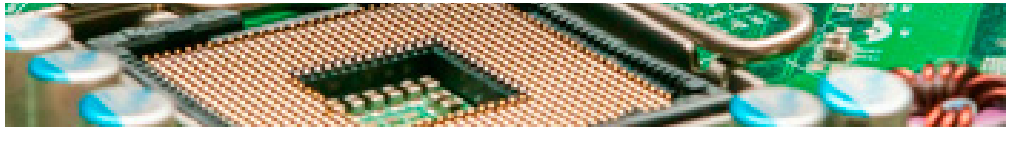
Máster Universitario en Ingeniería Informática

Sistemas Inteligentes

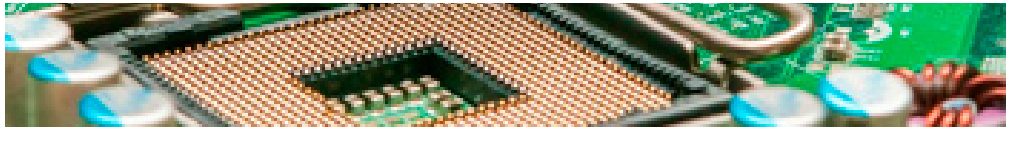
Unit 6. Maximum Entropy Models - Seminar

2022/2023

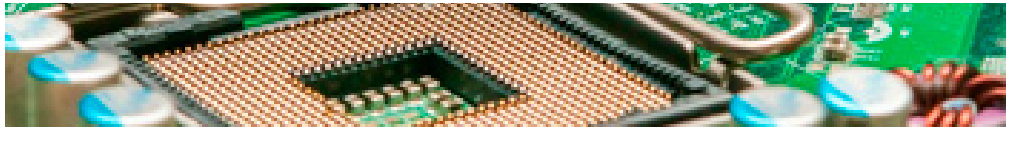


Outline

1. Introduction
2. MALLET installation
3. Introduction to MALLET
4. Creating a classifier with MALLET
5. Example: Classification
6. Student Projects



Outline



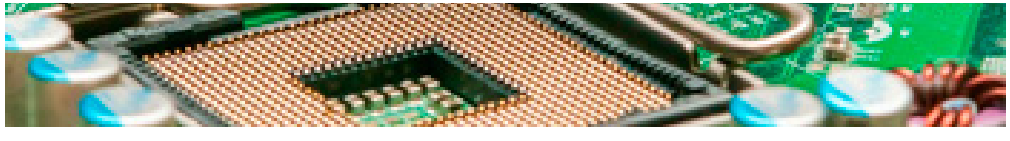
Introduction

Available ME tools:

- **MALLET**: https://mimno.github.io/Mallet/index
- NLTK: http://www.nltk.org/
- WEKA: http://www.cs.waikato.ac.nz/ml/weka/

Recommended web pages:

- http://www.cs.cmu.edu/afs/cs/user/aberger/www/html/tutorial/tutorial.html
- http://en.wikipedia.org/wiki/Principle_of_maximum_entropy
- http://www.inference.phy.cam.ac.uk/hmw26/crf/

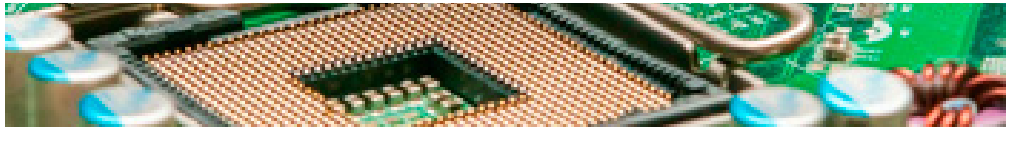


MALLET Toolkit

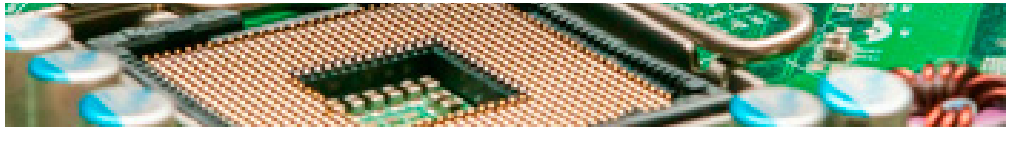
MALLET: **MA**chine **L**earning for **L**anguag**E** **T**oolkit, written by A. McCallum

MALLET is a Java-based package for

- Statistical natural language processing
- Document classification
- Clustering
- Topic modeling
- Information extraction
- . . . other machine learning applications



