Module M1103 - ADA
Projet "Classification automatique" - Partie 1

Lisez d'abord attentivement les sections 1, 2 et 3 pour comprendre l'objectif du projet. Vous serez guidé pour atteindre cet objectif dans les sections suivantes.

1. Tâche

L'objectif du projet est de créer un programme de classification automatique de dépêches. Une dépêche est un court texte correspondant à une information journalistique, telle que :

> Coupe de l'UEFA : l'OM victorieux, les Girondins défaits
> L'OM a remporté son match à domicile sur le score de 3 à 0. Bordeaux a perdu dans les arrêts de jeu.

Le programme devra attribuer à chaque dépêche, l'une des 5 catégories suivantes :

- ENVIRONNEMENT-SCIENCES
- CULTURE
- ECONOMIE
- POLITIQUE
- SPORTS

2. Principes de fonctionnement du programme

2.1 Un lexique par catégorie

Pour réaliser cette classification, le programme va s'appuyer sur des mots marquant l'appartenance à une catégorie particulière. Par exemple, les mots *match* et *joueur* semblent indiquer l'appartenance à la catégorie *SPORTS*. Ainsi, pour chaque catégorie, un lexique correspondant à un ensemble de mots "marqueurs" est créé. De plus, à chaque mot est associé un poids correspondant au degré avec lequel le mot indique l'appartenance à la catégorie. Trois poids sont possibles : 1, 2 ou 3. Un poids maximal de valeur 3 indique que le mot marque très fortement l'appartenance à la catégorie.

Par exemple, le lexique de la catégorie *SPORTS* pourrait être composé des mots et poids associés suivants : *sport, 3; coupe, 3; ;match, 3; joueur, 2; domicile, 1*. Dans la première partie du projet nous allons construire les lexiques manuellement. Pour cela, vous pourrez vous appuyer sur une centaine de dépêches données en exemple pour chaque catégorie. Il s'agira de choisir parmi les mots présents dans les dépêches d'une catégorie ceux qui vous paraissent les plus représentatifs de la catégorie, c'est-à-dire des mots spécifiques à cette catégorie (que l'on ne retrouve donc pas trop dans les autres catégories). Dans la seconde partie, nous construirons automatiquement les lexiques.

2.2 Calcul d'un score par catégorie

Pour chaque dépêche, un score est calculé pour chaque catégorie (ce qui fait donc 5 scores). Le score de la dépêche pour une catégorie particulière est calculé en faisant la somme du poids des mots présents dans le lexique de la catégorie. La catégorie attribuée à la dépêche est celle obtenant le score maximal (en cas d'égalité, on en choisit une parmi les meilleures). Par exemple, dans le cas de la dépêche donnée en exemple ci-dessus et si on considère le lexique de la catégorie *SPORTS* donné ci-dessus, le score de la dépêche pour la catégorie *SPORTS* est de *3(coupe)+3(match)+1(domicile)= 7*

3. Les fichiers et leurs formats

3.1 Les fichiers des dépêches

Les dépêches sont regroupées dans 2 fichiers ***depeches.txt*** et ***test.txt***. Le fichier ***test.txt*** ne doit pas être utilisé pour construire les lexiques, il servira aux tests de votre programme de classification. Dans chacun de ces fichiers, on trouve exactement 100 dépêches par catégories. Ces dépêches ont été récupérées sur le site du journal *Le Monde* dans les cinq catégories que nous avons énoncées ci-dessus. Chaque dépêche correspond à une suite de lignes dans un fichier texte qui se présentent comme suit :

```
.N 002
.D 140208
.C ENVIRONNEMENT-SCIENCES
.T Un test de prédisposition au cancer de la prostate suscite une
polémique. Délivré par Internet, uniquement à des médecins
selon l'entreprise, ce test est mis en vente au prix de 500 dollars (341
euros).
```

- La première ligne correspond au numéro de la dépêche (qui permet de l'identifier)
- La seconde correspond à la date à laquelle la dépêche a été récupérée.
- La troisième correspond à la catégorie de la dépêche
- Les lignes suivantes correspondent au texte de la dépêche.

Les dépêches sont séparées par une ligne vide.

