TP 2
Résumé automatique
Session 212

1 INTRODUCTION

Dans ce TP, vous allez construire un programme qui construit le résumé d'un (ou plusieurs) texte(s). Ce logiciel va lire les textes à résumer à construire un résumé pour chaque texte. Votre logiciel devra aussi, sur demande, pouvoir évaluer les résultats à l'aide de la métrique ROUGE-2. Le fonctionnement du logiciel est décrit dans la prochaine section. La section 3 contient les critères de correction du devoir et les modalités de remise.

2 DESCRIPTION

Le logiciel que vous allez construire va recevoir en entrées un fichier (décrit à la sous-section 2.1) contenant le nom des textes à résumer. Ce fichier contiendra aussi un indicateur pour savoir si nous demandons l'évaluation des résultats. Un résumé sera produit pour chaque texte. Finalement, au besoin, ce résumé sera comparé avec un résumé attendu à l'aide de la métrique ROUGE-2.

Votre programme va diviser le texte en phrase et ensuite utiliser un algorithme (décrit à la sous-section 2.2) pour choisir un sous-ensemble de phrases qui feront partie du résumé. Pour évaluer les résultats, vous devrez implémenter la métrique ROUGE-2 (décrit à la sous-section 2.3). Vous allez pouvoir ajuster votre algorithme à l'aide d'un coefficient (λ). Les sorties du logiciel sont décrites à la sous-section 2.4.

2.1 DESCRIPTION DES ENTRÉES

Votre programme va demander le nom d'un fichier de description en entrées. Chaque ligne de ce fichier représente un texte à résumer. Une ligne contient premièrement le nom du fichier qui contient le texte à résumer. Ensuite, si la ligne contient un autre nom, alors ce deuxième fichier contient un résumé possible pour le texte. Votre programme devra comparer ce résumé avec le résumé possible en utilisant la métrique ROUGE-2. Voici un exemple de fichier de description.

```
Texte1.txt resultat1.txt

Texte2.txt

Texte3.txt

Texte4.txt resultat4.txt
```

Dans ce cas, le programme doit construire un résumé pour les textes contenus dans les fichiers : Texte1.txt, Texte2.txt, Texte3.txt et Texte4.txt. Aussi, le programme va devoir comparer le résumé qu'il produit pour Texte1.txt avec le contenu du fichier resultat1.txt. Enfin, le programme doit comparer le résumé qu'il produit pour Texte4.txt avec le contenu du fichier resultat4.txt.

2.2 DESCRIPTION DE L'ALGORITHME

Dans cette sous-section, nous allons décrire comment votre programme doit construire le résumé d'un texte. Cette technique sera appliquée indépendamment pour chaque texte.

Votre programme doit premièrement diviser le texte en phrases. Pour cela, vous pouvez utiliser la même technique que vous avez utilisée pour le TP 1. Aussi, votre programme doit diviser chaque phrase en mots. Nous aurons donc les notations suivantes :

- Un texte (S_T) est une suite de phrases $S_T = [P_i, P_2, \dots, P_n]$. Où n représente le nombre de phrase dans le texte.
- Une phrase est une suite de mots $P_i = [m_{i,1}, m_{i,2}, \dots, m_{i,k_i}]$. Où k_i est le nombre de mots dans la phrase P_i.
- Nous allons spécialement dénoter la première phrase P_1 d'un texte par Q.
- Le programme construit un résumé S_R qui sera une suite de phrase extraite de S_T.

Votre programme doit pouvoir faire les calculs suivants.

2.2.1 Idf d'un mot

Calcule l'idf d'un mot. Dans ce cas, la collection est le texte et les documents sont les phrases. Cela nous indique donc le nombre de phrases qui utilise le mot. Un mot est rare dans le texte lorsque peu de phrases l'utilisent.

$$\mathrm{idf}(m) = \log_2 \frac{n}{|\{P_i | m \in P_i\}|}$$

Où $|\{P_i | m \in P_i\}|$ représente le nombre de phrases qui contiennent le mot m.

