M1103 - ADA
Projet "Classification automatique" - Partie 2

1. Génération automatique de lexiques

Dans notre système de classification de dépêches (partie 1 du projet), nous avons utilisé des lexiques (un par catégorie) que nous avons construits manuellement. L'objet de la seconde partie du projet est de construire ces lexiques automatiquement. Contrairement à la première partie du projet, vous aurez une certaine liberté dans la définition de la méthode. Le but du jeu étant de trouver la méthode donnant les meilleurs résultats.

2. Description générale de la méthode de construction

La méthode que nous proposons pour construire automatiquement les lexiques consiste à analyser automatiquement le fichier ***depeches.txt*** (ne surtout pas utiliser ***test.txt*** gardé pour les tests) en vue d'extraire pour chaque catégorie, les mots les plus représentatifs. La méthode consiste plus précisément à calculer un score pour tous les mots présents dans au moins une dépêche de la catégorie. Ce score est fonction de la fréquence du mot dans la catégorie (c'est-à-dire le nombre de fois où le mot apparaît dans des dépêches de cette catégorie) et de sa spécificité (qui dépend du nombre de fois où le mot apparaît dans une dépêche d'une autre catégorie).

Ce score peut par exemple, être simplement : *le nombre de fois où le mot apparaît dans des dépêches de la catégorie - le nombre de fois où le mot apparaît dans des dépêches d'autres catégorie* (ce qui peut donc donner un score négatif). Mais vous pouvez imaginer autre chose. Sur la base du score de chacun des mots, pour une catégorie donnée, on décidera d'ajouter ou non le mot au lexique de cette catégorie et d'attribuer le cas échéant un poids de 1, 2 ou 3 au mot. Là encore, à vous de décider la façon d'attribuer les poids.

3. Développements

3.1 Initialisation d'un vecteur de mot/score

a) Déclarez dans **p_generation.ads** :

Les types suivants :

```
type TR_Entree is record
        mot : String(1..30);
        score: integer;
end record;

type TV_Dico is array(1..2000) of TR_Entree;
```

l'en-tête de la fonction suivante :

function Recherche(VM: in Tv_Dico; N: in Integer; M:String) return Integer;

{ } => {resultat = indice du mot ***M*** dans le vecteur ***VM*** si il est présent et -1 sinon.
N est le nombre de mots rangés dans le vecteur}

et l'en-tête de la procédure suivante :

procedure Init_Dico(VD: in Tv_Depeche;C : in T_Categorie; VM: out Tv_Dico; N : out Integer);

{ } => {Charge dans ***VM*** tous les mots présents dans au moins une dépêche de la catégorie ***C*** du vecteur de dépêches ***VD.*** Attention, même si le mot est présent plusieurs fois, il ne doit apparaître qu'une fois dans le vecteur ***VM***. La procédure initialise aussi tous les scores à 0 et range dans ***N*** le nombre de mots ajoutés dans ***VM*** }

b) Dans **p_generation.adb** *écrivez le corps de la fonction et de la procédure*

c) Testez ces fonctions/procédures dans un programme principal **p6.adb** *affichant notamment le contenu de VM après initialisation*

3.2 Calcul des scores des mots

a) Déclarez dans **p_generation.ads** :

procedure Calcul_Scores(VD: in Tv_Depeche;C: in T_Categorie; VM: in out Tv_Dico;N: in Integer);

{ } => {Cette procédure met à jour les scores des différents mots présents dans ***VM.*** Lorsqu'un mot présent dans ***VM*** apparaît dans une dépêche du vecteur ***VD***, son score est décrémenté si la dépêche n'est pas dans la catégorie ***C*** et incrémenté si la dépêche est dans la catégorie ***C*** }

b) Dans **p_generation.adb** *écrivez le corps de la procédure*

c) Testez cette procédure dans un programme principal **p7.adb** *affichant le contenu de VM après initialisation et calcul des scores*

3.3 Attribution d'un poids en fonction d'un score

a) Déclarez dans **p_generation.ads** :

L'en-tête de la fonction suivante :

function Poids_Score (S : in Integer) return Integer ;

{ } => {resultat = valeur du poids à attribuer étant donné un score ***S***}

b) Dans **p_generation.adb** *écrivez le corps de cette fonction*

3.4 Génération d'un fichier lexique

a) Déclarez dans **p_generation.ads** :

procedure Generation_Lexique (VD: in Tv_Depeche;C : in T_Categorie; Fl : in String);

{ } => {Cette procédure créé *pour la catégorie* ***C*** le fichier lexique de nom ***Fl*** à partir du vecteur de dépêches de nom ***VD***. Cette procédure doit déclarer un vecteur de type ***TV_Dico*** puis le remplir en utilisant ***Init_Dico***, puis ***Calcul_Scores*** et enfin utiliser le vecteur résultant pour créer un fichier lexique en utilisant la fonction ***Poids_Score***}

b) Dans **p_generation.adb** *écrivez le corps de la procédure*

c) Dans un programme principal **p8.adb**, *Appelez 5 fois la procédure* ***Generation_Lexique*** *pour générer les lexiques de chaque catégorie.*

3.5 Utilisation des lexiques générés automatiquement

Les fichiers lexiques générés automatiquement peuvent maintenant être utilisés par le programme de classification écrit dans la partie 1 du projet. Testez les sur le fichier ***test.txt*** et comparez les résultats obtenus avec ceux initialement obtenus avec les lexiques que vous aviez remplis manuellement. Expérimentez différents calculs de score (*Calcul_Scores*) et de poids(*Poids_Score*) en vue d'obtenir les meilleurs résultats possibles.

3.6 Optimisation (facultatif)

Tous les vecteurs utilisés dans ce projet ne sont pas triés. Pourtant, le tri des vecteurs de lexiques et la réécriture des procédure d'accès en tenant compte accélèrerait les traitements (recherche dichotomique par exemple). Complétez vos programmes dans cet objectif et évaluez le gain (facteur d'accélération). Pour calculer les durées d'exécution, on utilise le package *calendar*. Prenez exemple sur *temps_execution.adb*.

3.7 Ce qu'il faut rendre

A l'issue de ce projet vous devez rendre :

- un petit rapport (format pdf) de 4 à 5 pages comportant :
  - une introduction présentant le sujet
  - un point sur ce que vous avez réalisé (ce que vous avez fait, ce que vous n'avez pas fait)
  - une présentation des résultats obtenus et une discussion de ces résultats.
  - une conclusion sur les perspectives d'amélioration de votre système de classification.
- vos programmes
- les lexiques construits manuellement
- les lexiques construits automatiquement
- le fichier réponse pour les lexiques construits manuellement
- le fichier réponse pour les lexiques construits automatiquement

Il suffira de placer tout cela dans votre dossier *ProjetAP123* et de taper la commande *cd* puis *fin-projet-ap123*

3.8 Présentation Orale

Lors de la dernière séance vous ferez une petite présentation orale de votre projet (5-10 minutes). Prévoyez un programme principal enchaînant la génération automatique des lexiques et la classification. Ce programme devra afficher les résultats de la classification et les temps d'exécutions (voir *temps_execution.adb*) de la génération des lexiques et de la classification. Votre enseignant testera ensuite par quelques questions votre maîtrise du projet.