Introducción a la Clasificación Automática de Texto con PySS3

Sergio G. Burdisso[1,2]

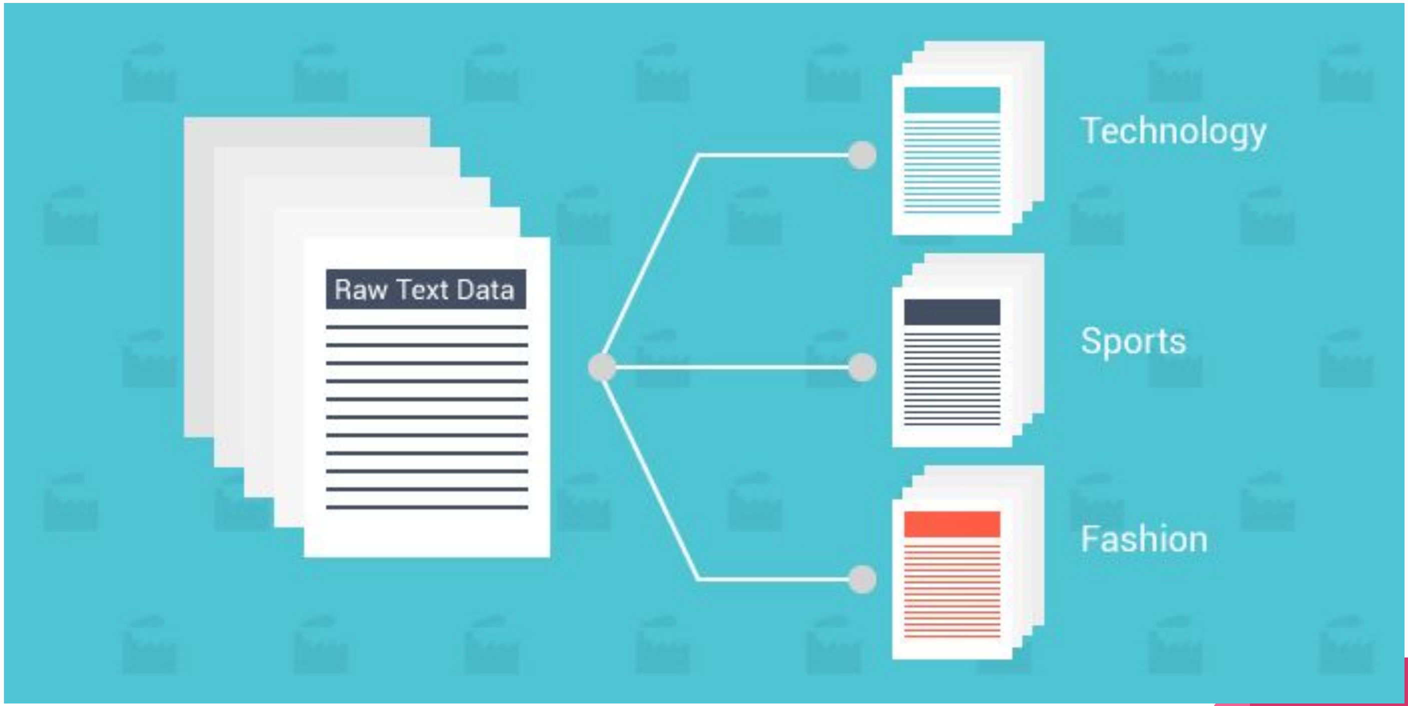
1 Universidad Nacional de San Luis (UNSL), Argentina.
2 Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), Argentina.

