Deploying a text classification service

Instituto Nacional de de Estadísticas - Chile

November 2021
Contents

1. Motivation
2. The model
3. Deployment of the model
2. Motivation
Duplicated efforts

Many teams are carrying out similar classification tasks

- Occupation
- Economic activity
- Production
- Consumption
The challenge

- To create one single platform for text classification
  - datasets
  - methodology (algorithm, quality measures, etc.)
  - programming language

```python
import random

def game():
    """An addition game that prompts user to answer simple addition problems
    using random numbers from 0-9 and prints whether they are correct or not."
    wrk = True
    ons = 0
    for i in range(10):
        val1 = random.randint(1,10)
        val2 = random.randint(1,10)
        wrk = True
        while wrk == True:
            ans = int(input(str(val1) + " + " + str(val2) + " = "))
            if(ans == val1+val2):
                print("Correct!!!")
                wrk = False
                ons = 1
            else:
                print("Incorrect....")
                wrk = False
        except:
            print("That's not a valid value :( Try again!!!")
    print("You got " + str(ons) + " out of " + str(n) + " correct!!!")
```
Building of the datasets

• We created 2 train datasets
  • Economic activity
  • Occupation

• Trained analysts labelled records

• \( \sim 50.000 \) records for economic activity
• \( \sim 30.000 \) records for occupation

• 90% of records was labelled twice
2. The model
The model

- 4 strategies were tested and the selected one was:
  - Neural net with GRU architecture
  - Spanish word embeddings for text representation

### Economic activity (1 digit classification)

<table>
<thead>
<tr>
<th>modelo</th>
<th>acc</th>
<th>macro</th>
<th>micro</th>
<th>weighted</th>
</tr>
</thead>
<tbody>
<tr>
<td>seq_1d</td>
<td>0.9384</td>
<td>0.8757</td>
<td>0.9384</td>
<td>0.9386</td>
</tr>
<tr>
<td>tfidf_1d</td>
<td>0.9327</td>
<td>0.8658</td>
<td>0.9327</td>
<td>0.9330</td>
</tr>
<tr>
<td>emb_simple_1d</td>
<td>0.9274</td>
<td>0.8641</td>
<td>0.9274</td>
<td>0.9280</td>
</tr>
<tr>
<td>emb_gru_1d</td>
<td>0.9327</td>
<td>0.8694</td>
<td>0.9327</td>
<td>0.9328</td>
</tr>
</tbody>
</table>

[https://github.com/dccuchile/spanish-word-embeddings](https://github.com/dccuchile/spanish-word-embeddings)
The model

Occupation (1 digit classification)

<table>
<thead>
<tr>
<th>modelo</th>
<th>acc</th>
<th>macro</th>
<th>micro</th>
<th>weighted</th>
</tr>
</thead>
<tbody>
<tr>
<td>seq_1d</td>
<td>0.8858</td>
<td>0.8599</td>
<td>0.8858</td>
<td>0.8855</td>
</tr>
<tr>
<td>tfidf_1d</td>
<td>0.8684</td>
<td>0.8362</td>
<td>0.8684</td>
<td>0.8686</td>
</tr>
<tr>
<td>emb_simple_1d</td>
<td>0.8793</td>
<td>0.8519</td>
<td>0.8793</td>
<td>0.8807</td>
</tr>
<tr>
<td>emb_gru_1d</td>
<td>0.8989</td>
<td>0.8796</td>
<td>0.8989</td>
<td>0.8990</td>
</tr>
</tbody>
</table>

- 4 models were created
  - Occupation 1 digit
  - Occupation 2 digits
  - Economic activity 1 digit
  - Economic activity 2 digits

https://github.com/inesscc/ineclassifiers
Sharing the models

`keras::save_model_hdf5(model, "awesome_model")`
Dependencies

- Also, sometimes it is necessary to create a **virtualenv** in a specific directory.
Dependencies

Even in the case the user is able to install the dependencies, the output may not be the same.

- R version
- Packages version
- OS
Dependencies and reproducibility

Reproducibility Spectrum

Publication only

Publication +

Code

Code and data

Linked and executable code and data

Full replication

Not reproducible

Gold standard
2. Deployment
First idea: R package

R package (library) containing the model

Better than sharing the raw files

This strategy still relies on the user setup
Upload the model to a server with all the dependencies

We can provide a service through an API

The user needs only a tool to make a request
library(httr)
library(feather)
caenes <- read_feather("src/data/split_train_test/test.feather")
request <- http::POST("http://143.198.79.143:8080/predict",
    encode = "json",
    body = list(text = caenes$glosa_caenes[1:10],
    classification = "caenes",
    digits = 1))

httr::status_code(request)
response <- http::content(request)
Adventages of this solution

- The user does not require any dependence
- The output will be always the same (reproducibility)
- Different programming languages
Final thoughts

Efficiency considerations:

• The government has a duty to reduce the public expenditure
• We have to avoid duplicated efforts and expand the automation
• We are exploring the possibility to open this API to provide a service to the general public (statistical offices, universities, NGOs, private sector).

Technical considerations:

• We have to keep in mind the harmonization across the official statistics
• It is very important monitoring changes in the fieldwork methodologies (questionnaires, data collection conditions, etc).
• Our predictions can be affected by those changes.
Contact

kilehmannm@ine.gob.cl: Klaus Lehmann - Senior Analyst
ifaglonij@ine.gob.cl: Ignacio Agloni - Project Manager

https://github.com/inesscc