TECHNIQUES D'APPRENTISSAGE

IFT 603

FeedUS

Classificateur de flux RSS

Travail présenté à

M. Shengrui Wang

Par

Marc-Alexandre Côté-Harnois	07 166 997
Julien Filion	07 177 770
Simon Renaud-Deputter	07 149 640

Université de Sherbrooke

Hiver 2010

Table des matières

Tables de figures :

Introduction

Le projet présenté dans ce document a été réalisé dans le cadre du cours IFT603 de l'Université de Sherbrooke par l'équipe Hocus, composée de Marc-Alexandre Côté, Julien Filion et Simon Renaud, à l'hiver 2010.

Notre projet

Notre projet est une application permettant de classifier des articles récupérés à partir de flux RSS. Le but est de permettre à l'utilisateur de définir des catégories dans lesquelles seront classés les documents. Une interface simple permet la visualisation des articles directement dans le logiciel ainsi que leur classification. Cette application simplifie la vie de l'utilisateur en répartissant automatiquement les nouveaux documents dans les différentes catégories prédéfinies par celui-ci.

Introduction au flux RSS

Les flux RSS sont des fichiers dont le contenu est généré automatiquement en fonction des mises à jour d'un site web. On en retrouve sur les sites d'actualité ou les blogs pour présenter les titres des dernières informations consultables en ligne.

Le terme RSS signifie que le contenu du fichier respecte le format RSS qui s'appuie lui-même sur le langage XML. Généralement, chaque flux contient les informations suivantes :

- Titre : Le titre de l'article.
- Description : Brève description de l'article. (Qualité selon l'émetteur du flux)
- Source : Lien Internet vers l'article intégrale.
- Date : Date d'issue de l'article.

Difficultés

La classification de flux RSS présente quelques difficultés. Premièrement, le flux en tant que tel ne contient pas beaucoup d'information. En effet, la majorité des émetteurs de flux RSS n'indique aucune description ou bien simplement la première phrase de l'article. Pour cette raison, il est très difficile de seulement s'y fier.

Afin de contourner le manque d'information dans le flux, on peut suivre le lien menant à la page web où se trouve le document en entier. Cela amène une deuxième difficulté : récupérer uniquement le contenu de l'article.

Troisièmement, l'utilisateur s'attend à voir des résultats rapidement, c'est-à-dire sans avoir à classifier des centaines d'articles avant d'obtenir un classement potable. Pour cette raison, il faut que les algorithmes de classification et de segmentation performent même en présence de données peu nombreuses.

En raison de ces difficultés, un excellent traitement des données avant de les utiliser dans les algorithmes est nécessaire. Également, une part de responsabilité revient à l'utilisateur lorsqu'il classifie les documents. Les articles d'une même classe devront être nombreux et présenter de fortes similitudes.

L'application FeedUS

L'application offre plusieurs fonctionnalités permettant de gérer les flux, les catégories, les documents et les algorithmes. L'arborescence dans le cadre de gauche perme de voir les articles disponibles dans l'application. Les items avec une étoile représentent les catégories, crées soit par l'utilisateur, soit par l'algorithme de segmentation. L'icone de cadenas indique que le document en question a été assigné à une catégorie manuellement par l'utilisateur et qu'il est verouillé. L'enveloppe ouverte indique un document qui a été lu, tandis que l'enveloppe fermée, un nouvel article.

Voici l'interface principale de l'application :