Outline

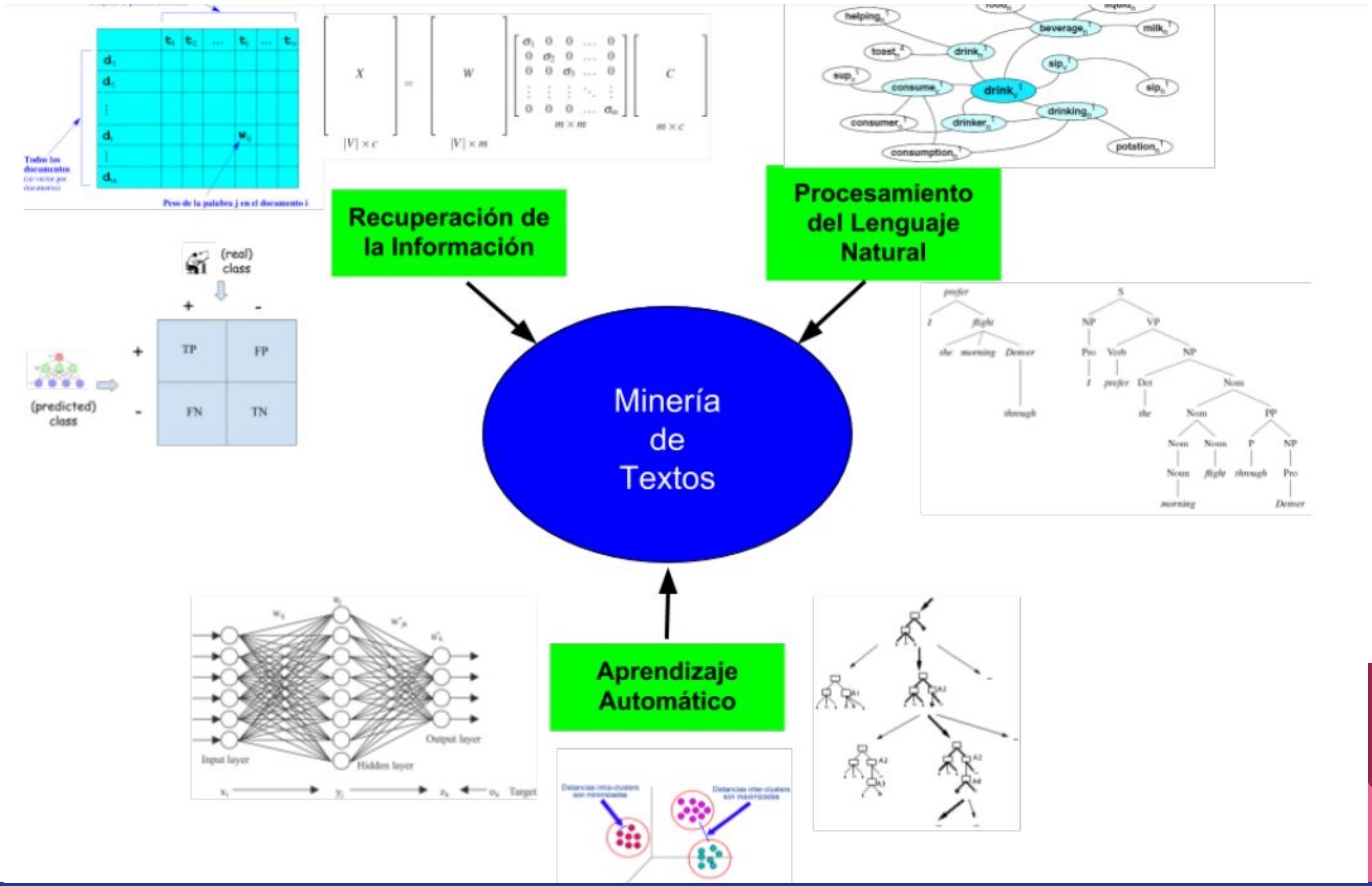
- ¿Qué es la Clasificación Automática de textos/documentos?
- ¿Qué es SS3?
- ¿Qué es PySS3?