Outline



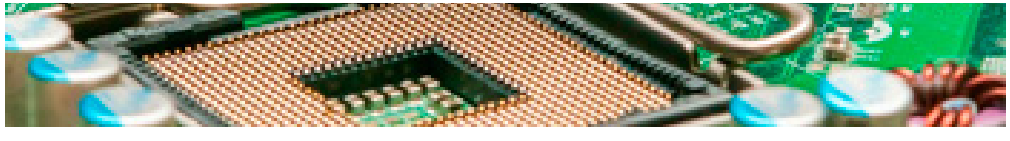
MALLET installation

Create a directory for MALLET tools:

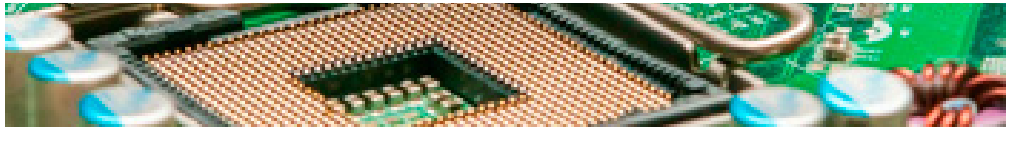
```
$ mkdir $HOME/W/mallet
$ cd $HOME/W/mallet
```

Download MALLET latest version

```
$ wget https://github.com/mimno/Mallet/releases/download/v202108/\
  Mallet-202108-bin.tar.gz
$ tar zxvf Mallet-202108-bin.tar.gz
$ cd Mallet-202108
$ ls -l
total 48
drwxr-xr-x 2 ...     0 abr 27 09:57 bin
-rwxr-xr-x 1 ...  4107 jun 13  2021 build.xml
-rwxr-xr-x 1 ...  1437 jun 13  2021 CHANGELOG.md
drwxr-xr-x 2 ...     0 abr 27 09:57 class
-rwxr-xr-x 1 ...    27 jun 13  2021 _config.yml
drwxr-xr-x 2 ...     0 abr 27 09:57 dist
drwxr-xr-x 2 ...     0 abr 27 09:57 lib
-rwxr-xr-x 1 ... 11620 jun 13  2021 LICENSE
...
```

Outline



Internal Data Representation in MALLET

MALLET keeps an internal representation for the data

For a given text, the text is transformed into a vector in which the position is defined with a mapping, and the position in the vector stores the ocurrences of each word in the text (each word is a feature)

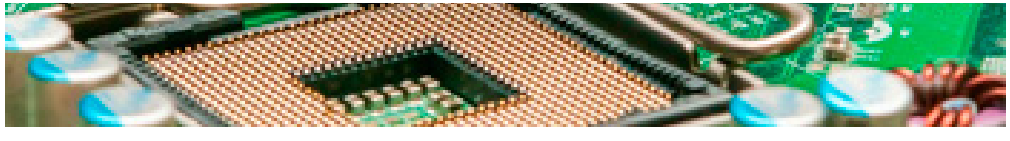
Example

Supose that MALLET uses a `hash` for the mapping:

```
f("This") = 345    f("is") = 174      f("and") = 705
f("a") = 5         f("table") = 15
f("this") = 798    f("chair") = 191
```

Then a sentence is coded as follows:

```
This  is a table and this  is a chair
345  174 5  15   705 798  174 5  191
```

Internal Data Representation in MALLET

MALLET can keep internally the information in several formats

- As a sequence of features:

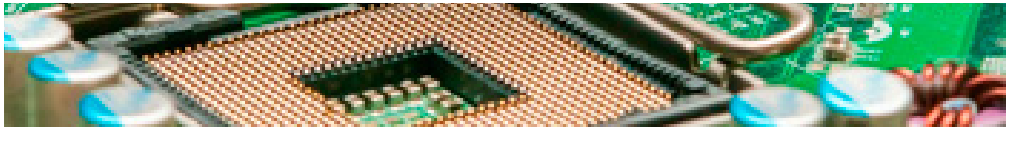
```
345, 174, 5, 15, 705, 798, 174, 5, 191
```

- As a bag of features:

```
5 15 174 191 345 705
2  1  2   1   1   1
```

MALLET registers the following information of each instance:

- Instance name
- Data (as explained above)
- Label
- Source (the original data)



MALLET Characteristics

MALLET supports two different ways of working: scripts and Java classes

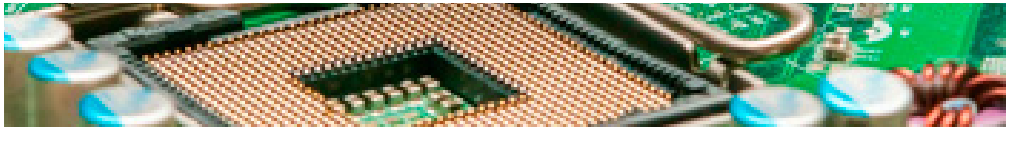
- Scripts: develop full processes
- Classes: allow to implement your own processes based on basic tools

