

# RECHERCHE D'INFORMATION

## M2 DAC

### TME 3. Evaluation

Au cours de ce TME, le travail à réaliser est d'implémenter un module d'évaluation des modèles de RI selon des requêtes prédéfinies et des jugements de pertinence pour ces requêtes et les collections documents considérés. L'ensemble des classes à définir au cours de ce TME seront placées dans un package **evaluation**.

1 CHARGEMENT DES REQUÊTES
---------------------------

Afin d'être à même de comparer les différents modèles de recherche d'information implémentés, il s'agit de développer un module de chargement d'un ensemble de requêtes prédéfinies, associées à des jugements de pertinence pour la collection de documents considérée. Pour les collections de documents proposées pour ces TMEs, ces requêtes et ces jugements sont contenus dans les fichiers d'extensions respectives .qry et .rel des répertoires des deux corpus. Le format de ces fichiers est décrit dans le TME1.

De la même manière que pour le chargement des documents, nous allons définir une classe *QueryParser* qui, à partir d'un fichier de requêtes et d'un fichier de jugements, propose la méthode abstraite *nextQuery()*, à redéfinir dans une classe spécifique correspondant au format de fichier donné en entrée et retournant un objet Query correspondant à une nouvelle requête lue dans les fichiers d'entrée. Cet objet contient à minima:

- *id*: L'identifiant de la requête;
- *text*: Le texte de la requête;
- *relevants*: Liste des documents pertinents pour la requête. Pour chaque document pertinent, on associe une liste de couples (sous-thème,score), correspondant respectivement aux quatrième et troisième colonnes du fichier de jugements (pour le moment ces colonnes ne contiennent que des 0 mais stockez les quand même, votre Parser sera utilisé en fin de semestre sur d'autres collections où ces valeurs auront de l'importance).

## 2 MESURES D'ÉVALUATION

Pour la définition de toutes les mesures d'évaluation de la plateforme, nous nous appuyerons sur un objet *IRList* contenant la requête (un objet *Query*) et une liste ordonnée de documents associés à leur score de pertinence.

Toute mesure d'évaluation doit dériver d'une classe abstraite générique *EvalMeasure* qui contient à minima une méthode abstraite *eval(l)* retournant le résultat de l'évaluation de la liste ordonnée de l'objet *IRList* *l* passé en paramètre. Au moins deux classes d'évaluations doivent être définies:

- Une mesure de precision-rappel  $P(l)_{R \geq k}$ , qui retourne la liste des precisions interpolées pour différents niveaux de rappel  $k$  de la liste  $l$ :  $\max_{i \in 1..nbDocs, R(l)@i \geq k} P(l)@i$ , avec  $P(l)@i$  et  $R(l)@i$  respectivement des mesures de precision et de rappel au rang  $i$  de la liste contenue dans l'objet  $l$  considéré. Cette mesure pourra prendre un paramètre *nbLevels* qui indique le nombre de points de precision-rappel à calculer;
- Une mesure de precision moyenne  $AP(l)$ , qui retourne la moyenne des precisions calculées après chaque document pertinent de la liste  $l$ :  $AP(l) = \frac{1}{|Pert(l)|} \sum_{i \in 1..nbDocs, D_{l,i} \in Pert(l)} P(l)@i$ , avec  $P(l)@i$  le score de precision au rang  $i$  de la liste  $l$ ,  $Pert(l)$  l'ensemble des documents pertinents pour la requête considérée dans  $l$  et  $D_{l,i}$  le document de rang  $i$  dans la liste  $l$ .

## 3 PLATEFORME D'ÉVALUATION

Définir une classe *EvalIRModel* permettant l'évaluation de différents modèles de recherche sur un ensemble de requêtes et selon différentes mesures d'évaluation. Les résultats devront être résumés pour l'ensemble de requêtes considérées en en présentant la moyenne et l'écart-type pour chaque modèle.