Clasificación Automática de Texto/Documentos

Clasificación Automática de Texto



Clasificación Automática de Texto (Contexto)



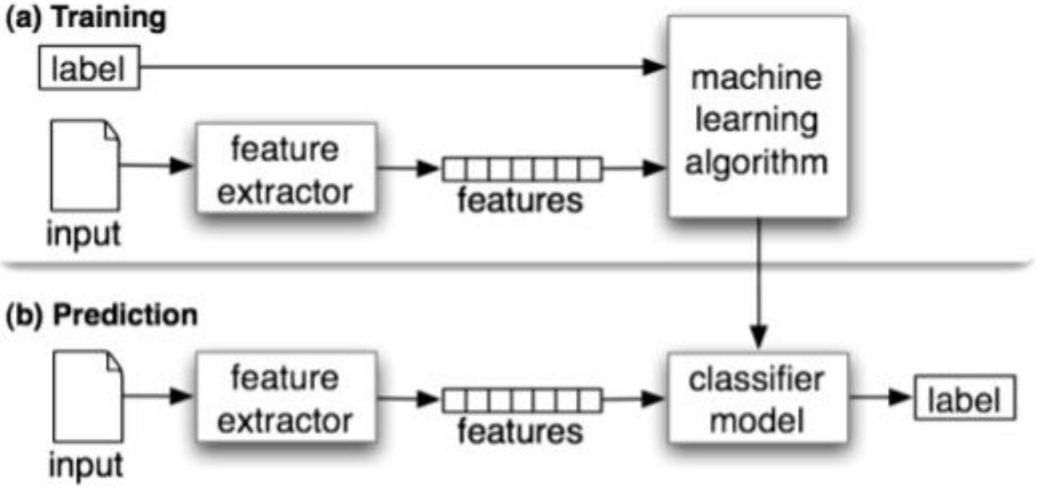
Recuperación de la Información

Procesamiento del Lenguaje Natural

Minería de Textos

Aprendizaje Automático

Clasificación Automática de Texto (Detalles)



Clasificación Automática de Texto (Ejemplos)

- Sentiment Analysis
  - **Social media monitoring:** analyze tweets and/or Facebook comments and detect if they are talking positively or negatively about a brand.
  - **Customer service:** analyze support queries to quickly detect angry and frustrated customers.
  - **Customer feedback:** analyze comments or survey responses to find if customers like or dislike particular aspects of a product or service.
- Language Detection
  - Routing customer support tickets to the correct team.
  - Sort through documents according to their languages.
  - Filtering incoming messages in undesired languages.
- Spam filtering
- Topic Categorization
- Profiling
- **Health and Safety**
  - **Early risk prediction on the Internet** (e.g. Depression, Anorexia, Self-harm, Terrorism, etc.)

The SS3 classification model

Supervised Machine Learning Model for Text Classification

The SS3 text classifier (1/3)

First, it uses a **function**, *gv*, to value **word relevance** relative to **each category**, for example:

$$\begin{array}{ll}
gv(\text{`}sushi\text{'}, food) = 0.85; & gv(\text{`}the\text{'}, food) = 0; \\
gv(\text{`}sushi\text{'}, music) = 0.09; & gv(\text{`}the\text{'}, music) = 0; \\
gv(\text{`}sushi\text{'}, health) = 0.50; & gv(\text{`}the\text{'}, health) = 0; \\
gv(\text{`}sushi\text{'}, sports) = 0.02; & gv(\text{`}the\text{'}, sports) = 0;
\end{array}$$

This function has a **vectorial version**:

