A2DI - LSI

Antonin Carette - Quentin Baert

5 janvier 2016

L'ensemble du code se trouve dans le fichier `src/main.py`.

Les documents suivants sont considérés :

- **d1** : Romeo and Juliet
- **d2** : Juliet: O happy dagger
- **d3** : Romeo die by dagger
- **d4** : Live free or die, that's the New-Hampsphire motto
- **d5** : Did you know, New-Hampsphire is in New England

Ils seront étudiés selon la requête **q** : die dagger.

L'exécution `python src/main.py` depuis la racine du projet permet d'afficher une trace qui présente les différents classements des document et en affiche une visualisation 2D.

Question 1

Pour la voir le code correspondant à la construction de la matrice terme-document, se référer à la fonction `create_doc_matrix()`.

Question 2

Pour voir le code correspondant à la transformation de la requête sous forme vectorielle, se référer à la fonction `create_doc_matrix()`.

Question 3

La fonction `get_cos_distance()` permet de calculer la distance cosinus entre un document et la requête. Selon cette fonction, voici les distances respectives des documents à la requête.

- **d1** : 0.0
- **d2** : 0.353553390593
- **d3** : 0.707106781187
- **d4** : 0.25
- **d5** : 0.0

On en déduit l'ordre suivant : **d3**, **d2**, **d4**, **d5**, **d1**

On constate que les documents sont classés en fonction du nombre de mots de la requête qu'il contiennent. C'est pourquoi **d3** est premier (il contient les mots *die* et *dagger*) alors que **d1** est dernier (il ne contient aucun des deux mots). Cependant, on pourrait supposer que **d1** mérite de se trouver plus haut dans le classement car il contient le mot *Romeo* dont on sait qu'il est mort par dague d'après **d3**.

Question 4

La fonction `get_reduced_elements()` utilise la SVD afin de donner une réduction des documents et de la requête en 2 dimensions.

Question 5

Après réduction en 2 dimensions à l'aide de la SVD, on observe les distances suivantes :

- **d1** : 0.976675542878
- **d2** : 0.971726773566
- **d3** : 0.999999745587
- **d4** : 0.948244200834
- **d5** : -0.170018973726

On en déduit l'ordre suivant : **d3**, **d1**, **d2**, **d4**, **d5**

On constate ici que la SVD "ajoute du sens" aux données ou prend plus en compte le contexte. En effet, en prenant en compte notre remarque précédente (voir QUESTION 3), **d1** est maintenant classé deuxième alors qu'il ne contient aucun mot de la requête. À l'inverse **d4** descend dans le classement et semble moins pertinant alors qu'il contient le mot *die*.

Question 6

Voici la visualisation des documents et de la requête en 2 dimensions.

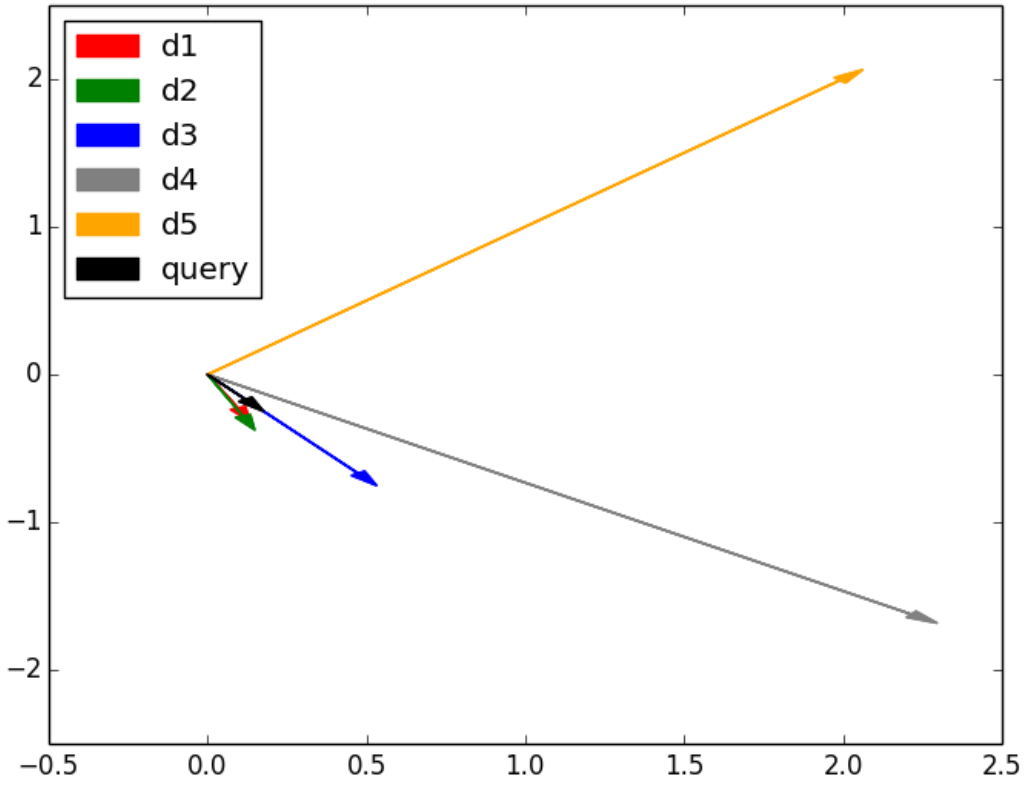

Figure 1: Visualisation des documents et de la requête en deux dimensions