Définition des algorithmes

Deuxième Labo - Gnuside

Lundi 8 août 2011

1 Objet

On se propose de réaliser trois prototypes d'algorithmes pour le projet vocabulari.se, outil d'aide à la catégorisation de documents et candidat au concours Mendeley

Chacun des algorithmes prendra en paramètre un mot entré par l'utilisateur (désigné par T_0 par la suite) et retournera un ensemble final de mots (désigné par $FinalTags = \{FT_0, ..., FT_n\}$) choisis selon les propositions suivantes.

2 Définitions

2.1 Documents liés à un tag

On définit la fonction $related\_documents(T_0)$ comme suit :

$related\_documents(T_0)$ = les documents $\{D_0, ..., D_n\}$ taggués par T_0 dans Mendeley

2.2 Documents liés à une association de tags

On définit la fonction $related\_documents(Ta, Tb)$ comme suit :

$related\_documents(Ta, Tb) = related\_documents(Ta) \cap related\_documents(Tb)$

2.3 Tags liés à un tag

On définit la fonction $related\_tags(T_0)$ comme suit :

$RelatedTags \leftarrow \emptyset$
Si T_0 est un tag dans Mendeley, alors :
    $Documents \leftarrow related\_documents(T_0)$
    Pour chaque document $D_i \in Documents$, faire :
        $DocumentTags \leftarrow$les tags $\{DT_{0,}...DT_n\}$ liés au document D_i
        $RelatedTags \leftarrow RelatedTags \cup \{DT_{0,}...DT_n\} \setminus T_0$
    FinPour
Sinon
    Si T_0 possède une entrée dans Wikipedia, alors :
        $RelatedTags \leftarrow$les premiers liens internes qui apparaissent dans l'article
    FinSi
FinSi
Retourner $RelatedTags$

Note :
– On permettra l'utilisation d'un filtre sur les $RelatedTags$, pour réduire leur nombre si nécessaire.

3 Algorithmes

3.1 Association audacieuse

Il s'agit de proposer les associations de tags qui apparaissent le moins souvent ensembles parmis les tags les plus populaires.
On définit la fonction $association\_audacieuse(T_0)$ comme suit :

$Documents \leftarrow \emptyset$
$RelatedTags \leftarrow related\_tags(T_0)$
`Pour chacun des tags` $RT_i \in RelatedTags$ `on calcule :`
    $Documents \leftarrow related\_documents(T_0, RT_i)$
    $vues(RT_i) \leftarrow$`la somme des vues des élements dans` $Documents$
    $apparitions(RT_i) \leftarrow$ `le nombre d'éléments dans` $Documents$ `(associant` T_0 `et` RT_i`)`
    $pente(RT_i) \leftarrow \frac{apparitions(RT_i)}{vues(RT_i)}$
`FinPour`
$FinalTags \leftarrow$`les` RT_i `triés par` $pente(RT_i)$ `croissante,` $\forall RT_i \in RelatedTags$
`Retourner` $FinalTags$

Notes :
– On limitera le nombre de tags liés si nécessaire ;
– On combinera éventuellement l'association audacieuse avec une recherche des associations les moins fructeuses dans Wikipedia.

3.2 Association par « hotness rank »

On définit la fonction $tag\_hotness(T_0, RT_i)$ comme suit :

$hotness \leftarrow 0$
$Articles \leftarrow$`liste des articles issus d'une recherche de` T_i `AND` RT_i `sur Wikipedia`
`Pour chaque` $A_i \in Articles$`, faire :`
    $hotness\_article(A_i) \leftarrow$`nombre de sections sur la page de discussion (éventuellement artchivée)de` A_i
    $hotness \leftarrow hotness + hotness\_article(A_i)$
`FinPour`
`Retourner` $hotness$

Puis la fonction $association\_hotness(T_0)$ comme suit :

$RelatedTags \leftarrow related\_tags(T_i)$
`Pour chaque` $RT_i \in RelatedTags$`, faire :`
    `Calculer` $tag\_hotness(T_0, RT_i)$
`FinPour`
$FinalTags \leftarrow$ `les` RT_i `triés par` $tag\_hotness(T_0, RT_i)$ `croissant,` $\forall RT_i \in RelatedTags$
`Retourner` $FinalTags$

Notes :
– On limitera le nombre de tags liés si nécessaire ;
– On limitera le nombre d'articles d'une recherche si nécessaire.

3.3 Association extra-disciplinaire

Il s'agit de proposer les associations de tags qui mettent en valeur les tags les plus multi-disciplinaires, hors de la discipline la plus évidente.

On définit la fonction $association\_extradisciplinaire(T_0)$ comme suit :

$RelatedTags \leftarrow related\_tags(T_0)$
`Pour chacun des tags` $RT_i \in RelatedTags$ `on fait :`
$Documents \leftarrow related\_documents(T_0, RT_i)$
$Disciplines(RT_i) \leftarrow$`les disciplines` $\forall Doc_j \in Documents$
$ReadersAvg(RT_i, Disc_j) \leftarrow$`la moyenne des quantités (%) de readers de` $Disc_j$`,`
$\forall Disc_j \in Disciplines(RT_i)$
$FinalDisciplines(RT_i) \leftarrow Disciplines(RT_i) \setminus$
$\{$`la discipline majoritaire pour` $ReadersAvg(RT_i, ...)\}$
`FinPour`
$FinalTags \leftarrow$`les` RT_i `triés par cardinal de` $FinalDisciplines(RT_i)$
`puis par` somme de $ReadersAvg(RT_i, Disc_j), \forall Disc_j \in FinalDisciplines(RT_i)$
`Retourner` $FinalTags$

Notes :

– On pourra également prendre en compte le nombre absolu de lecteurs par discipline pour supprimer ladite discipline si ce nombre est trop faible.