatique en abrégé COLEXA dont l'objectif est de fournir une classification en arbre d'un ensemble de langues sur la base de décomptes lexicostatistiques. Le présent document expose les principes de ce système informatique.

Mots clés: Comparaison lexicale, Alignement, Distance, Intelligence Artificielle

INTRODUCTION

Si on s'arrête à la définition de l'Intelligence Artificielle dûe à PITRAT "faire faire par une machine tout ce que l'homme est capable de faire", alors COLEXA est un produit IA. Si on préfère parler d'informatique symbolique, plus exactement de symbolic processing, COLEXA est aussi dans le domaine. COLEXA utilise la capacité de l'ordinateur à traiter beaucoup d'informations (comparaison de 120 objets entre eux). Il enregistre le savoir faire d'un ou de plusieurs experts linguistes, et le traitement automatique qui s'ensuit introduit une part de subjectivité. Il fait appel à des outils de sélection aléatoire et d'analyse de données, nécessitant l'outil informatique.

A - EXPOSE DES PRINCIPES ET DES OBJECTIFS

I - PRINCIPES

Les données à traiter, recueillies sur le terrain par les linguistes, sont des listes standard de 120 concepts, traduites dans chacune des langues du pays ou de la région étudiée. Une langue est donc représentée par 120 mots. C'est peu, mais suffisant pour fonder une classification de ces listes, c'est-à-dire de ces langues,

[1]Initiateur et coordonnateur du projet, décédé en 1992

selon leur degré de ressemblance mutuelle. C'est le principe de la lexicostatistique que nous résumons de la façon suivante:

- Soit une liste de concepts fondamentaux (partie du corps, animaux familiers, premiers numéraux, actions élémentaires...) traduite dans deux langues A et B;

- Si pour un concept donné les traductions en A et en B se ressemblent (en fait si leurs formes phonétiques sont telles qu'on pense qu'elles proviennent d'une origine étymologique commune), on compte un point de ressemblance (ou zéro de différence); si à l'inverse, aucune parenté ne peut être décelée entre ces deux traductions, on compte zéro de ressemblance (ou un point de différence);

- Le pourcentage de différence obtenu sur l'ensemble de la liste de concepts fournit la mesure de la dissimilarité (ou de l'inverse, la similarité) entre A et B;

- On peut ainsi calculer toutes les dissimilarités entre les langues prises deux à deux, et les ranger dans une matrice carrée symétrique à diagonale nulle du type de celle qui indique les distances kilométriques entre les villes d'un pays;

- A partir de cette matrice de dissimilarité, en se fondant sur les procédures mises au point par la taxinomie numérique[6], il est possible d'obtenir une classification des langues par regroupements successifs hiérarchisés (sous-groupes de langues, groupes, sous-familles, familles, etc...), représentable par un arbre dont les noeuds auront des ordonnées proportionnelles à la dissimilarité des entités qu'ils regroupent.

II - OBJECTIFS

Les travaux de lexicostatistique ont souvent fait usage de moyens informatiques. On conçoit aisément en effet, que décompter les pourcentages de dissimilarité entre deux langues, remplir la matrice de dissimilarité pour l'ensemble des langues entrant dans la comparaison, et en dernier lieu appliquer l'algorithme de regroupement hiérarchique soient des tâches que le caractère fastidieux et répétitif rend justifiable d'un traitement informatique. Mais jusqu'à présent, la phase initiale et cruciale de la démarche lexicostatistique qui est celle de l'établissement des jugements de ressemblance a toujours été accomplie à la main: c'est le linguiste qui, au vu des deux formes phonétiques, décrète leur similarité ou au contraire leur dissimilarité, en fonction de ce qu'il connaît des lois phonétiques en général et de celles qui sont attestées en particulier dans l'aire où la famille linguistique est étudiée.

L'originalité de COLEXA, c'est précisément l'automatisation des jugements de ressemblance, pour résoudre le problème de masse que pose la comparaison deux à deux d'un nombre élevé de listes de 120 mots, mais aussi pour tenter d'éliminer par l'automatisation de cette procédure, la part de subjectivité qui subsiste dans toute activité de comparaison à la main et qui peut biaiser les résultats.

