Automatic web content categorization
Machine Learning

Masumi Mutsuda Zapater

February 2012

Abstract

"In the information era where large amounts of content are being generated in a daily basis, autonomous processes of sorting and categorizing become of high interest. This project consists in the design and implementation of a web content classification application including well-known document classification algorithms such as Naïve Bayes and Maximum Entropy and inspired by proven tools like Mallet or Rapidminer. Accuracy and scalability are the main goals of the project."

Contents

Chapter 1

Introduction

"FlowSight is a Massively Parallel Processing platform that scales using Commercial Off-The-Shelf Hardware and Directly Attached Storage." Near the end of the year 2011 the company was asked by one of their Telecom clients to extend the functionality of one of the features their application was offering: they wanted to know not only which sites where most visited by their subscribers, but also a generic categorization of those top sites.
One way of solving this task would be doing it manually by someone in the company by reading those top sites and applying categories to them. The problem of such a manual approach is that maybe today the client asked for the top 10 sites, but you never know when will they ask for the top 100000. Another way could be using the metadata information provided by the website itself, but not every site provides its category in the metadata. The best approach to solve this task in a real scenario ended up being the artificial intelligence. This project is the result of applying the power of machine learning algorithms to a real case scenario.

Chapter 2

Supervised learning

The main goal of the project was to be able to automatically classify webpages depending on their content into different categories. By the goal itself it is clear that Machine Learning algorithms might be the best approach to try to solve the problem.

2.0.1 Client requirements

The original set of categories that we were asked to identify were the following.

`Social Networking`, `Email`, `Sport`, `News`, `Shopping`, `Weather`, `Travel`, `Maps`, `Photos`, `Instant Messaging`, `Entertainment`, `Financial`, `Adult`, `Other`

Before going any further and just looking at the categories we were asked to classify, we can observe that some of them could be more difficult to classify. Some categories like `Adult` had a clearly characteristic language in their content whereas other categories like `Email` might contain generic language. We can also see that the category `Entertainment` might be overlapping with others like `Photos`, `Travel`, `Social Networking`, etc. Finally one of the difficulties of the task was to categorize something as `Other`. For initial testing purposes that could help taking later decisions I used rapidminer. A fast-to-set and easy-to-use machine learning tool that, among other things, allowed me to test different machine learning algorithms with the same datasets in order to compare their performance.

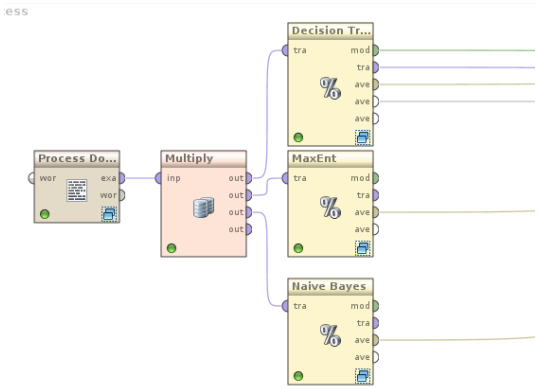
2.0.2 Crawling training data

Supervised Machine Learning algorithms need a set of training (labeled) data to infere the function that will later determine wether a new example is from one class or the other. In this case we needed web content for each of the categories. We decided to use the **Yahoo! Directory**[6] as a source for this data.

2.0.3 Tests with Rapidminer

I used rapidminer[5] to do inicial tests and try to diagnose how might some categories be overlapping and at the same time detecting which algorithms performed better.

Figure 2.1: Rapidminer process



2.0.4 MaxEnt implementation in the Framework

- Compute $d_j, j = 1, ..., k+1$
- Initialize $\lambda_j^{(1)}$ to zero
- Repeat until converge
- For each j
  - Compute $E_{p^{(n)}} f_j = \sum_{x \in \varepsilon} p^{(n)}(x) f_j(x)$ where $p^{(n)}(x) = \frac{e^{\sum_{j=1}^{k+1} \lambda_j^{(n)} f_j(x)}}{Z}$
  - Update $\lambda_j^{(n+1)} = \lambda_j^{(n)} + \frac{1}{C}(log \frac{d_i}{E_{p^{(n)}} f_j})$

Why maximum entropy? Maximize ntropy = Minimize commitment

Model all that is known and assume nothing about what is unknown. Model all that is known: satisfy a set of constraints that must hold

Assume nothing about what is unknown: choose the most "uniform" distribution choose the one with maximum entropy

Chapter 3

Language drawbacks

There are several difficulties in the classification task that can be optimized by some refination. The first difficulty is that some servers returned their content in the language of the country they detected the request was coming from. Changing the header in the requests and using a proxy solved this problem. Another difficulty in this classification task is the fact that the content of some of the URIs that are being classified is written in languages different from english. As in this specific problem the classifier is trained to work in English, this situation will potentially cause wrong classifications. We could just filter those URIs from foreign domains (`.cz, .jp, .cn`) and automatically discard them, but some of those webpages could be written in english, and we would still have the same problem with the generic `.com` domain. A better approach would be identifying the language itself and discarding the webcontents which are not in english.