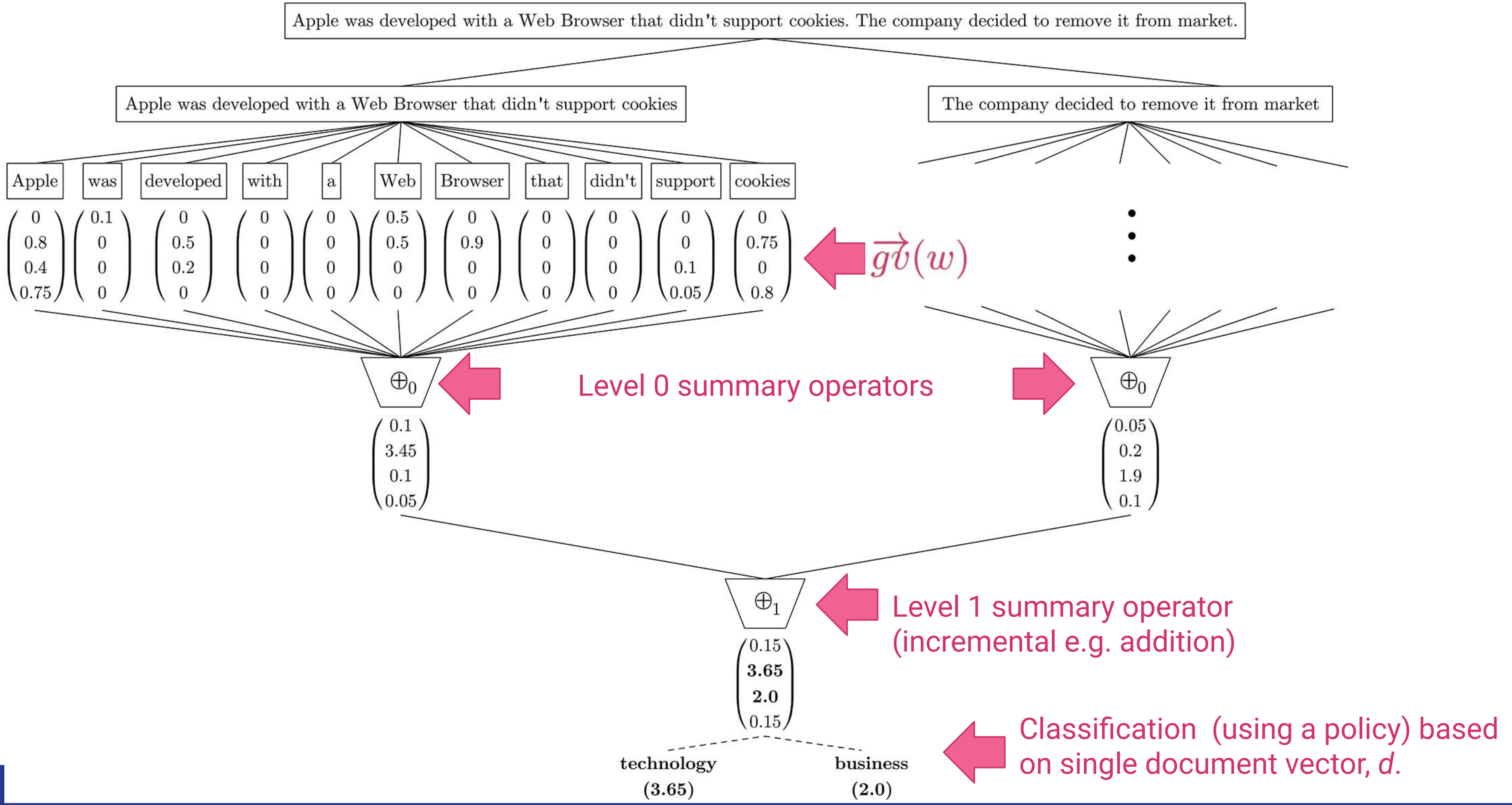
$$\overrightarrow{gv}(w) = (gv(w, c_0), gv(w, c_1), \ldots, gv(w, c_k))$$

Thus, the previous example becomes:

$$\begin{array}{l}
\overrightarrow{gv}(\text{`}sushi\text{'}) = (0.85, 0.09, 0.5, 0.02); \\
\overrightarrow{gv}(\text{`}the\text{'}) = (0, 0, 0, 0);
\end{array}$$

The SS3 text classifier (2/3)

Then, it converts the input into a **hierarchy of blocks**, computing a *confidence vector* for each hierarchy block. Classification is made using the single document *confidence vector*.