COLEXA doit donc être capable de simuler l'activité du linguiste qui émet un jugement de ressemblance sur deux mots donnés. Or cette activité n'est en aucune manière une opération simple: elle met en jeu une "**expertise**" qui ne peut se réduire à l'application mécanique de règles explicites. Beaucoup d'éléments semblent entrer en jeu, différents selon les divers cas de figure, certains aisément quantifiables, d'autres non, et leurs poids respectifs dans la décision finale ne sont pas précisés... D'où l'idée de concevoir un système qui vise à analyser cette expertise. Cette analyse se fondera sur un échantillon de jugements portés à la main par le linguiste sur des paires de formes tirées aléatoirement parmi toutes les paires à comparer. Elle mettra au jour des corrélations entre, d'une part un certain nombre d'indices "**objectifs**" caractéristiques des formes comparées et d'autre part les jugements associés. En d'autres termes le système va chercher comment prédire au mieux le jugement des linguistes au vu d'indices que pour chaque paire il sait calculer. La méthode de prédiction mise au point sur l'échantillon de paires jugées à la main sera appliquée à l'ensemble des données.

B - DESCRIPTION DE COLEXA

Un concept fondamental sera appelé **item**. Chaque item aura autant de formes qu'il y a de langues dans le groupe considéré. Chaque forme est la traduction de cet item dans l'une des langues. Comme concepts fondamentaux on peut citer: bouche, manger, arbre, boire, oeil, etc... Un item est donc un objet ou une notion simple exprimable dans toutes les langues étudiées.

Pour écrire un item, les linguistes utilisent des signes qui ne sont pas uniquement constitués des 26 lettres de l'alphabet (dans l'un des groupes de langues étudié au cours de ce travail on utilise 19 voyelles et 143 consonnes). Ces signes que nous appelons ici **segments** ne se retrouvent pas tous sur un clavier d'ordinateur. Pour permettre la saisie de ces formes sur un ordinateur, une codification simple a été introduite. Dans cette codification chaque segment est écrit sur deux positions. la première position contient l'une des 26 lettres de l'alphabet français (les voyelles a,e,i,o,u; les semi-voyelles w et y; les consonnes) et la deuxième contient un chiffre de la base 10 (0 à 9).

Un segment est une voyelle si sa première position contient une voyelle. Un segment est une consonne si sa première position contient une consonne. Un segment est une semi-voyelle si sa première position contient une semi-voyelle.

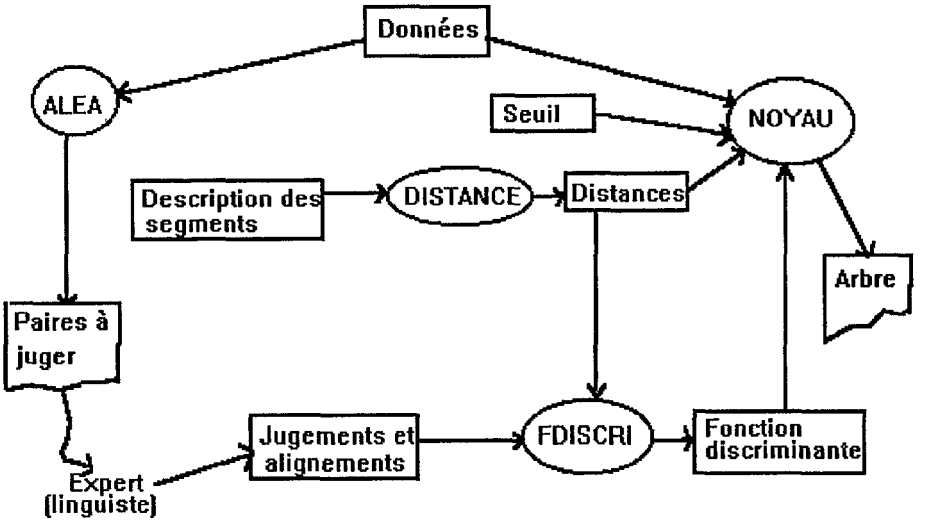
Exemples: Le segment **mb** est codé **B5** (consonne)
Le segment **mpf** est codé **P7** (consonne)
Le segment **gbw** est codé **W3** (semi-voyelle)
Le segment **ü** est codé **U1** (voyelle)

Quand le chiffre qui suit la lettre est nul, on peut le remplacer par un espace vide. Ainsi le segment **T0** peut aussi s'écrire **T** suivi d'un espace.