3.1 Language Identification

Language identification is the process of determining which natural language given content is in. There are several computational/statistical approaches to solve this problem which is split into two different stages: modeling and classification

3.1.1 Modeling techniques

Before being able to classify languages we need to generate models that will represent each language. This can be done using different techniques:

- **Common words technique**: the most common words in a corpus are sorted by frequency in order to get a probability distribution.
- **N-Gram technique**: similar to common words technique but instead of sorting the most common words, the most common successions of N characters are sorted.

The disadvantage of the **Common words technique** is that although common words may occur in large amounts of text, they might not occur in shorter input examples. With the **N-Gram** technique we avoid that disadvantage by not using just the words but also all the N-partitions in the text.

3.1.2 Classification techniques

Once we have a model for each language we want to identify, we have to classify them.

- **Relative entropy**: compares the compressibility of the text we want to identify to the compressibility of the previously generated models.
- **Rank order statistics**: generates a probability by comparing how far an N-Gram is from the position of the same N-Gram in the model.

3.1.3 Application

For the scope of this project I chose the **3-Gram technique** to generate the language models and the **Rank order statistics** to identify the languages. I used a Python implementation by Damir Cavar[3] consisting of a model generator and a classifier. For generating the language models, I used the Wikipedia as a text source by automatically accessing random articles and crawling their content. After generating the models a simple Python algorithm rewrote the output file with the language information.

```
import lid
import os
import csv

myLid = lid.Lid()
output_file = "language_categorized.csv"
input_file = "categorized.csv"

seed_write = csv.writer(open(output_file, 'wb'), delimiter=',',
    quotechar='|', quoting=csv.QUOTE_MINIMAL)
seed_read = csv.reader(open(input_file, 'rU'), delimiter=',')

for row in seed_read:
  try:
    # This call returns the language of the file content
    language = myLid.checkText(open("unknown-data/http:"+row[0]+".out
        ").read())
    if (language != "English"):
      seed_write.writerow([row[0],"foreign_language"])
    else:
      seed_write.writerow([row[0],row[1]])
  except Exception, e:
    print "error identifying language of "+row[0]
    print e
    seed_write.writerow([row[0],"other"])
```

Before applying the LID algorithm, websites in other languages were being categorized as random categories, adding unuseful noise to the results. After applying the algorithm, those websites were categorized as `foreign_language` and were easily identified. We can see how the foreign language webpages were wrongly classified both in absolute values and percentually.

Chapter 4

Further issues

After applying all sorts of refining to the classificaction there are still some sites that are not being classified correctly. A clear example is the url "youtube.com". Whereas it is supposed to be a social network or entertainment website, the content and text in it is generated by the users themselves and just a small percentage of the content in the website talks about what it really is. This causes the website to be classified differently depending on the "social trends" of whenever the content was crawled.

Chapter 5

Conclusions

Bibliography

[1] Ronald L. Graham, Donald E. Knuth and Oren Patashnik,*Concrete mathematics* Addison-Wesley, Reading, MA, 1995.

[2] alias-i, "Logistic Regression Tutorial", *Lingpipe Home*, 23 Nov. 2011, http://alias-i.com/lingpipe/demos/tutorial/logistic-regression/read-me.html

[3] Damir Cavar, "Language Identification LID Examples", *Personal Site*, 9 Jan. 2012, http://www.cavar.me/damir/LID

[4] Arjen Poutsm, "Applying Monte Carlo Techniques to Language Identification", *Personal Site*, 9 Jan. 2012, http://ajwp.home.xs4all.nl/langident.pdf

[5] Rapid-I, "Rapidminer", *Rapidminer site*, 9 Nov. 2011, http://rapid-i.com/content/view/181/190

[6] Yahoo! Directory, "Yahoo", *Yahoo*, 10 oct. 2011, http://dir.yahoo.com/

Chapter 6

Annex: Diary

This annex contains the diary notes I took during the project development. It is written in catalan and it may contain partial or disconnected information. Reader discretion is advised.

