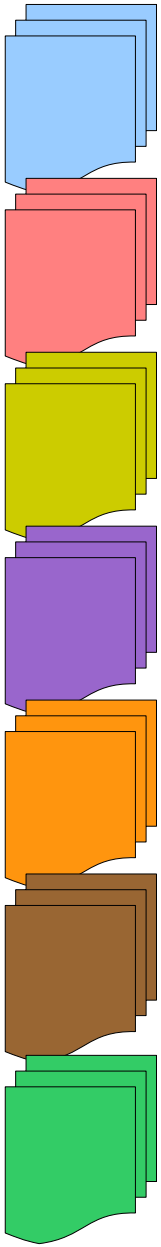
Challenge « Aide à la décision »

Principe du challenge



accueil

blog

commerce

FAQ

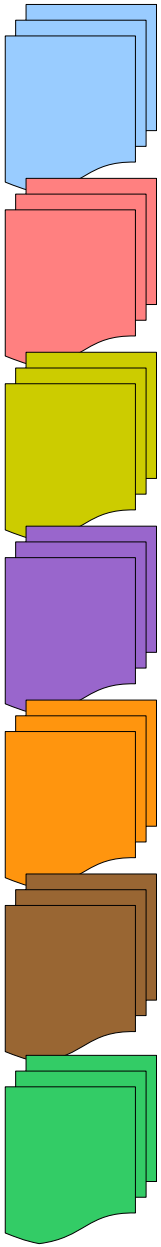
home

liste

recherche

Classer des pages web selon 7 classes

Principe du challenge



accueil

blog

commerce

FAQ

home

liste

recherche

Nguyen Quang Huy

PhD in Computer Science

PROTHEO group - LORIA & INRIA Lorraine (until August 2003)

I have moved to **Formal Methods & Security Group**
(Schlumberger Advanced Research on Smartcards)

Contact Address

LORIA
BP 219, 54600 Villers-lès-Nancy
FRANCE

PhoneOffice:
Mobile:
PhoneOffice:

Quang-Huy*dot*Nguyen*at*loria*dot*fr

Research Interests

- Formal methods
- Term rewriting & Rewriting calculus
- Theorem proving & Type theory

Software

- An ELAN-based tactic for AC rewriting in Coq is available here
- A Coq/ELAN interface for equational reasoning in Coq is available here

Publications

PhD dissertation

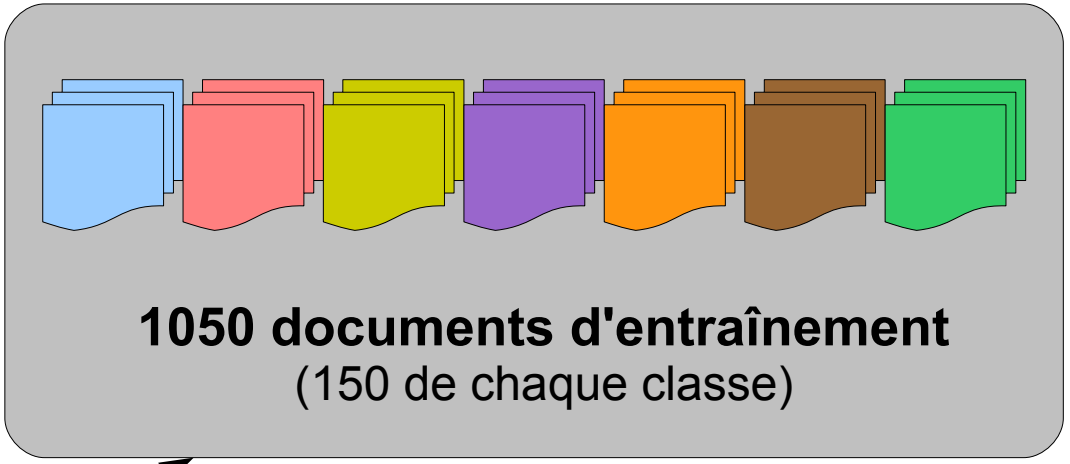
CV (and here is a printer-friendly version)