2.2.2 Présence d'un mot dans une phrase.

Cette petite fonction va simplement déterminer si un mot est présent dans une phrase. Elle retourne 1 lorsque le mot est présent, 0 sinon.

$$\mathrm{mSim}(m, P_d) = \begin{cases} 1 & \text{si } m \in P_d \\ 0 & \text{sinon} \end{cases}$$

2.2.3 Similarité d'une phrase par rapport à une autre phrase.

Calcul la similarité d'une phrase par rapport à une autre phrase. Cette similarité est directionnelle, elle n'est pas symétrique. Le calcul est pondéré par l'idf des mots.

$$\mathrm{hSim}(P_s, P_d) = \frac{\sum_{m \in P_s} \mathrm{mSim}(m, P_d) \times \mathrm{idf}(m)}{\sum_{m \in P_s} \mathrm{idf}(m)}$$

2.2.4 Similarité entre deux phrases.

Cette métrique utilise la similarité directionnelle pour calculer une similarité symétrique entre deux phrases.

$$\mathrm{pSim}(P_1, P_2) = \frac{1}{2}[\mathrm{hSim}(P_1, P_2) + \mathrm{hSim}(P_2, P_1)]$$

2.2.5 Similarité maximum

Soit une phrase P et une suite de phrase S, cette fonction trouve la phrase P_i parmi S qui est le plus similaire avec P et ensuite retourne la similarité entre ces deux phrases.

$$\mathrm{maxSim}(P, S) = \max_{P_i \in S} \mathrm{pSim}(P, P_i)$$

2.2.6 Algorithme

Le programme va devoir construire un résumé S_R qui contient un minimum de $t = 125$ mots. Pour cela, le programme choisi une phrase à la fois et l'ajoute dans S_R, jusqu'à ce que le nombre de mots du résumé dépasse t.

Pour choisir une phrase P_i, vous devez trouver parmi les phrases du texte S_T qui ne sont pas encore dans le résumé S_R, la phrase qui donne la plus haute valeur dans l'équation suivante.

$$\lambda \times \mathrm{pSim}(P_i, Q) - (1-\lambda) \times \mathrm{maxSim}(P_i, S_R)$$

Commencer avec un $\lambda = 0.5$. Ensuite, ajustez cette valeur pour obtenir des meilleurs résumés. Cette valeur doit rester entre 0 et 1. Cette formule essaie de trouver une phrase à ajouter au résumé, qui est semblable à la première phrase, toutes en ajoutant de l'information nouvelle.

2.3 Métrique ROUGE-2

La métrique ROUGE-2 est très simple à calculer. Soit deux résumés, le résumé produit par votre logiciel S_R et le résumé produit par un humain S_H. Alors, nous construisons l'ensemble de tous les bigrammes pour chaque résumé (B_R, B_H). Il ne reste qu'a calculer la précision et le rappel entre ces deux ensembles.

La précision est le nombre de bigrammes qui sont dans B_R et B_H divisé par le nombre de bigrammes dans B_R. Le rappel est le nombre de bigrammes qui sont dans B_R et B_H divisé par le nombre de bigrammes dans B_H.

$$p = \frac{|B_R \cap B_H|}{|B_R|}$$

$$r = \frac{|B_R \cap B_H|}{|B_H|}$$

Finalement, vous calculez la métrique f_1 à l'aide de ces valeurs.

$$f_1 = \frac{2pr}{p + r}$$

2.4 SORTIES

Pour chaque texte à résumer, votre programme va écrire le résumé dans un fichier qui porte le même nom avec les caractères '_r' ajoutés à la fin du nom du fichier. Dans ce fichier, vous écrivez une phrase du résumé par ligne. Si le programme devait calculer la métrique f_1 pour ce résumé, alors vous écrivez le résultat à l'écran : Le résumé pour le texte *nomDuFichier* à donné une f1 = …

3 REMISE ET ÉVALUATION

Vous devez remettre :

- votre code Java archivé

Le tout doit être remis avant le 19 août 23 :55 sur Moodle.

Vous serez évalué sur les éléments suivants.

- Votre code fait le travail demandé.