COMPTE RENDU DÉTECTION AUTOMATIQUE DE LANGUE :

Sujet : Détection automatique de langue
Élèves : Aymeric AGARD, Maïawella FEVE, Alyson FONTAINE,
Nassim GHEMID et Emilien SPIQUEL -- BANDIN
Enseignants : Monsieur Créchet, Madame Herminier, Monsieur Pelletier
Etablissement : Marguerite de Navarre
Chercheur : NGUYEN Benjamin , INSA CENTRE VAL DE LOIRE

1. Intitulé du sujet

Proposer et implémenter un algorithme qui prend en entrée un texte en Français, Anglais, Espagnol ou Allemand et dit dans quelle langue le texte est écrit. On pourra essayer de donner également une barre d'erreur sur la prédiction.

2. Présentation du sujet

Nous avons tous déjà utilisé GOOGLE traduction ne serait-ce que pour nos devoirs de langues. En effet, il existe dans ce logiciel une fonction permettant de définir la langue rentrée directement lorsqu'on rentre un texte dans la barre de recherche. Nous voulions donc faire en sorte de pouvoir définir la langue d'un texte mais cette fois-ci sans utiliser l'outil préféré des élèves et à l'aide d'un programme informatique qu'on a codé sur Python.

3. Recherche de solutions

La première idée qui nous est venue est de comparer les fréquences d'apparition des différentes lettres dans le texte par rapport à une fréquence de référence. Nous avons eu d'autres idées, notamment repérer certains caractères spéciaux ou signes de ponctuations spécifiques à des langues ou encore comparer les fréquences d'apparitions des syllabes. Finalement, par manque de temps, nous nous sommes consacrés uniquement à la première piste, c'est-à-dire comparer les fréquences des lettres de l'alphabet en fonction de fréquences moyennes trouvées sur Internet.

langues	textes	A	B	C	D	E	F
	1	7,49%	0,60%	3,71%	4,11%	19,43%	0,63%
	2	9,46%	0,90%	3,64%	3,81%	17,01%	1,29%
	3	8,79%	0,86%	3,31%	4,34%	16,84%	1,37%
	4	9,47%	0,79%	2,84%	4,14%	15,43%	0,96%
Français	5	9,66%	0,68%	3,51%	4,00%	16,06%	1,17%

Exemple de fréquences d'apparition de quelques lettres dans différents textes.

4. Calcul de fréquences

Nous nous sommes séparés en deux groupes. Le premier avait pour objectif de calculer les fréquences d'apparition des lettres dans 10 textes pour chaque langue afin de créer une base de données pour le second groupe.

Nous avons créé un programme python pour cela. Toutes les lettres de l'alphabet ont été renseignées dans un dictionnaire. Une variable "texte" a été créée, on y renseigne le texte choisi. Toutes les lettres majuscules sont converties en lettres minuscules. Nous avons utilisé la fonction "len()" pour calculer la longueur du texte entré. Une fois ces premières actions réalisées, le programme parcours chaque caractère du texte et ajoute un à la variable qui correspond à la lettre. Nous avons décidé de nous baser sur l'alphabet anglais, pour que cela corresponde au plus grand nombre de langues possibles. Ainsi, lorsque l'algorithme tombe sur un "ë", il ajoute un à la variable "e". Cependant, le calcul restait encore imprécis. En effet, les espaces et la ponctuation étaient pris en compte, ce qui faussait les résultats. Nous avons donc fait en sorte qu'ils ne le soient plus. Une fois tous les caractères de l'alphabet comptés, le programme divise le nombre d'apparition de chaque lettre par le nombre total de lettres du texte pour avoir la fréquence d'apparition. Et celui-ci nous l'affiche.

```
for position in range (longueurTexte):
    for i in range(26):
        if texte [position] == Alphabet[i]:
            Liste[i] = Liste[i] + 1
    if texte [position] == "ë":
        Liste[4] = Liste [4] + 1
    if texte [position] == "ä":
        Liste[0] = Liste [0] + 1
    if texte [position] == "ï":
        Liste[8] = Liste [8] + 1
    if texte [position] == "ö":
        Liste[14] = Liste [14] + 1
    if texte [position] == "ß":
        Liste[18] = Liste [18] + 2
    if texte [position] == "ü":
        Liste[20] = Liste [20] + 1
    if texte [position] == "á":
        Liste[0] = Liste [0] + 1
    if texte [position] == "í":
        Liste[8] = Liste [8] + 1
    if texte [position] == "ó":
        Liste[14] = Liste [14] + 1
    if texte [position] == "ú":
        Liste[20] = Liste [20] + 1
    if texte [position] == "ñ":
        Liste[13] = Liste [13] + 1
    if texte [position] == "é":
        Liste[4] = Liste [4] + 1
    if texte [position] == "ù":
        Liste[20] = Liste [20] + 1
    if texte [position] == "à":
        Liste[0] = Liste [0] + 1
    if texte [position] == "è":
        Liste[4] = Liste [4] + 1
    if texte [position] == "ê":
        Liste[4] = Liste [4] + 1
```