Level 0 summary operators

Level 1 summary operator (incremental e.g. addition)

Classification (using a policy) based on single document vector, *d*.

The SS3 text classifier (3/3) - Hyperparameters

$$global\ value = local\ value \cdot significance \cdot sanction$$

- σ (“Smoothness”)
- λ (“Significance”)
- ρ (“Sanction”)

```
clf = SS3(s=0.32, l=1.24, p=1.1)
```

```
clf.set_hyperparameters(s=0.32, l=1.24, p=1.1)
```

PySS3

PySS3 - ¿Qué es?

Un paquete python que implementa SS3 y que además viene con un conjunto de herramientas de desarrollo y visualización.

Compuesto por:

- Módulo Principal
- 3 Submódulos:
    - pyss3.server
    - pyss3.cmd_line
    - pyss3.util

PySS3 - SS3 class

```python
from pyss3 import SS3

clf = SS3()

clf.fit(x_train, y_train)
y_pred = clf.predict(x_test)
```

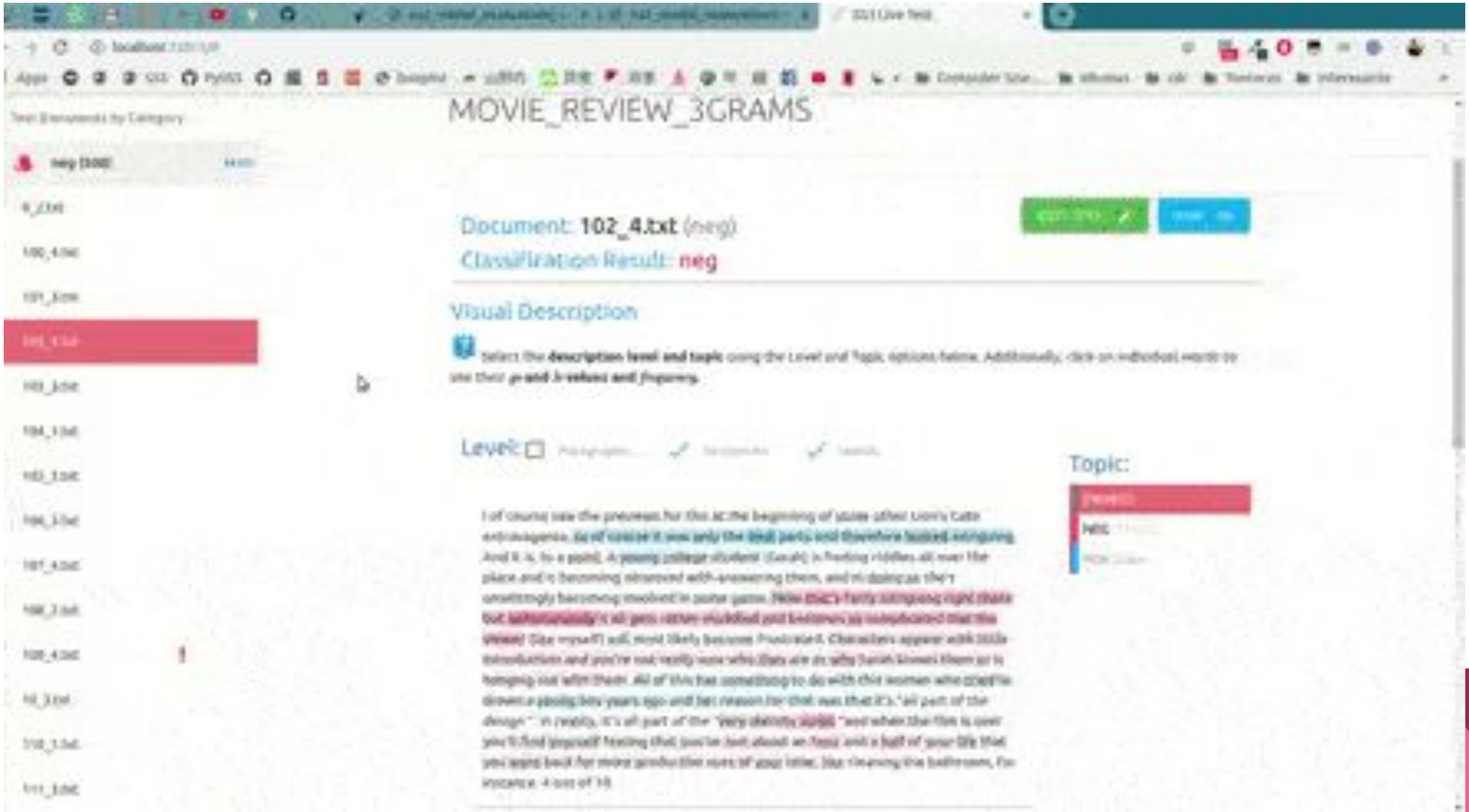
PySS3 - “Live Test” tool

```
from pyss3.server import Server
from pyss3 import SS3

clf = SS3()
clf.fit(x_train, y_train)

Server.serve(clf, x_test, y_test) # <- this one! cool uh? :)
```

