Machine Learning webcontent categorization
Artificial Intelligence

Masumi Mutsuda Zapater

February 2012

Abstract

This is the abstract

Contents

Chapter 1

Introduction

Hello my name is here [1]

```
#include "TaskFormatFieldsTest.hh"
#include "Zen.hh"

#define ZEN_TEST_NEGATE_CFG       "task-format-negate-workflow.cfg"
#define ZEN_TEST_FILTER_CFG       "task-format-filter-workflow.cfg"


TEST_F(TaskFormatFieldsTest, TimestampWorks)
{
  // Actual test starts here
  Zen zen(ZEN_BINARY, ZEN_TEST_TIMESTAMP_CFG, ZEN_TEST_ROLE,
      VERBOSITY);
  int status = zen.run();

  ASSERT_EQ(0, status);

  checkStages(STAGES_OK);
}
```

Chapter 2

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.

2.1 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

2.1.1 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.

2.1.2 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.

2.1.3 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 3

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

Chapter 4

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.

- Identificació del problema: On es determina la viabilitat de la construcció del SBC i la disponibilitat de les fonts de coneixement.
- Conceptualització: Descripció semiformal del coneixement del domini del problema i descomposició en subproblemes, segons la visió d'un expert.
- Formalització: Cal definir el mecanisme adequat de representació del coneixement, en aquest cas segons la visió de l'enginyer de coneixement.
- 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.

05-01-2012

Proves a jabato

Figure 4.1: Gràfic de blabla