Pour faire la comparaison de deux formes nous mettons face à face les différents segments composant ces deux formes. Nous appellerons cette opération un alignement. Supposons que nous voulons comparer les formes **efaplos** et **pabilohi**. L'alignement peut se faire de la façon suivante:

. p a b i l o h i
e f a p . l o s .

Dans l'alignement on évite de mettre face à face une voyelle et une consonne. C'est ainsi que dans l'alignement ci-dessus la voyelle **e** n'a aucun segment en face d'elle. Nous dirons dans ce cas qu'il y a un **trou**. Il en est de même des deux voyelle **i**. Dans l'alignement proposé il y a donc trois trous. Une semi-voyelle peut se mettre en face d'une voyelle, d'une consonne ou d'une autre semi-voyelle. Dans l'alignement on évite aussi de mettre deux trous consécutifs.



SCHEMA GENERAL DE COLEXA

Ce schéma montre une vue générale simplifiée du système COLEXA. Il est constitué de quatre modules d'inégale importance représentés sur le schéma par quatre ronds (ALEA, FDISCRI, DISTANCE et NOYAU). Ces différents modules sont décrits dans les lignes qui suivent. Les informations à fournir et les informations passant d'un module à l'autre sont dans un rectangle. Le module NOYAU imprime en sortie un arbre.

Les données en entrée sont constituées des 120 formes pour chacune des langues du groupe à étudier. Elles sont généralement présentées langue par langue (d'abord les 120 formes de la première langue, puis les 120 formes de la deuxième langue et ainsi de suite). Ces données en entrée sont fournies à ALEA et au NOYAU. La description des différents segments est fournie au module DISTANCE.

Les trois premiers modules préparent l'environnement de travail du dernier qui constitue le noyau même du système.

I - MODULE ALEA

Le module ALEA reçoit en entrée les différentes formes correspondant aux items pour toutes les langues. S'il y a par exemple N langues il y aura **Nx120 formes**. Avec N langues et 120 items par langue, il faudra faire **C=120xNx(N-1)/2** comparaisons entre deux formes. ALEA numérote ces paires de **1** à **C** et effectue un choix aléatoire de m entiers compris entre **1** et **C**. Pratiquement

ALEA produit une liste de **m triplets (x,y,z)** où x est le numéro de l'item (1 à 120) et y et z sont des numéros de langues (1 à N). Chaque triplet désigne les deux formes de l'item x, l'une des formes appartenant à la langue y et l'autre à la langue z.

Les m paires ainsi désignées par ALEA seront manuellement jugées par le linguiste et ce jugement sera retransmis à FDISCRI (module décrit plus loin) avec l'alignement qui le motive.

Si ALEA fournit par exemple **(50,4,13)** parmi les m triplets, ceci voudra dire que le linguiste doit comparer les formes de l'item **50** dans les langues numéros **4** et **13** et fournir son jugement sur cette paire à FDISCRI ainsi que l'alignement sur lequel il fonde ce jugement.

II - MODULE DISTANCE

Le module DISTANCE établit à partir des informations fournies par le linguiste[9] et concernant la description des différents segments, les distances entre les consonnes, les distances entre les voyelles et les distances à zéro. Les distances ainsi calculées sont rangées dans des tables et serviront d'une part au module FDISCRI pour déterminer les caractéristiques d'une paire, d'autre part au noyau même de COLEXA pour élaborer son jugement.

Pour le calcul des distances, les linguistes décrivent les consonnes en fournissant pour chacune:

- le **voisement** (voisé ou non voisé)
- la **nasalité** (nasal ou non nasal)
- le **mode d'articulation** (explosive, fricative, affriqué, sonnante, ...)
- le **lieu d'articulation** (labial, dental, palatale, vélaire)

Pour les voyelles les linguistes fournissent:

- le **degré d'aperture** (fermé, mi-fermé, mi-ouvert, ouvert)
- la **localisation** (antérieure, centrale, postérieure, arrondie ou non)

Ces caractéristiques sont appelées des dimensions. Les linguistes définissent des distances entre les différentes valeurs d'une dimension donnée, et la distance globale entre deux segments est obtenue en faisant la somme des distances qui les séparent sur chacune de leurs dimensions.