PySS3 - “Live Test” tool



MOVIE_REVIEW_3GRAMS

Test Documents by Category

neg (500)

- 9_2.txt
- 100_4.txt
- 101_3.txt
- 102_4.txt
- 103_3.txt
- 104_1.txt
- 105_3.txt
- 106_1.txt
- 107_4.txt
- 108_2.txt
- 109_4.txt
- 10_3.txt
- 110_1.txt
- 111_3.txt

Document: 102_4.txt (neg)

Classification Result: neg

Visual Description

Select the **description level and topic** using the Level and Topic options below. Additionally, click on individual words to see their *gv* **and** *lv* **values and frequency.**

Level: □ Paragraphs ✓ Sentences ✓ Words

Topic:

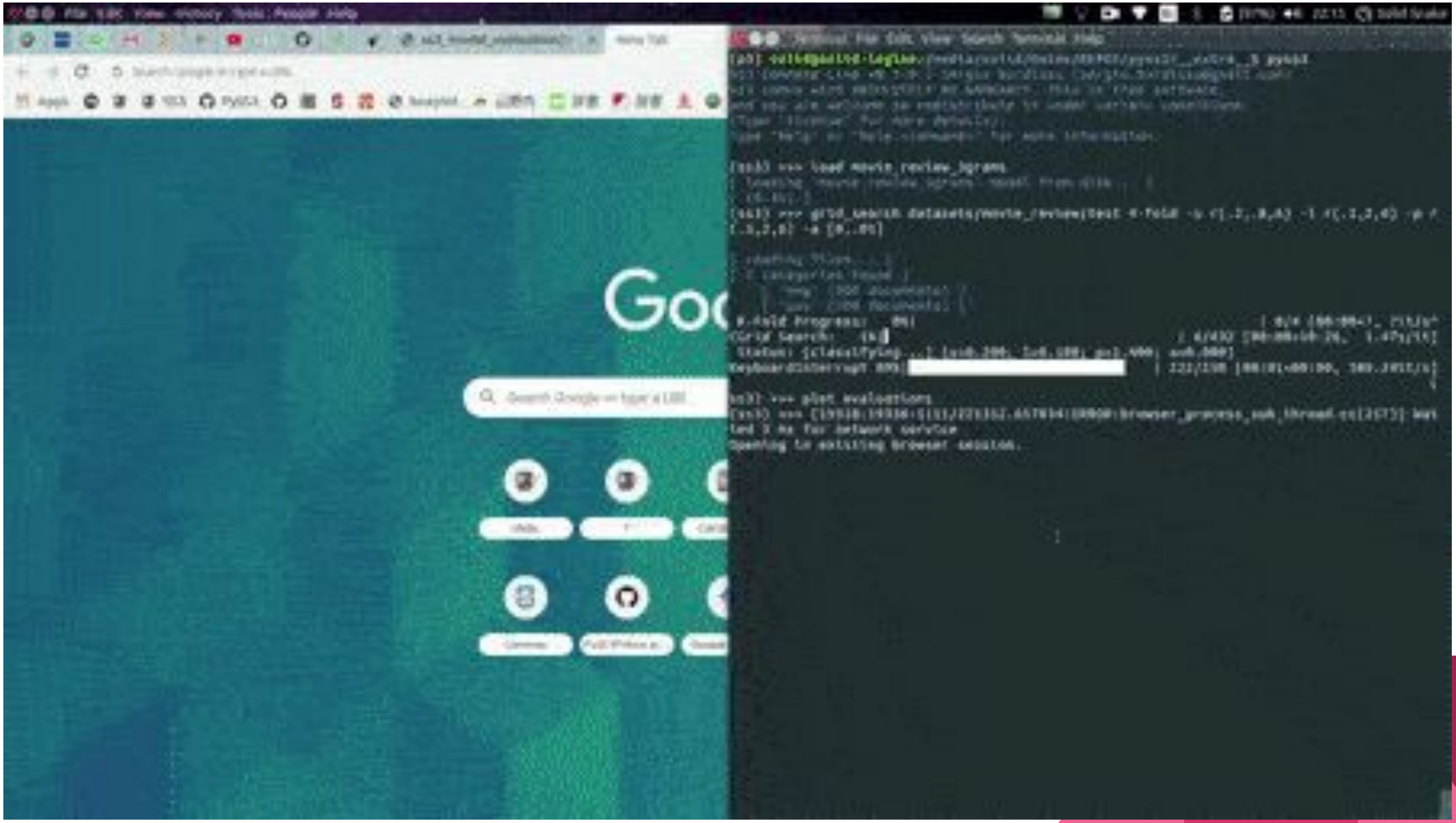
- (neg)
- NEG
- POS

I of course saw the previews for this at the beginning of some other Lion's Gate extravaganza, so of course it was only the best parts and therefore looked intriguing. [illegible]

PySS3 - Command Line

```
your@user:/your/project/path$ pyss3
(pyss3) >>> load my_model
(pyss3) >>> grid_search path/to/dataset -s [.2,.5,.8] -l [.1,1,2] -p [.5,1.5,2]
(pyss3) >>> plot evaluations
```

PySS3 - Command Line (Evaluation Plot)



File Edit View History Tools People Help
New Tab
Google
Search Google or type a URL
Terminal File Edit View Search Terminal Help
22:11

```
