

Requêtes personnalisées sous un modèle de profil utilisateur généralisé basé sur les préférences SPARQL floues

Olfa Slama

7 Rue Paul Vaillant Couturier, 92300 Levallois-Perret
oslama@umanis.com

1 Introduction

Cet article s'appuie sur les travaux de Pivert et al. (2016) où nous avons proposé FURQL, une extension floue du langage SPARQL, et représente la première solution pour une intégration en douceur de la personnalisation de la requête tel qu'il a été proposé par Koutrika et Ioannidis (2010). Nous proposons un *framework d'interrogation de requête personnalisé* qui permet d'améliorer la requête utilisateur FURQL avec ses préférences les plus pertinentes stockées séparément dans son profil afin de fournir des réponses personnalisées et taillées sur mesure.

2 Personnalisation des requêtes FURQL à préférences

Le modèle de profil contient les caractéristiques de l'utilisateur (comme le *nom*, etc) de chaque utilisateur accédant au système. Chaque utilisateur peut exprimer différentes formes de préférences floues avec un score, compris dans $[-1, 1]$, qui indique son degré d'intérêt. Un score *positif* sert à exprimer un intérêt plus élevé ("J'aime l'acteur B. Cooper") avec un degré 1, indiquant un intérêt extrême. Il peut également être *négatif* pour exprimer une aversion ("Je n'aime pas les films d'actions") avec un degré -1 pour les valeurs "les plus déplaisantes" ou complètement indifférent avec un degré 0. Chaque préférence est associée à un score pouvant indiquer : 1) Le *poids de chaque attribut du noeud*, par exemple, un utilisateur peut être intéressé par la *date* d'un album plus que par le *genre*. Nous capturons ces variations de priorité en associant chaque attribut de noeud à un degré d'intérêt. 2) Les *relations non-floues/floues entre les entités*, par exemple, "je suis très intéressé par les films récents". Ici, les préférences indiquent dans quelle mesure les entités associées dépendent les unes des autres.

Basé sur ce modèle, nous avons proposé un Framework de personnalisation des requêtes qui procède en deux phases. 1) *Sélection des préférences* : nous extrayons du profil les meilleures préférences pertinentes pour la requête FURQL Q_u . Le principe ici est de trouver le meilleur enrichissement de la requête avec le profil utilisateur en identifiant les variables candidates impliquées dans Q_u . Pour chaque variable v_i , nous récupérons les informations nécessaires (les préférences de profil utilisateur ainsi que leurs scores respectifs) qui correspondent à cette variable et doivent être prises en compte pour le calcul de μ_{p,v_i} (qui est le produit de tous les scores candidats concernant les préférences du profil). 2) *Évaluation personnalisée des requêtes* : pour chaque résultat non-flou renvoyé, nous calculons le *score final*,

défini par $ScoreFinal = \sum_{i=1}^n (\mu_{P_{v_i}/v_i \in Q_u}) / n$, qui est la moyenne de tous les degrés de satisfaction des préférences utilisateur candidates à la requête utilisateur Q_u . Où $V = \{v_1, \dots, v_n\}$ est un ensemble de variables candidates de la requête, n son numéro et $\mu_{P_{v_i}}$ est le degré de satisfaction des préférences de profil utilisateur détectées concernant la variable v_i . Enfin, les résultats de requête avec un score positif sont triés par ordre décroissant suivant le $ScoreFinal$.

3 Problèmes de mise en oeuvre

Nous avons implémenté notre approche dans le prototype **SURF** (voir Figure 1) décrit dans Pivert et al. (2016). La partie principale de notre travail s’articule autour de la nouvelle phase de *Personnalisation de la requête* qui consiste en deux modules. 1) *Sélection des préférences* : on sélectionne les meilleures préférences ainsi que leurs scores correspondants aux variables de la requête utilisateur initiale Q_u . 2) *Calculateur de score personnalisé* : Pour chaque résultat non-flou renvoyé par le *moteur SPARQL*, on collecte le score de chaque préférence fournie par le profil utilisateur avec les degrés de satisfaction produits par le *module de compilation FURQL*, puis on calcule le *score personnalisé final* selon lequel les résultats sont ordonnés.

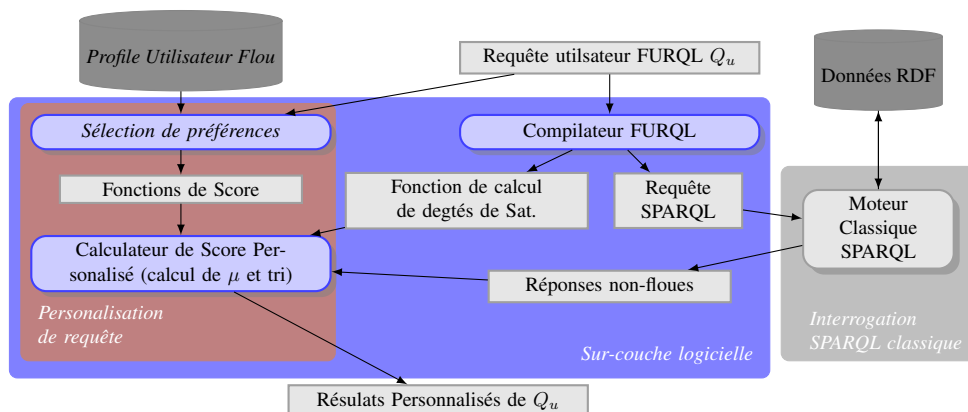


FIG. 1 – Architecture logicielle de SURF.

Afin de mesurer les performances de notre nouvelle approche, nous avons effectué une série d’expériences et les résultats sont encourageants et démontrent l’utilité d’inclure un profil d’utilisateur dans une requête FURQL. Notre approche proposée surpasse les travaux de Pivert et al. (2016), dans le sens où elle renvoie des résultats *plus petits* et *plus centrés* sur l’utilisateur.

Références

Koutrika, G. et Y. Ioannidis (2010). Personalizing queries based on networks of composite preferences. *ACM Transactions on Database Systems (TODS)* 35(2), 13.
 Pivert, O., O. Slama, et V. Thion (2016). An extension of SPARQL with fuzzy navigational capabilities for querying fuzzy RDF data. In *Proc. of the Intl. Conf. on Fuzzy Systems*, pp. 2409–2416. IEEE.