Si pour les 19 voyelles il était pensable de procéder à un calcul manuel et d'entrer directement dans l'ordinateur la table des distances entre voyelles, il n'en va pas de même pour les 143 consonnes qui conduisent au calcul de 10153 distances. Chaque segment a été décrit par un vecteur (7 chiffres pour une consonne et 5 pour une voyelle) qui traduit sa définition sur les différentes dimensions. Le calcul de la distance entre deux segments s'est alors fait automatiquement.

La distance entre deux segments est un entier positif, nul si les deux segments sont identiques, non nul dans les autres cas. Cette distance concerne des segments de même nature (deux consonnes ou deux voyelles). Aucune distance n'est définie entre une consonne et une voyelle. Plus la distance est grande, plus les deux segments sont loin l'un de l'autre et plus il est difficile qu'au cours du temps l'un des segments évolue pour devenir l'autre.

Le module DISTANCE donne par exemple les résultats suivants:

- la distance ente **B1** et **C**1 est de **7**
- la distance entre **B1** et **P1** est de **1**
- la distance entre **C4** et **B7** est de **10**
- la distance entre **R** et **L** est de **1**
- la distance entre **M** et **P** est de **3** etc...

Ces exemples montrent qu'il est plus facile qu'au cours du temps le segment **B1** évolue et devienne **P1** (ou inversement). Il est par contre beaucoup plus difficile que le segment C4 devienne B7.

Le module DISTANCE donne ces distances entre consonnes sous forme d'une matrice carrée symétrique d'ordre 143 qui matériellement dans le système est représentée par un vecteur.

III - MODULE FDISCRI

Le module FDISCRI établit une fonction discriminante[1, 2] à partir des différents jugements du linguiste sur les paires choisies au hasard par le module ALEA décrit plus haut. Pour chaque paire le linguiste a fourni non seulement son **jugement** (semblable ou non) mais aussi **l'alignement** à partir duquel il fonde ce jugement La fonction ainsi obtenue dépend donc des caractéristiques de l'alignement. Les caractéristiques retenues sont:

- Nombre de couples de segments appariés, sans compter les appariements avec un trou (NST)

- Nombre de trous au total (**NTT**)
- Nombre de trous à l'initial de l'alignement (**NTI**)
- Nombre de trous ailleurs qu'à l'initial (**NTNI**)
- Nombre de couples de consonnes appariées (**NC**)
- Nombre de couples de voyelles appariées (**NV**)
- Somme des distances entre les segments de tous les couples de consonnes (**DC**)
- Somme des distances entre les segments de tous les couples de voyelles (**DV**)
- Distance entre les segments du premier couple de consonnes appariées (**DPC**)
- Distance à zéro des segments appariés aux trous initiaux ,ou poids du trou initial (**DTI**)
- Poids des trous non initiaux (**PTNI**)
- Rapport du nombre de trous sur le nombre de couples de segments appariés (**RT**)
- Distance moyenne entre les consonnes (**DMC**)
- Distance moyenne entre les voyelles (**DMV**)

Il y a au total 14 valeurs caractérisant un alignement. Les six premières valeurs s'obtiennent par simple comptage des segments selon leur nombre et leur type (consonne, voyelle et trou). Les trois valeurs suivantes (DC, DV et DPC) sont calculées à l'aide des tables de distances entre consonnes et entre voyelles. Les valeurs DTI et PTNI sont lues dans la table des distances à zéro. Les trois dernières valeurs sont obtenues à partir des 9 premières de la façon suivante:

RT=NTT/NST
DMC=DC/NC
DMV=DV/NV

Le module FDISCRI utilise une méthode d'analyse discriminante pour déterminer parmi toutes les valeurs caractéristiques celles qui sont plus fortement liées au jugement du linguiste. Il construit une fonction linéaire de ces variables satisfaisant aux deux conditions suivantes:

- Si pour une paire donnée la valeur de cette fonction est négative, les deux formes constituant la paire se ressemblent.

