Gaspard Palay / Recommandation auto et adaptative d'emojis

Gaspard Palay

29/01/2021

1. Critères d'évaluation

1. Comportement du Rmd lors de son exécution
2. Qualité de la rédaction du dossier
3. Didactisme et pertinence du dossier
4. Qualité et lisibilité du Rmarkdown
5. Qualité des explications du modèle mathématique

2. Lien vers le document commenté

En cliquant **ici**, vous trouverez le lien menant au GitHub de Olfa LAMTI hébergeant le fruit de sa réalisation.

3. Auteurs du document commenté

Le document évalué dans le cadre de ce rendu a été produit par Olfa LAMTI, étudiante en MSc Data Management à Paris School of Business.

4. Synthèse du document

Il s'agit d'un document résumant une thése portant elle même sur l'algorithme permettant la recommandation automatique et adaptative des emojis. Je trouve le sujet très interessant puisqu'il nous concerne tous au quotidien.

A travers ce document, l'autrice illustre son propos à l'aide d'un document Python. Elle comence par définir son sujet et présenter au lecteur l'interêt de sa production. L'autrice foiunit ses informations sur la base d'une thèse longue, dont elle extrait trois partie portant sur la prédiction automatique d'emojis.

L'autrice informe le lecteur que l'algorithme de génération d'emojis repose sur l'algorithme de Levensthein. Elle nous présente son fonctionnement en définissant la formule, puis étudie l'alogorithme en prenant l'exemple d'un mot, à davoir "chien" sur lequel elle va appliquer l'algotihme de Levenshtein. Elle finit par founrir au lecteur un tutoriel vidéo, expose les limite de la distance de Levenshtein et conclu son devoir en fournissant ses sources qui lui ont permi de le rédiger.

Après lecture de sa production, je comprends que : La distance de Levenshtein, développée par ceui portant ce nom, sert à mesurer la différence entre deux chaines de caractères. Cette distance est égale au nombre minimal de caractères qu'il faut supprimer, insérer ou remplacer pour passer d'une chaîne à l'autre.

L'algorithme de Wagner et Fischer permet de calculer la distance de levensthein entre deux chaines de caractères courtes. C'est un algorithme de programmation dynamique qui utlise une matrice de dimension $(n+1)\times(m+1)$ où n et m sont les dimensions des deux chaînes de caractères.

5. Extrait commenté des parties de code

L'autrice écrit son code par Python et importe une capture d'écran de celui ci sur le document qu'elle fournt au lecteur.

Elle fournit d'abbord l'algoritme de Levensthein

$$\text{lev}_{a,b}(i,j) = \begin{cases} \max(i,j) \\ \min \begin{cases} lev_{a,b}(i-1,j)+1 \\ lev_{a,b}(i,j-1)+1 \\ lev_{a,b}(i-1,j-1)+1_{(a_i \neq b_j)} \end{cases} \end{cases}$$

La formul rend une nombre positif ou nul.

```
entier DistanceDeLevenshtein(caractere chaine1[1..longueurChaine1],
                             caractere chaine2[1..longueurChaine2])

   // d est un tableau de longueurChaine1+1 lignes et longueurChaine2+1 colonnes
   // d est indexé à partir de 0, les chaînes à partir de 1
   déclarer entier d[0..longueurChaine1, 0..longueurChaine2]

   // i et j itèrent sur chaine1 et chaine2
   déclarer entier i, j, coûtSubstitution

   pour i de 0 à longueurChaine1
       d[i, 0] := i
   pour j de 0 à longueurChaine2
       d[0, j] := j

   pour i de 1 à longueurChaine1
      pour j de 1 à longueurChaine2
          si chaine1[i] = chaine2[j] alors coûtSubstitution := 0
                                     sinon coûtSubstitution := 1
           d[i, j] := minimum(
                                d[i-1, j  ] + 1,                // effacement du nouveau caractère de chaine1
                                d[i,   j-1] + 1,                // insertion dans chaine2 du nouveau caractère de chaine1
                                d[i-1, j-1] + coûtSubstitution  // substitution
                              )

   renvoyer d[longueurChaine1, longueurChaine2]
```

L'algorithme ci dessus mesure la distance de Levenshtein permettant de passer de la chaine de caractère 1 : CHIENS à la chaine de caractère 2 : NICHE

Afin de s'approprier l'algorithme, l'autrice construit une matrice.

6. Evaluation du travail suivant les 5 critères précités

1. Comportement du Rmd à l'exécution

L'autrice n'a pas choisi de travailler sur un RMD mais de fournir au lecteur un fichier python. Ce choix a été fait afin de réaliser plus facilement l'agorithme de Levensthein. Quoi qu'il en soit, le fichier s'execute proprement et sans erreurs.

2. Qualité de la rédaction du dossier

La rédaction de ce document est de bonne qualité. Il est bien structuré, et les étapes de pregression son respectées.

Elle suit cette logique de vouloir démontrer l'application de l'algotihme de Levensthein pour la prédiction et la recommandation adaptative et automatique d'emojis. Les titres sont mis en forme.

3. Didactisme et pertinence du dossier

La lecture de ce dossier est aisée et accessible. Les modèles mathématique sont bien illustrés. Le document suit une logique de démonstration et atteint son objectif. L'autrice appuie sa démonstration sur une matrice complémentaire à son algorithme.

L'autrice arrive à faire adhérer le lecteur à sa production. Le document a vocation à transmettre une connaissance et le but, pour ma part, est atteint.

4. Qualité et lisibilité du Python

Le fichier python est bien écrit.

Il est lisible et aéré.

Il s'agit d'un code bien réalisé.

5. Qualité des explications du modèle mathématique

Les applications sont simples, de qualités et maitrisées. l'autrice est allée au dela de la présentation de l'algorithme en s'appputant sur une démonstration d'une matrice afin de mesurer la distance entre les chaines de caractères CHIENS et NICHE. Elle a fait cela pour mieux se l'approprier et cela m'a aidé à comprendre le cheminement de l'algorithme initial.

7. Conclusion

Selon moi, il s'agit globalement d'un très bon travail.

L'autrice fournit un dossier recherché, documenté et accessible. Elle founrit des vidéos tutorielles complémentaires, étudie les limite de cet algorithme et intéresse le lecteur grâce à un sujet moderne et actuel.

On y apprend des choses nouvelles.

Vous retrouvez ce document sur mon **GitHub**.