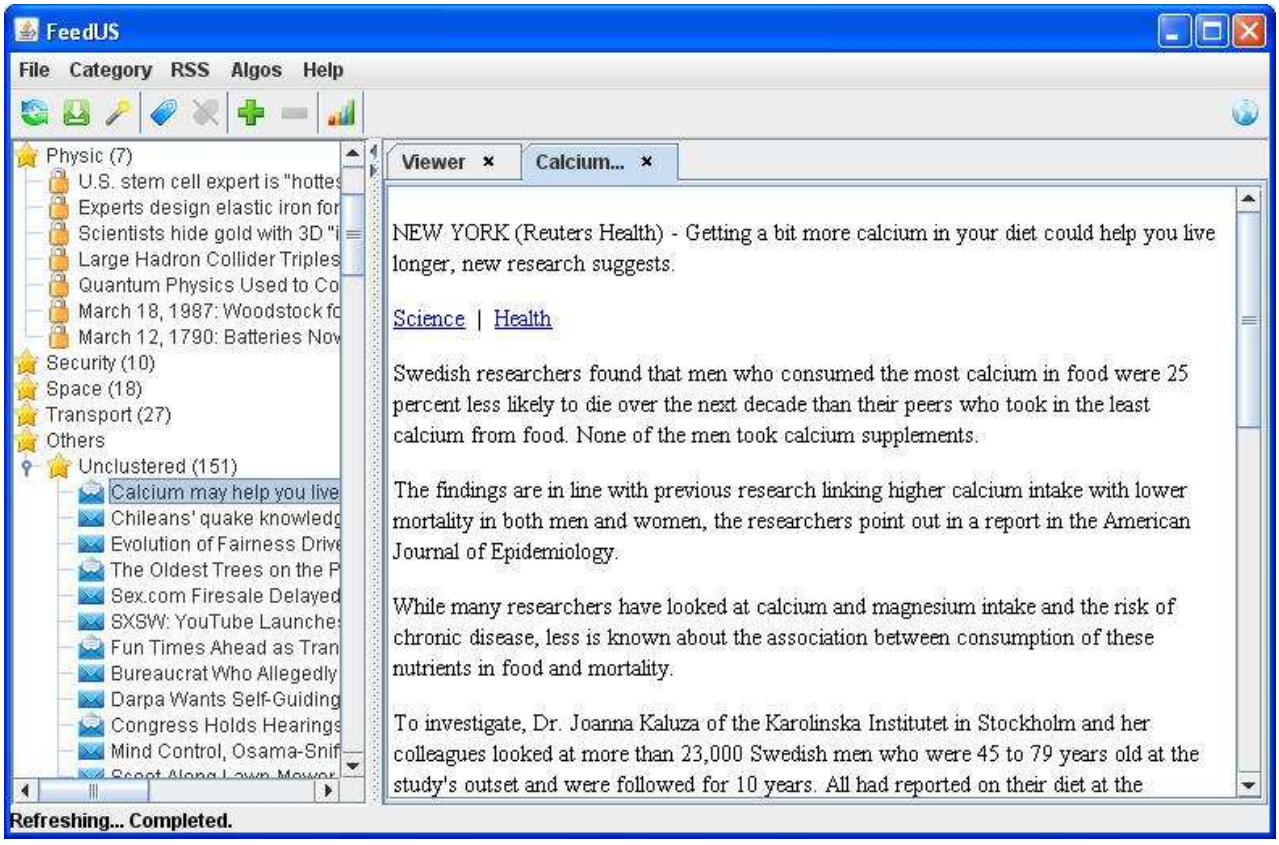

Figure 1 – Interface principale de l'application

Comparaison

Un outil très utile dans l'application est la possibilité de comparer visuellement la justesse des algorithmes de classifications.

Voici le tableau de comparaison :

Figure 2 – Tableau de comparaison des algorithmes de classification

Fonctionnement

Ajouter un Flux RSS

Au premier démarrage de l'application, l'utilisateur doit inscrire des flux RSS à partir desquels les articles désirés seront récupérés. Pour ajouter un flux, il suffit d'avoir son adresse source.

Malheureusement, certains sites ne suivent pas de standards en ce qui concerne l'emplacement du contenu des articles. C'est pourquoi, l'utilisateur peut spécifier certains filtres (sur le nom des balises ainsi que sur les attributs ID, class et name) qui seront appliqués à tous les textes récupérés à partir de ce flux RSS. Il arrive parfois (souvent) que les articles d'un même flux ne soient pas cohérents entre eux. Dans de tels cas, l'application ignore les textes fautifs.