- Si pour une paire donnée la valeur de cette fonction est positive, les deux formes constituant la paire ne se ressemblent pas.

Un essai a été réalisé avec un échantillon de 600 alignements tiré (par le module ALEA) de 24 langues de la famille tchadienne assortis du jugement émis par le linguiste. L'analyse discriminante a permis de dégager les critères les plus

discriminants entre les deux groupes (paires de formes qui se ressemblent et paires de formes ne se ressemblant pas). Ce sont les cinq critères suivants:

DMC distance moyenne entre les consonnes
DMV distance moyenne entre les voyelles
DC distance consonantique totale
NC nombre de couple de consonnes appariés
NTT nombre total de trou

Les pourcentages de bien classés (parmi les 600 paires) par l'utilisation de ces cinq variables seulement sont de 79% pour les non ressemblants et 88,8% pour les ressemblants. Il y a un seul axe factoriel discriminant dans lequel l'histogramme de la variable canonique sépare les deux groupes: non ressemblants à droite, ressemblant à gauche.

Le premier critère, distance moyenne des consonnes a un pouvoir discriminant très supérieur à ceux des autres. Des valeurs très élevées de **DMC**, **DMV**, **DC** et **NTT** correspondent aux paires non ressemblantes. La fonction discriminante a la forme suivante:

$$F = -1{,}27 + 0{,}62NTT - 0{,}62NC + 0{,}25DC + 0{,}17DMC + 0{,}66DMV$$

On notera que le pourcentage de bien classés est de près de 10 points supérieur pour les ressemblants à celui des non ressemblants. C'est dire que la simulation sera d'autant meilleure qu'il y aura un plus fort taux de ressemblance entre les langues à comparer. Ce qui confirme l'intérêt qu'il y a à ne soumettre au système que des données relativement homogènes (des langues d'une même famille par exemple).

On notera aussi qu'il est plus grave d'être trop laxiste, c'est -à-dire de juger ressemblante une paire non ressemblante, que d'être trop sévère, c'est-à-dire de juger non ressemblante une paire ressemblante. En effet, l'application du principe de transitivité de la ressemblance risque de répercuter sur d'autres jugements l'erreur par laxisme, alors que l'erreur par excès de sévérité a moins de probabilité de se répercuter, et peut en outre être corrigée par l'application de la transitivité.

Dans ces conditions on peut modifier légèrement la fonction discriminante dans le sens de la sévérité: pour qu'une paire soit jugée ressemblante on exigera non plus simplement $F<0$ mais $F<\alpha<0$, c'est-à-dire F nettement négative. Le seuil α peut

être modulé après examen du résultat du test sur l'échantillon, d'après les différentes familles.

La fonction discriminante ainsi calculée et éventuellement modifiée par un seuil sera fournie au noyau de COLEXA pour déterminer son jugement sur une paire quelconque de formes.

IV - NOYAU DE COLEXA

Le noyau de COLEXA reçoit en entrée les formes des différentes langues du groupe (données), la fonction discriminante, les tables de distances (entre consonnes, entre voyelles et à zéro) et le seuil α. Il fournit en sortie l'arbre associé à la classification obtenue.

Le traitement effectué par le noyau de COLEXA se décompose en deux grandes parties:

- la comparaison des formes lexicales attestées dans les différentes langues item par item, qui aboutit à la construction de la matrice d'équivalence;

- la comparaison des langues entre elles, qui conduit à leur classification c'est-à-dire à la construction de l'arbre.

1 - Construction de la matrice d'équivalence

Considérons le premier des 120 items dont nous possédons la traduction dans chacune des langues à comparer . Limitons-nous à huit langues par exemple.

L1	L2	L3	L4	L5	L6	L7	L8
tul	**hed**	**tsul**	**sul**	**tat**	**yuu**	**dzul**	**yoo**

Dans cet exemple **tul** est la forme de **l'item 1** dans la langue **L1**, **hed** est la forme du même item dans la langue 2 etc... Nous avons éliminé les chiffres dans chaque segment pour faciliter l'explication.

