Automatic documents classification
SortDC

Godefroy de Compreignac   Ronan Letellier

July 5, 2011

Contents

Introduction

Nowadays, the always increasing amount of information available on the Internet makes it impossible to let it be managed exclusively by humans. One important remaining challenge in computer sciences is making it possible to use computers to accomplish tasks that would normally require free will. Indeed, some tasks can require a significant amount of time without producing any added value. Automating some of these process would allow one to concentrate on more complicated issues, thus gaining in productivity.

Our work during this semester (February 2011 - June 2011) consisted in finding a mathematically correct algorithm in order to automatically sort « human-readable » documents in different categories, and programming an application using this algorithm to actually accomplish the task.

1 Needs and goals

1.1 What did we want to achieve

Our intention was to conceive an Open Source product that would be easy to configure and use, as we could not find an existing one. To reach these requirements, we had to be able to provide a RESTful[1] API[2] that would allow anyone to try and use our software features. To make it really flexible, we also wanted it to work on different DBMS[3].

1.2 Why do we think this is useful

While the internet content keeps growing, we need powerful devices to keep some order in all human oriented items. Automatically understanding what a text document is about allows search engines to determine wether a website suits one's key words or not, our mailboxes to send junk mail directly in the spam folder, and so on. However, such automated softwares are not available to anyone, and we still have to order our own web contents by hand.

En fonctionnant côté serveur et grâce à une API RESTful, la solution que nous proposons permet par exemple de catégoriser des articles de blogs, de détecter la langues de textes écrits par des internautes sur un site, ou encore de détecter les spams envoyés dans les commentaires d'un blog ou dans un formulaire de contact.

2 Studied technologies

Sorting documents is not a one step process. We decided to go for texts stemming and tokenization. Therefore, we tried to find existing solutions that would help us to do so.

[1] Representational State Transfer

[2] Application Programming Interface

[3] Data Base Management System

2.1 Existing solutions

2.1.1 Apache Mahout

Apache Mahout est un outil open source de la communauté Apache, conçu pour être scalable et fournir des implémentations de plusieurs algorithmes utilisés en machine learning, comme les K-means, Naive Bayes classifier, les recommandations, etc.

Mahout est une application Java et fonctionne côté serveur sur des clusters Hadoop. Son utilisation est adaptée pour des calculs à grande échelle sur plusieurs serveurs. L'installation n'est pas aisée, car il faut compiler, installer et configurer Hadoop et Mahout pour la faire fonctionner.

2.1.2 RapidMiner

2.2 Our selection

We have not been able to find any Open Source software matching all our expectations.

3 Bayes classifier

3.1 Considerations

3.2 Algorithm

4 Our application

4.1 Quick overview

4.2 Features

4.3 Some tests

Conclusion

Bibliography