Partie du programme transformant les caractères spéciaux en caractères

5. Détermination de la langue (1ère partie)

Une fois ceci fait, le deuxième groupe a été cherché sur internet la fréquence d'apparition des lettres pour les quatre langues citées précédemment, bien que nous ayons commencé avec le Français et l'Anglais uniquement et nous les avons entrés dans un nouveau programme informatique qui va nous permettre de déterminer la langue du texte que nous lui rentrerons.

Pour se faire, il nous a d'abord fallu simuler à la main ce que notre code devra exécuter. Nous avons donc pris la lettre « a » pour lancer une simulation. Sa fréquence d'apparition en anglais est de 8.08% et en français celle-ci est de 8.15%. Nous supposons donc que dans le texte que nous venons de rentrer dans le programme, la fréquence d'apparition de la lettre « a » est de 7.00%, nous calculons donc l'écart entre le français et notre texte, puis entre l'anglais et celui-ci et celui qui a le plus petit écart remporte le point. Point qui est stocké dans une variable appelée « score » suivi de la langue à laquelle elle appartient : « score_français » et « score_anglais ».

Cependant, lorsque nous avons codé cela sur Python, nous avions un pourcentage de réussite très faible, le plus souvent notre programme nous indiquait dans presque la totalité des textes ; et peu importe la langue de celui-ci ; que la langue était l'anglais. Même lorsque nous

utilisions les textes de l'autre sous-groupe avec leurs fréquences, celles que le programme nous donnait étaient différentes.

6. Détermination de la langue (2ème partie)

Nous avons donc été amenés à trouver une autre solution pour que notre programme soit opérationnel. Et nous avons décidé de coefficienter nos fréquences.

Ainsi, si nous prenons l'exemple de la lettre « a » dont la fréquence dans le texte entré est de 8.12 %. Ce qui fait un écart de 0.03 % avec la fréquence de référence en anglais et un écart de 0.04 % avec la fréquence de référence en français. Le point sera donc attribué à l'anglais. Mais avec le coefficientage du point, le programme multiplie 1 par l'écart entre les fréquences de référence (ici, 8.12 % et 8.08 %) c'est-à-dire 0.07 % et par l'écart des écarts (c'est-à-dire l'écart entre 0.04 % et 0.03 %), ici 0.01 %. Le nombre ajouté à la variable score_anglais est donc 1*0.07*0.01 ce qui donne 0.0007.

Le programme répète l'opération avec chacune des lettres de l'alphabet, puis compare les scores de chaque langue. Celle qui est la plus grande et déterminée comme étant la langue du texte.

7. Conclusion

Avec cette nouvelle méthode, le programme a donné la bonne langue sur 100% des textes entrés. Cependant, il ne fonctionne que très rarement sur de simples phrases qui sont trop courtes pour lui et donc pas assez proches des fréquences de référence. Il faut donc que le texte entré fasse au moins une dizaine de ligne pour que le programme marche dans 100 % des cas.