Dans cette phase du traitement, tout le travail de COLEXA consiste à transformer cette séquence de huit formes linguistiques en une séquence de huit nombres (c'est-à-dire un vecteur) telle qu'à deux formes jugées ressemblantes correspond un même nombre et à deux formes jugées non ressemblantes des nombres différents.

Avant tout jugement de ressemblance, les huit formes sont supposées différentes les unes des autres, ce que l'on traduira par le vecteur de l'item 1 suivant:

1 2 3 4 5 6 7 8

Après le traitement, c'est-à-dire après que toutes les formes aient été comparées et jugées, le vecteur de l'item 1 aura par exemple la forme suivante:

1 2 1 1 5 6 1 6

Ceci traduit le fait que les formes **tul**, **tsul**, **sul** et **dzul** ont été jugées ressemblantes. Il en est de même des formes **yuu** et **yoo**. La forme **tat** ne ressemble à aucune autre forme.

Ce traitement va être effectué sur tous les 120 items et donnera pour chaque item un vecteur (**vecteur item**). L'ensemble de ces 120 vecteurs constitue la matrice d'équivalence (120 lignes et 8 colonnes dans cet exemple). Elle incorpore toutes les relations de ressemblance existant entre l'ensemble des formes de chacun des items. Nous l'appelons matrice d'équivalence parce que la relation de ressemblance entre deux formes linguistiques est une relation d'équivalence au sens mathématique sur l'ensemble des formes .

Si nous avons à comparer N langues, pour chaque item il faudra faire N(N-1)/2 comparaisons de deux formes avant d'avoir le vecteur de cet item. Le noyau de COLEXA doit donc effectuer 120N(N-1)/2 comparaisons de deux formes pour construire la matrice d'équivalence.

La comparaison de deux formes **f1** et **f2** est réalisée en plusieurs étapes:

a) - les deux formes **f1** et **f2** sont identiques, c'est-à-dire contiennent les mêmes segments dans le même ordre. Ces deux formes sont jugées ressemblantes, le vecteur item est modifié en conséquence et on passe à la comparaison suivante.

b) - les deux formes sont vraiment différentes étant donné les segments composant ces formes. on passe à la comparaison suivante sans modifier le vecteur item.

c) - On ne peut rapidement juger les deux formes (elles ne sont ni identiques, ni vraiment différentes), dans ce cas COLEXA construit une matrice dite d'alignement, recherche les alignements qui peuvent lui permettre de dire que les

deux formes sont ressemblantes. Pour chaque alignement il calcule les valeurs caractéristiques et détermine la valeur correspondante de la fonction discriminante. Si cette valeur est négative, les deux formes sont ressemblantes, le vecteur est modifié et on passe à la comparaison suivante. Sinon COLEXA recherche un autre alignement. S'il a épuisé les alignements, les deux formes sont dites non ressemblantes.

Ces opérations sont résumées dans l'algorithme ci-dessous où N désigne le nombre de langues à comparer, **Fi** et **Fj** deux formes, l'une appartenant à la langue numéro i et l'autre à la langue numéro j.

```
POUR k=1 JUSQU'A 120 FAIRE
      POUR I=1 jusqu'à N-1  FAIRE
            POUR j=I+1 jusqu'à N  FAIRE
                SI Fi identique à Fj ALORS DEBUT Fi ressemble à Fj;
                      modifier le vecteur de l'item K ;FIN;
                SINON
                   SI Fi pas très différent de Fj ALORS
                  DEBUT
                   Construire la matrice d'alignement;
                   juge:=0;
                     TANT QU'il y a alignement et juge=0 FAIRE
                           Prendre un alignement;
                           Calculer ses valeurs caractéristiques;
                           Calculer la valeur F de la fonction
                             discriminante;
                           SI F<α ALORS juge:=1;
                      FIN TANT QUE
                    SI juge=1 ALORS modifier le vecteur  de l'item k;
                    FIN;
            FIN POUR;
      FIN POUR;
FIN POUR ;
```

En pratique COLEXA n'effectue pas toutes les 120N(N-1)/2 comparaisons. Il exploite judicieusement les formes identiques pour éliminer les comparaisons inutiles.