Pré-traitement des articles

Par la suite, l'utilisateur demande au programme d'aller chercher les nouveaux textes sur Internet. Pour chaque document récupéré plusieurs étapes doivent êtres effectuées afin de le rendre compatible avec les différents algorithmes de classifications. Tout d'abord, la page Internet référencée par le document RSS est récupérée. Pour n'obtenir que le contenu de l'article, les règles associées au flux RSS sont utilisées. Une fois les informations récupérées, elles doivent être épurer de toutes les balises HTML. Ensuite, les mots se trouvant dans la liste de « StopWords », c'est-à-dire la liste des mots les plus fréquents de la langue, sont retirés. Ensuite, une lexémisation (stemming) est appliquée aux mots restants. Finalement, le vecteur TF-IDF de ce document, sera généré, à l'aide de la fréquence de chaque mot et le nombre de documents dans lesquels ce même mot apparaît.

Affichage des articles

Une fois la mise à jour complétée, l'utilisateur peut faire un rafraîchissement de l'arborescence afin de voir les nouveaux articles. L'affichage dépend des algorithmes sélectionnés dans le menu « ***Algos*** ».

Par défaut, l'algorithme de classification est « ***User*** » et celui de regroupement est « ***None*** », ce qui résulte en l'affichage de tous les articles classés par catégorie que l'utilisateur a assignés. Le reste des articles seront regroupé sous une même branche « ***Unclustered*** ».

L'utilisateur peut changer l'algorithme de classification ce qui affichera le classement des articles selon la méthode utilisée à l'exception des articles verrouillés. Il peut également modifier l'algorithme de segmentation causant ainsi le regroupement des articles non-classifiées sous la branche « ***Others*** ».

Entraînement des algorithmes

Après que l'utilisateur ait classé quelques articles dans les catégories qu'il aura préalablement créées, l'application peut (ré)entraîner les algorithmes en se basant sur les décision de l'usager ainsi que sur les paramètres propre à chaque méthode qu'il aura définis.

Algorithmes

Algorithme non-supervisé

Nous avons implanté un algorithme de clustering pour aider l'usager à sélectionner les articles qui l'intéressent. Nous appliquons cet algorithme sur tous les documents non classés. Nous avons choisi d'implémenter Bisecting K-Mean comme algorithme non-supervisé. Nous utilisons la distance du cosinus comme mesure de distance.

Algorithme supervisé

Les algorithmes supervisés nous permettent de proposer à l'utilisateur des documents qu'il risque d'aimer. Nous avons choisi d'utiliser trois algorithmes de classification : K-NN, Random Forest et Naive Bayes. Nous avons utilisé l'implantation du logiciel Rapid Miner pour Random Forest et Naive Bayes alors que nous avons notre propre implémentation de K-NN.

K-NN

Nous avons rencontré un problème avec le K-NN de base, car cet algorithme est fait pour trouver une classe pour tous les documents. Or, dans notre application, certain document ne devrait se retrouver dans aucune classe. Pour cette raison, nous avons ajouté un seuil de similarité comme paramètre. Ce seuil intervient lorsque nous calculons les K plus proche voisins. Un document ne peut faire partie des K plus proche voisins que si la similarité entre les documents est plus grande que le produit du seuil et de la similarité moyenne de la classe du document. Pour calculer la similarité moyenne, nous calculons la similarité entre chaque document et son K^{ieme} plus proche voisin de la même classe. On fait ensuite la moyenne de ces similarités pour chaque classe. Nous ne classons pas un document lorsque nous ne pouvons pas lui associer K voisins. Il est donc important que le seuil se situe entre 0 et 1. Une valeur de 0 est le cas général de K-NN. De plus, nous utilisons la similarité du cosinus comme mesure de similitude.

Évaluation

Nous avons estimé l'erreur de généralisation de notre classification à l'aide de la technique Leave-One-Out (variante de Cross Validation). L'ensemble d'entrainement est composé de 228 documents divisés en 12 catégories où chaque catégorie possède entre 7 et 35 documents.

K-NN

Erreur de généralisation = 27%
Erreur de classification = 15 %
Paramètres : Nombre de voisins requis = 3

Naive Bayes

Erreur de généralisation = 38%
Erreur de classification = 0 %
Paramètres : N/A

Random Forest

Erreur de généralisation = 63%
Erreur de classification = 0 %
Paramètres : Nombre d'arbres utilisés = 10
Profondeur limite de l'arbre = infini
Nombre d'attributs utilisés = log(M+1) , M étant le nombre total d'attributs.