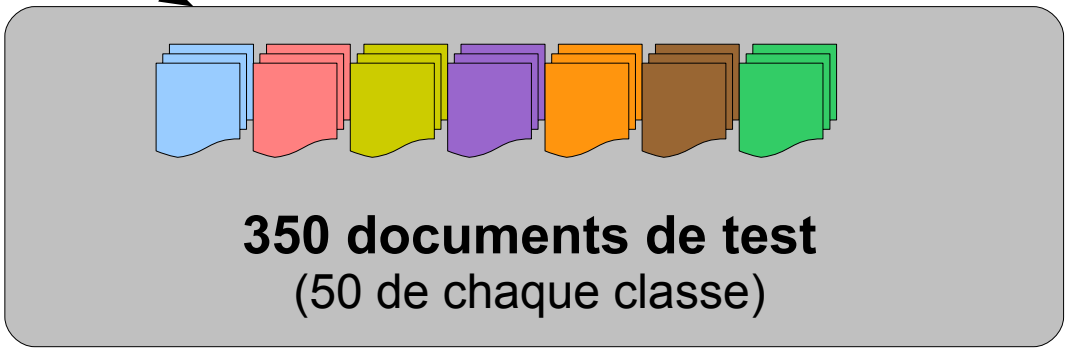
Wanna see my son ?

Last Modified on May 2004 Readlog

Exemple : « recherche »

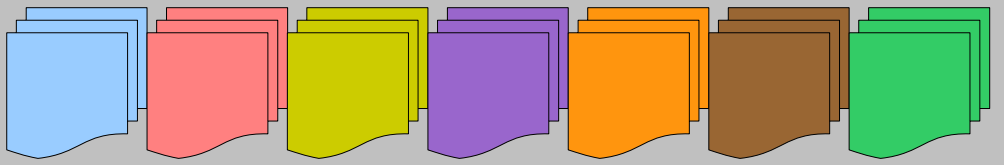
Principe du challenge





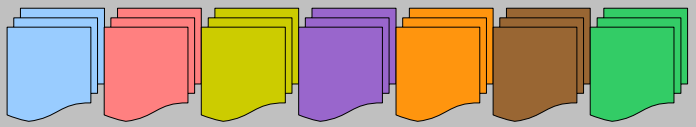
Principe du challenge

Données d'entraînement
Disponible sur Célène



1050 documents d'entraînement
(150 de chaque classe)

Données d'évaluation
Réservé aux enseignants



350 documents de test
(50 de chaque classe)

Principe du challenge

Etape 1

A partir des données d'entraînement : vous construisez un classifieur capable d'attribuer une (et une seule) classe à un document donné

classer<-function(fic){...}

- *fic* : est une chaîne de caractères contenant le nom d'un fichier à classer (e.g. "../toto.html")

- La fonction doit renvoyer une étiquette parmi : "accueil", "blog", "commerce", "FAQ", "home", "liste", "recherche" (attention à la casse)

Principe du challenge

Etape 1

A partir des données d'entraînement : vous construisez un classifieur capable d'attribuer une (et une seule) classe à un document donné

classer<-function(fic){...}

- *fic* : est une chaîne de caractères contenant le nom d'un fichier à classer (e.g. "../toto.html")
- La fonction doit renvoyer une étiquette parmi : "accueil", "blog", "commerce", "FAQ", "home", "liste", "recherche" (attention à la casse)

La fonction *classer* doit prendre au plus 5 secondes*
pour renvoyer la classe d'un document

(* sur les machines de l'IUT)

Principe du challenge

Etape 2

Le chef de groupe dépose sur Célène votre classifieur sous forme d'une Archive **Nom1Nom2.zip** dont l'extraction donne lieu à un répertoire Nom1Nom2 et contenant au minimum 2 fichiers (à la racine):