2 - Construction de l'arbre

Cette dernière partie permet de passer de la matrice d'équivalence construite précédemment à la matrice de dissimilarité, puis de la matrice de dissimilarité à sa représentation sous forme d'arbre. A ce niveau, toutes les données sont sous forme de nombre. Il n'y a plus de forme ou de segment.

Chaque langue **Li** est caractérisée par le vecteur que constitue la ième colonne de la matrice d'équivalence. La mesure de la dissimilarité entre deux langues **Li** et **Lj** consiste à comparer les colonnes **i** et **j** de la matrice d'équivalence. Si la **n** ième composante de la colonne **i** est différente de la **n** ième composante de la colonne **j**, on compte **1** point de dissimilarité. Si les deux composantes sont égales, on compte **0** point de dissimilarité. On procède ainsi pour les valeurs de n de 1 à 120 en additionnant les points de dissimilarité. Le score total, ramené à un pourcentage, est le taux de dissimilarité entre **Li** et **Lj**.

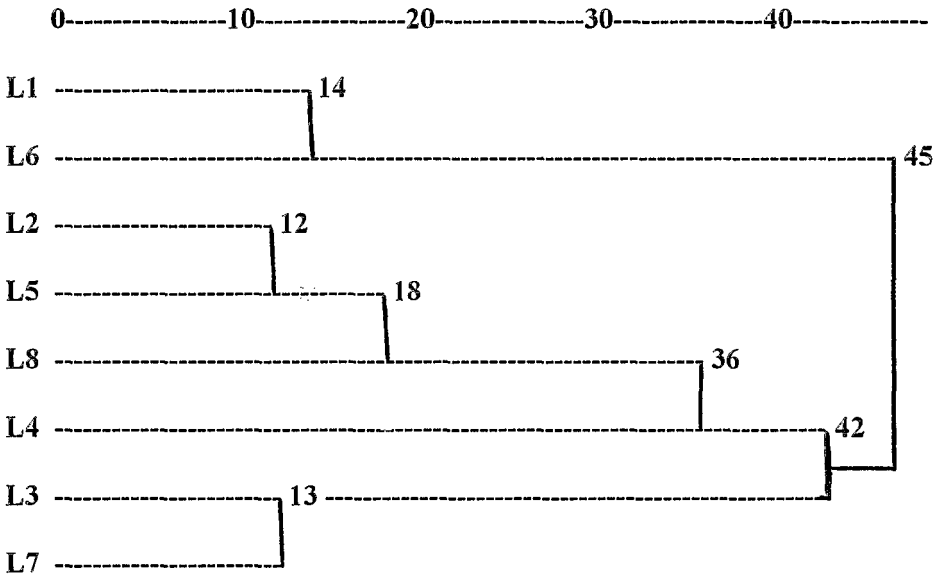
Le calcul du taux de dissimilarité est effectué pour chaque couple Li et Lj. On obtient ainsi une matrice carrée NxN symétrique à diagonale nulle. Cette matrice est déjà plus aisément interprétable que la matrice d'équivalence. Ainsi en suivant la ligne **i** et la colonne **i** on peut déterminer quelle est la langue la plus voisine de **Li**.

La figure ci-dessous montre une matrice de dissimilarité sur un groupe de **8** langues désignées par **L1, L2, ... L8**. Cette matrice montre par exemple que la langue la plus proche de **L2** est **L5**.

	L1	L2	L3	L4	L5	L6	L7	L8
L1	**0**	**50**	**70**	**25**	**45**	**14**	**75**	**55**
L2		**0**	**35**	**40**	**12**	**38**	**60**	**16**
L3			**0**	**38**	**35**	**42**	**13**	**46**
L4				**0**	**30**	**30**	**46**	**33**
L5					**0**	**31**	**42**	**20**
L6						**0**	**47**	**52**
L7							**0**	**44**
L8								**0**

En appliquant le principe de la taxinomie numérique[6,9] à la matrice de dissimilarité, on arrive à une représentation sous forme d'arbre. La méthode de groupement hiérarchique employée est de type agglomératif et procède par pas

successifs. La dissimilarité entre deux groupes de langues est calculée en utilisant la méthode de la moyenne [9]. L'arbre correspondant à la matrice ci-dessus est le suivant:



