Licence d'informatique
Module de C/Unix

Projet de C

ham, spam, graham...

Philippe MARQUET

Décembre 2003

L'objet de ce projet est de mettre en œuvre le principe utilisé par les filtres de pourriels (courriels non sollicités, ou spam) proposé par Paul GRAHAM et décrit sur son site `http://www.paulgraham.com/spam.html`.

Dans un ensemble de courriels (courriers électroniques, emails), on distingue les *spams*, « mauvais » courriers, des *hams*, « bons » courriers. Une première phase d'apprentissage est basée sur un corpus existants de courriels identifiés comme spams et un corpus de courriels existants identifiées comme hams. La seconde phase permet d'utiliser les informations issues de cet apprentissage pour désigner un nouveau courriel donné comme spam ou ham.

1 Algorithme de Paul GRAHAM

Les phases suivantes sont identifiées pour la mise en place d'un tel filtre. Quelques justifications sont données à l'URL citée ci-dessus.

Création de tables de hachage À partir du corpus des spams et du corpus des hams, nous créons deux tables de hachage contenant pour chacune l'ensemble les mots du corpus et leur nombre d'apparition dans le corpus.

Cette création demande de définir précisément le terme *mot*. On peux par exemple choisir qu'un mot est une suite de caractères significatifs, les caractères significatifs étant les lettres et les signes `-` et `_`. Les autres caractères sont considérés comme délimiteurs et ignorés. Les lettres minuscules et majuscules sont indifférenciées.

Calcul de la probabilité associée à un mot Paul GRAHAM propose le calcul suivant, pour un mot `word` donné, de la probabilité qu'un courriel contenant ce mot soit un spam :

- soit $good$ le double du nombre d'apparitions de `word` dans le corpus ham ;
- soit bad le nombre d'apparitions de `word` dans le corpus spam ;
- si $good + bad$ est inférieur à 5, retourner une probabilité p de 0.4 ;
- sinon, si `word` appartient au corpus spam et pas au corpus ham retourner une probabilité de 0.99 ;
- sinon si `word` appartient au corpus ham et pas au corpus spam retourner une probabilité de 0.01 ;
- sinon retourner une probabilité

$$\frac{\min(1, \frac{bad}{nbad})}{\min(1, \frac{good}{ngood}) + \min(1, \frac{bad}{nbad})}$$

$nbad$ et $ngood$ sont respectivement les tailles (nombre de courriels) des corpus spam et ham.

Élection des mots significatifs À partir de l'ensemble des probabilités de chacun des mots d'un courriel, on ne retient que les 15 mots les plus significatifs. Un mot est d'autant plus significatif que sa probabilité est éloignée de la valeur neutre 0.5.

Calcul de la probabilité combinée Les probabilités p_i des 15 mots les plus significatifs sont combinées selon la formule suivante pour donner la probabilité que le courriel soit un spam :

$$\frac{\prod_i p_i}{\prod_i p_i + \prod_i (1 - p_i)}$$

Qualification du courriel Si la probabilité combinée est supérieure à 0.9, le courriel est considéré comme un spam.

2 Filtrage

Dans un premier temps votre programme :
- prend en entrée deux corpus et un courriel à qualifier ;
- fournit une qualification, ham ou spam, du courriel.

Vous utilisez pour manipuler les tables de hachage, les primitives `hcreate()`, `hsearch()` et `hdestroy()` de la bibliothèque Unix `hsearch`. Consultez le manuel en ligne ou `http://www.opengroup.org/onlinepubs/007908799/xsh/hsearch.html`.

3 Hachage

Dans une seconde étape, vous implantez votre propre bibliothèque de manipulation de tables de hachage.

4 Sauvegarde

Il s'agit ensuite de ne pas reconstruire les tables de hachage à chaque utilisation mais de mémoriser de manière permanente l'ensemble de l'information exploitée dans un fichier représentant une version indexée du corpus.

Vous développez alors
- une commande qui à partir d'un corpus donné sous forme textuelle fournit un corpus sous une forme indexée ;
- une variante de votre commande initiale qui prend en entrée des corpus indexés.

5 Validation

Un corpus de spams vous sera fourni. Vous pourrez utiliser vos propres courriels comme corpus de ham.

Fin du sujet