(p3) solid@solid-laptop:/media/solid/[illegible]/pyss3/__extra__$ pyss3
PySS3 Command Line v0.[illegible] | Sergio Burdisso ([illegible])
PySS3 comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to redistribute it under certain conditions
(type 'license' for more details).
Type 'help' or 'help <command>' for more information.

(pyss3) >>> load movie_review_3grams
[ loading 'movie_review_3grams' model from disk ... ]
[ [illegible] ]
(pyss3) >>> grid_search datasets/movie_review/test 4-fold -s r(.2,.8,4) -l r(.1,2,4) -p r(.5,2,6) -a [0,.01]

[ loading files ... ]
[ 2 categories found ]
   [ 'neg' (500 documents) ]
   [ 'pos' (500 documents) ]
 K-fold Progress:   0%|                                          | 0/4 [00:00<?, ?it/s]
Grid Search:   1%|▌                                   | 4/432 [00:00<10:26,  1.47s/it]
 Status: [classifying...] [s=0.200; l=0.100; p=1.400; a=0.000]
KeyboardInterrupt 89%|██████████████████              | 222/250 [00:01<00:00, 189.28it/s]

(pyss3) >>> plot evaluations
(pyss3) >>> [19918:19938:1111/221312.457814:ERROR:browser_process_sub_thread.cc(217)] Waited 3 ms for network service
Opening in existing browser session.
```

PySS3 - ¡Manos a la obra!

- Topic Categorization
- Sentiment Analysis on Movie Reviews