- Rapidminer per testejar diferents solucions, naive bayes rapid, maxent lent pero + accuracy
- Categoritzacio URL per contingut -¿ MaxEnt
- Classificador basat en categories de Yahoo Directory (training)
- Testing inicial amb petites dades
- Testing posterior amb 140.000 URLs d'Irlanda O2
- Descartar idiomes estrangers -¿ Trigraphs
- Fitxers amb trigrams d'idiomes generats via randomly crawling wikipedia
- Crawling a traves de proxy amb headers tema idioma
- Afegida categoria adult
- Afegit categoria other, massa generica
- Desmembrat categoria other en altres per posterior mapeig a other
- Generat script per calcular mitja de certesa de classficacio i distancia mitja amb segona opcio
- Millorats fitxers de train en base al punt anterior

El primer approach per la classificació va ser fent servir les categories que demanava O2 irlanda. La categoria entertainment englobava molts temes, movies, tv, music, literature..., i es requeria a més una categoria "Other". El classificador sempre assigna una categoria, la que té la màxima probabilitat de ser, i per tant s'havia de generar la categoria other a partir de categories que no tinguessin res a veure. El problema de crear categoria generica other o entertainment, es que passaven a tenir molt ambigüitat, i moltes coses passaven a considerar-se other i entertainment. La solucio va ser crear subcategories sense tenir en compte entertainment i other, i fer un post tractament del ficher resultant de la classificació assignant other o entertainment a allò que realment ho era. D'aquesta manera es té més granularitat, tot i que l'accuracy del classificador baixa al tenir més categories. Un problema comú és que algunes urls no es deixen crawlejar. Detecten que no ets un navegador corrent i et donen un contingut que no és significatiu. A l'hora de classificar aquestes urls acaben sent categoritzades

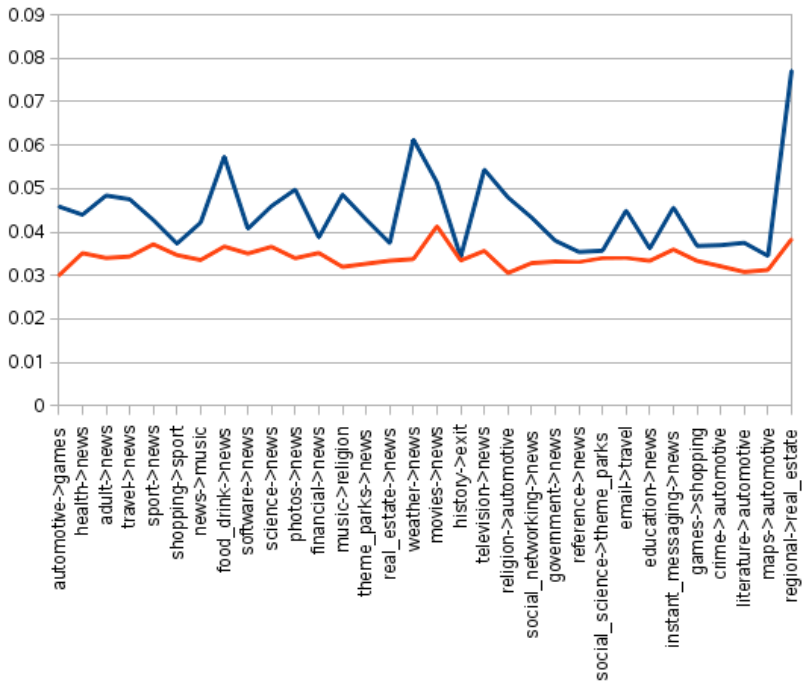
a categories que no tenen res a veure. Per exemple wikipedia retorna simplement una llista de països, que fa que la categoria passi a ser "Adult". Coses similars passen amb facebook. A youtube el problema és que el text crawlejat són els títols i descripcions de videos que la gent puja, per tant depenent de quan es produeixi el crawling, la classificacio de Youtube pot canviar.

29-12-2011

En afegir categories concretes per a ser englobades posteriorment per Entertainment, l'accuracy del classificador ha baixat fins al 73.33 adult education food_drink health literature music real_estate religion social_networking sport travel automotive email games history maps news reference science social_science television weather crime financial government instant_messaging movies photos regional shopping software theme_parks

Si la confusió és entre categories que seran englobades per "Entertainment" no hi ha problema. Observem que la distància disminueix lògicament perquè ara hi ha més categories.

Figure 6.1: Gràfic de blabla



05-01-2012

Proves a jabato