Cet arbre permet de voir qu'il y a trois sous-groupes de langues dans le groupe étudié. le premier est composé des langues **L1 et L6**; le second est composé des langues **L2, L5, L8 et L4** (**L4** étant un peu plus éloignée des trois premières) et le dernier est composé des langues **L3 et L7**. La première ligne donne l'échelle. Elle permet de voir que les langues **L1 et L6** ne diffèrent que de 14 %. C'est une information qu'on pouvait aussi lire dans la matrice de dissimilarité donnée plus haut.

V - ETAT D'AVANCEMENT DU LOGICIEL

Le système COLEXA a été mis au point sur l'ordinateur IBM 4331 du Centre de Calcul de l'Université de Yaoundé. Il a été utilisé par les linguistes sur des groupes de langues tchadiques particulièrement et les résultats été très satisfaisants.

Ce premier travail a été fait en FORTRAN et sur gros ordinateur. Actuellement nous sommes entrain de mettre ce logiciel sur micro-ordinateur pour une plus large diffusion. La version micro-ordinateur de COLEXA sera plus proche de l'utilisateur donc plus conviviale et facilement transportable.

Dans la version micro-ordinateur de COLEXA il y aura entre autre un module de démonstration de l'enchaînement des différentes opérations concernant la comparaison de deux formes (formes identiques, presque identiques, formes très différentes, matrice d'alignement, alignement, valeur de la fonction discriminante et jugement). Le linguiste peut alors tester à l'écran le comportement du système à partir de couples de formes de son choix. Il introduit deux formes et COLEXA montre le cheminement pas à pas dans la comparaison jusqu'à son jugement final qui peut utiliser un alignement, plusieurs alignements, ou pas d'alignement du tout.

CONCLUSION

Ce travail a été mené avec un double souci: l'adaptabilité et la compréhensibilité ou l'explicabilité, deux critères principaux pour reconnaître un raisonnement intelligent [10]. Si le logiciel a été conçu par des chercheurs au Cameroun, il est par son module FDISCRI exploitable dans beaucoup de groupes de langues (utilisant une écriture en segments qu'on peut décrire par un vecteur). Le rôle d'apprentissage est joué ici par le module de démonstration de l'enchaînement des différentes opérations signalé plus haut dans la version micro-ordinateur de COLEXA.

BIBLIOGRAPHIE

[1] **F. CAILLIEZ et J.P. PAGES**; Introduction à l'analyse des données, 1976, SMASH, Copédith

[2] **CH. BASTIN, J.P. BENZECRI, Ch. BOURGARIT, P. CAZES**; Pratique de l'analyse des données, Tome 2, Abrégé théorique; 1980, DUNOD

[3] **J.P. BENZECRI & COLLABORATEURS**; Pratique de l'analyse de données, Tome 3, Linguistique et lexicologie, 1981, DUNOD

[4] **E. DIDAY, J. LEMAIRE, J. POUGET et F. TESTU**; Eléments d'analyse de données, 1982, DUNOD

[5] **DIEU M. P. RENAUD et ALII**; Atlas Linguistique de l'Afrique Centrale, Structures et Méthodes, Agence de Coopération Culturelle et Technique, Paris, Centre Régional de Recherche et de Documentation sur les Traditions Orales et pour le Développement des Langues Africaines, Yaoundé. 1983

[6] **J.P. BENZECRI & COLLABORATEURS**; L'analyse de données Tomme 1, la Taxinomie, 1984 DUNOD

[7] **C.H. DOMINE**; Techniques de l'intelligence artificielle, 1988, DUNOD INFORMATIQUE

[8] **M. LE SEAC'H**; Développer un système expert, méthodes et exemples, 1989, édiTESTS

[9] **M. DIEU, J.P. NZALI, M.H. NGOA**; COLEXA, un système expert qui compare et classe des mots et des langues; 1990; document interne.

[10] **Y. KODRATOFF**; Le débat sur la définition de l'IA, BULLETIN DE L' AFIA, Octobre 1993, n° 15, pages 48 et 49