- readme.txt
  - quelques lignes expliquant l'approche utilisée : type de classifieur, nature et nombre de descripteurs utilisés
  - erreur calculée sur les données d'entraînement
- main.R (contenant au minimum la fonction *classer(fic)*)

N.B. Tout fichier ou ressource nécessaire à l'exécution du script R devra être fourni dans l'archive y compris les sources (.tar.gz) des librairies complémentaires utilisées (autres que tm, NLP, SnowballC, rpart et e1071). Mentionner les installations requises dans le fichier readme.

Principe du challenge

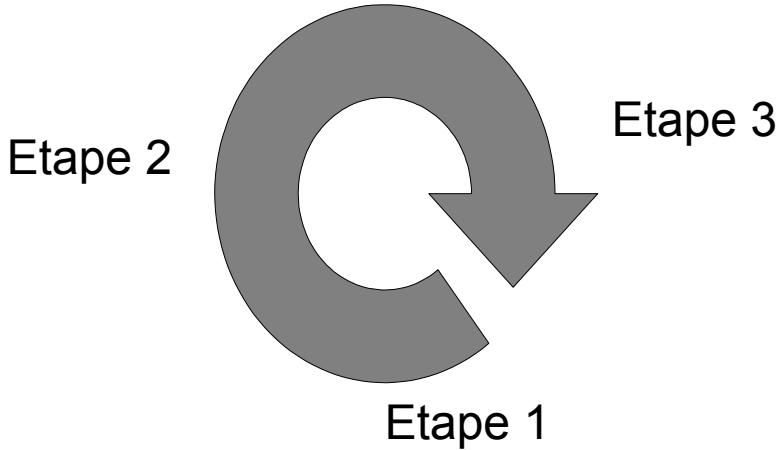
Etape 3

Les enseignants évaluent (automatiquement) votre classifieur sur les données de test

5 tentatives par binôme (la meilleure est retenue)

Dépôt ne respecte pas les spécifications = 0
(compte pour une tentative)

Toutes les tentatives doivent être déposées par le chef de groupe



Jusqu'au XXX* à minuit
(*) date à définir

Principe du challenge

Etape 4

Déposer sous Célène (autre dépôt) un document (Nom1Nom2.pdf) expliquant :

- La (les) description(s) que vous avez utilisées,
- comment avez-vous construit ces représentations (outils trouvés sur le web, outils développés par vous-même, …)
- quelle(s) méthode(s) d'apprentissage vous avez utilisé : vous pouvez indiquer les méthodes que vous avez essayées sans succès, celles qui donnent des résultats pertinents, …)
- si vous avez utilisé plusieurs méthodes, vous devez indiquer comment vous avez combiné les résultats fournis par ces méthodes,
- la ou les librairies R si vous en avez utilisées,
- vous indiquerez également les taux d'erreur (ou de bonne classification) de votre méthode sur les données d'entraînement

Compléments

Une note est disponible sur Célène afin de vous aider à traiter les corpus html bruts et en extraire uniquement le contenu textuel ou structurel (balises).

En suivant l'exemple de traitement du corpus proposé dans cette note et en appliquant le classifieur 1-plus-proche-voisin avec les spécifications (basiques) suivantes :

- Descripteurs = mots apparaissant au moins 200 fois dans le corpus
- Pondération = fréquence du mot dans le document
- Distance = distance euclidienne

… vous obtiendrez une solution avec 75% de bonne classification sur les données de test !!!

Calcul de la note

Votre note sera principalement calculée à partir de la performance de votre classifieur de sorte qu'elle sera :

- <8 : si votre classifieur obtient un score inférieur à 75%
  (croissance linéaire dans [0;75%])
- >8 : si votre classifieur obtient un score supérieur à 75%
  (croissance quartique dans [75%;100%]