3.2 Les fichiers lexiques

Cinq fichiers correspondant aux différents lexiques devront être créés à l'aide d'un éditeur de texte. Chaque ligne de ce fichier correspondra à un mot suivi de ":" et du poids associé au mot. Le contenu du lexique "SPORTS" pourrait par exemple avoir la forme suivante :

```
sport:3
coupe:3
match:3
jeu:2
domicile:1
```

3.3 Le fichier réponses

Un fichier donnant le résultat de la classification devra être généré automatiquement. Ce fichier aura impérativement la forme suivante :

```
001:ENVIRONNEMENT-SCIENCE
002:ENVIRONNEMENT-SCIENCE
003:ECONOMIE
...
500:SPORTS
ENVIRONNEMENT-SCIENCES:          89
CULTURE:                         93
ECONOMIE:                        68
POLITIQUE:                       80
SPORTS:                          76
MOYENNE :                        81.2
```

Les 500 premières lignes correspondent au numéro de la dépêche suivie de la catégorie attribuée par le programme. Les 5 lignes suivantes correspondent au nombre de réponses correctes (il suffit de comparer la réponse du programme à la catégorie réelle à laquelle la dépêche appartient). La dernière ligne correspond à la moyenne du nombre de réponses correctes toutes catégories confondues.

4. Organisation du travail

Le projet sera réalisé en binôme et donnera lieu à la rédaction d'un rapport et d'une petite présentation orale lors de la dernière séance. En plus de votre rapport, il vous faudra rendre les fichiers correspondant à vos programmes ainsi que les fichiers correspondants aux lexiques et aux résultats (voir le détail dans l'énoncé de la partie 2). Commencez par lancer le script ***debut-projet-ap123*** permettant la récupération du répertoire ***ProjetAP123*** contenant les fichiers qui vous seront utiles. Compilez tous les sources (*gnatmake *.adb*) et lancez l'exécutable ***p1***.

Parcourez les fichiers sources récupérés pour découvrir les différentes structures de données et les différentes fonctions/procédures mises à votre disposition. Les données permettant d'initialiser le vecteur des dépêches sont dans

le fichier texte ***depeches.txt*** que vous pouvez ouvrir avec n'importe quel éditeur de texte. Le fichier ***test.txt*** contient aussi des dépêches. Il sera exclusivement utilisé pour les tests à la fin des développements (5.6).

5. Développements

5.1 Création des fichiers lexiques

Commencez par créer un fichier lexique par catégorie. On pourra, dans un premiers temps, se limiter à une vingtaine de mots par lexique pour tester le programme et produire de premiers résultats. On pourra ensuite enrichir ces lexiques tout au long du projet. On construira ce lexique au regard des mots présents dans les dépêches de ***depeches.txt*** correspondant à la catégorie, mais on pourra aussi le compléter de mots à priori pertinents (pensez aux pluriels et en général aux différentes formes d'un mot). Le fichier ***test.txt*** qui contient aussi des dépêches ne doit pas être utilisé pour construire les lexiques, il sert uniquement aux tests de votre programme et ne doit être utilisé qu'à la fin (5.6).

5.2 Chargement des lexiques en mémoire et accès aux lexiques

a) Déclarez dans **p_lexique.ads** :

Les types suivants :

```
type TR_mot is record
        chaine : String(1..30);
        poids: positive;
end record;

type TV_lexique is array(Integer range<>) of TR_mot;
```

L'en-tête de la fonction suivante :

```
function Nb_Mots(nomfic : in String)  return Integer;
```

{ } => {résultat = nombre de lignes du fichier ***Nomfic*** et donc de mots dans le lexique correspondant}

Indication : La gestion des fichiers textes est décrite dans votre documentation technique. Vous pouvez aussi regarder le contenu de ***p_depeche.adb*** et en particulier le code de la procédure ***Charge***.

b) Dans **p_lexique.adb** *écrivez le corps de la fonction nb_mots*

c) Déclarez dans **p_lexique.ads** :

L'en-tête de la procédure suivante :

```
procedure Init_Lexique(Nomfic: in String; L : out Tv_lexique);
```