We will mainly use scripts

Config:

```
$ export PATH=$HOME/W/mallet/Mallet-202108/bin:$PATH
$ export CLASSPATH=$HOME/W/mallet/Mallet-202108/
```

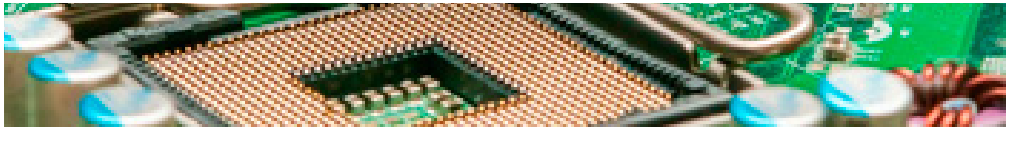
It is recommended to put these lines in your `.bashrc` file



MALLET Characteristics

Scripts use:

```
$ mallet
Unrecognized command:
Mallet 2.0 commands:

  import-dir          load the contents of a directory into mallet ...
  import-file         load a single file into mallet instances (one ...
  import-svmlight     load SVMLight format data files into Mallet ...
  info                get information about Mallet instances
  train-classifier    train a classifier from Mallet data files
  classify-dir        classify the contents of a directory with a ...
  classify-file       classify data from a single file with a saved ...
  classify-svmlight   classify data from a single file in SVMLight ...
  train-topics        train a topic model from Mallet data files
  infer-topics        use a trained topic model to infer topics for ...
  evaluate-topics     estimate the probability of new documents ...
  prune               remove features based on frequency or ...
  split               divide data into testing, training, and ...
  bulk-load           for big input files, efficiently prune ...

Include --help with any option for more information
```

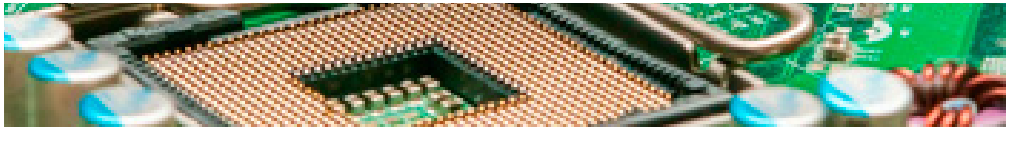
MALLET Characteristics

Java classes use:

```
$ java -cp "$CLASSPATH/class/:$CLASSPATH/lib/mallet-deps.jar" \
  cc.mallet.classify.tui.Csv2Vectors --help
A tool for creating instance lists of feature vectors from
comma-separated-values
--help TRUE|FALSE
  Print this command line option usage information. Give argument of
   TRUE for longer documentation
  Default is false
--prefix-code 'JAVA CODE'
  Java code you want run before any other interpreted code. Note that
   the text is interpreted without modification, so unlike some other
...
--print-output [TRUE|FALSE]
  If true, print a representation of the processed data
   to standard output. This option is intended for debugging.
  Default is false
```

Option `--cp` specifies the CLASS PATH variable

The API is available at http://mallet.cs.umass.edu/api/



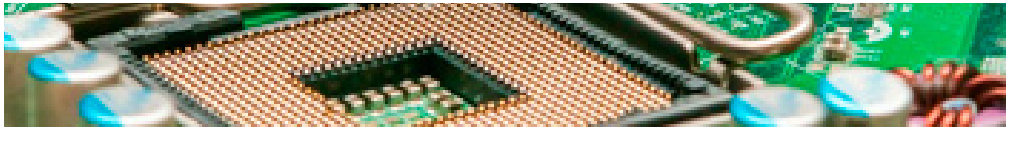
Importing Data for MALLET

MALLET can read data from directories and files

For directories: MALLET will use the directory names as labels and the filenames as instance names

Convert some data to MALLET format (script)