{On suppose que la taille du vecteur ***L*** correspond exactement au nombre de mots contenus dans le fichier}
=> {Range dans le vecteur ***L***, les mots contenus dans le fichier lexique ***Nomfic*** et les poids associés}

d) Dans **p_lexique.adb** *écrivez le corps de la procedure* **init_lexique**

Indication: Vous pourrez utiliser la fonction *index* (du package ***p_chaine***) qui retourne la position d'un caractère dans une chaine.

e) Déclarez dans **p_lexique.ads** :

L'en-tête de la fonction suivante :

```
function Poids_Mot(M  : in String; L : in Tv_lexique) return integer ;
```

{ } => {résultat = le poids de ***M*** dans le lexique ***L*** *et* ***0*** *si le mot n'est pas présent dans le lexique*}

f) Dans **p_lexique.adb** *écrivez le corps de la fonction* **Poids_Mot**

g) Dans un programme principal **p2.adb** :

- *Choisissez un de vos fichiers lexiques*
- *Déclarez 1 vecteur lexique (TV_Lexique) de la taille adéquate (utiliser la fonction Nb_Mots) en utilisant le mot clé* ***declare*** *(on ne connaît la taille du vecteur qu'au moment de l' exécution du programme)*
- *Appelez le chargement du lexique (Init_Lexique)*
- *Affichez le poids d'un mot saisi par l'utilisateur dans le lexique (utiliser la fonction Poids_Mot)*

5.3 Calcul du score d'une dépêche pour une catégorie

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

L'en-tête de la fonction suivante :

function Score(D: in TR_Depeche; L : in TV_Lexique) return Integer;

{ } => { résultat = score de la dépêche ***D*** pour la catégorie dont le lexique est ***L***}

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

c) Dans un programme principal **p3.adb** :

- *Choisissez un de vos fichiers lexiques*
- *Déclarez 1 vecteur lexique (TV_Lexique) de la taille adéquate (uiliser la fonction Nb_Mots) en utilisant le mot clé* ***declare*** *(on ne connaît la taille du vecteur qu'au moment de l' exécution du programme)*
- *Appelez le chargement du lexique (Init_Lexique)*
- *Déclarez 1 vecteur de depeche (TV_Depeche)*
- *Appelez le chargement du vecteur des dépêches à partir de depeches.txt (procédure charge déjà écrite)*
- *Affichez le score de différentes dépêches pour le lexique chargé et vérifiez le calcul manuellement.*

5.4 Recherche de la catégorie de score maximal

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

L'en-tête de la fonction suivante :

function Max_Score(VS : in TV_Score) return T_Categorie;

{ } => {resultat = Catégorie ayant le score maximal dans ***VS***}

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

c) Dans un programme principal **p4.adb :**

- *Déclarez un tableau de scores (TV_Score)*
- *Déclarez et chargez le vecteur des dépêches à partir de* ***depeches.txt***
- *Déclarez et chargez les vecteurs des lexiques*
- *Pour une dépêche remplissez le vecteur des scores*
- *Affichez la catégorie donnant le score maximal (en utilisant Max_Score)*

5.5 Création d'un fichier réponses

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

L'en-tête de la procédure suivante :

procedure Run (VD : in TV_Depeche;Lp,Ls,Lc,Le,Lt:TV_Lexique; Nomfic : String);

{ } => {Génère le fichier texte réponse ***Nomfic*** étant donné les dépêches et les lexiques passés en argument}

b) Dans **p_classification.adb** *écrivez le corps de cette procédure*

c) Dans un programme principal **p5.adb :**

- *Déclarez et chargez le vecteur des dépêches de depeches.txt*
- *Déclarez et chargez les vecteurs des lexiques*
- *Appelez la procédure Run*
- *Vérifiez le contenu du fichier réponses*

5.6 Tests et amélioration des résultats

Complétez et améliorez vos lexiques sur la base du contenu du fichier ***depeches.txt***. Notez les résultats obtenus en ayant remplacé ***depeches.txt*** par ***test.txt*** dans ***p5.adb***. Ne trichez pas, le fichier ***test.txt*** ne doit pas être utilisé pour construire les lexiques. Pour rendre la construction des lexiques automatique, passez à la partie 2 du projet.