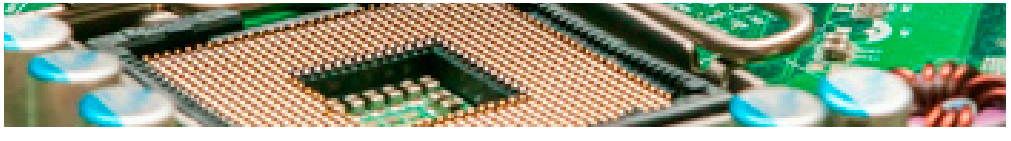
```
$ cd $HOME/W/mallet/Mallet-202108/
$ tree sample-data/web/
sample-data/web/
|-- de
|---|-- apollo8.txt
...
|-- en
    |-- elizabeth_needham.txt
    ...

$ mallet import-dir --input sample-data/web/* --output web.mallet
Labels =
   sample-data/web/de
   sample-data/web/en
```

Importing Data for MALLET

Convert some data to MALLET format (Java classes)

```
$ java -cp "$CLASSPATH/class/:$CLASSPATH/lib/mallet-deps.jar" \
  cc.mallet.classify.tui.Text2Vectors \
  --input sample-data/web/* --output web.mallet
Labels =
   sample-data/web/de
   sample-data/web/en
```

Importing Data for MALLET

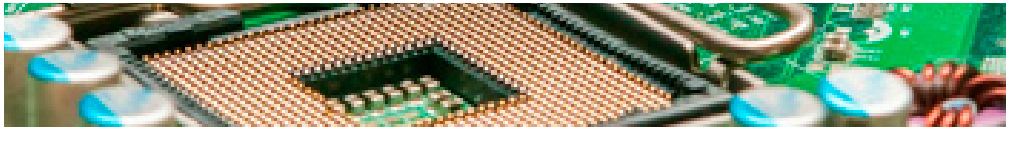
For files: one file with one line per instance, format is similar to this:

```
instance0 label0 w01 w02 ...
instance1 label1 w11 w12 ...
...
```

The MALLET script commands are:

```
$ csv2vectors --input sample-data/numeric/boxes.txt \
  --output boxes.vectors --token-regex '[\p{L}\p{N}\p{P}]+'
$ mallet import-file --input sample-data/numeric/boxes.txt \
  --output boxes.mallet --use-pipe-from boxes.vectors \
  --token-regex '[\p{L}\p{N}\p{P}]+'
```

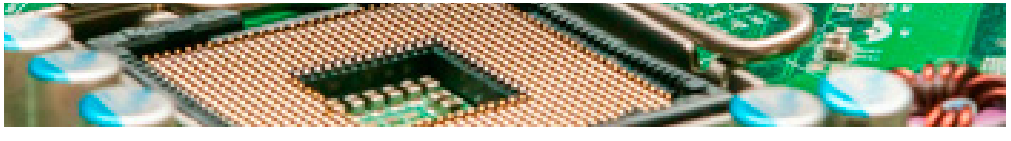
Option `--use-pipe-from` specifies that the word coding is stored in the file `boxes.vectors` for further use in other files different from file `boxes.txt`



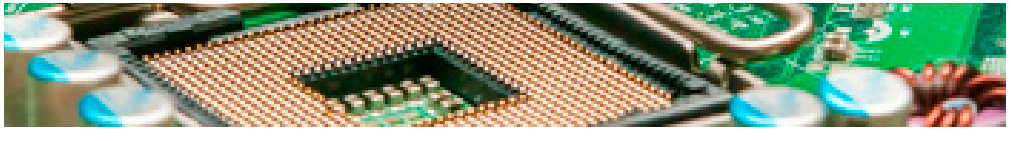
Importing Data for MALLET

Other relevant options are:

- `--keep-sequence`: If true, final data will be a FeatureSequence rather than a FeatureVector (default is false)
- `--preserve-case` Do not force all strings to lowercase (default is false)
- `--token-regex`: To define tokens!! VERY IMPORTANT!!



Outline

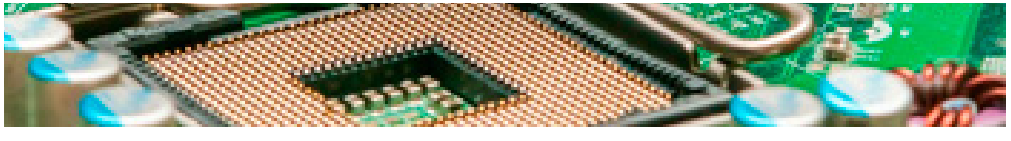


Creating a classifier with MALLET

For training a ME classifier with MALLET, we can proceed as follows:

```
$ mallet train-classifier --input web.mallet --trainer MaxEnt \
  --output-classifier web_MaxEnt.cl 2> /dev/null

-------------------- Trial 0  --------------------

Trial 0 Training MaxEntTrainer,gaussianPriorVariance=1.0 with 48
 instances
Trial 0 Training MaxEntTrainer,gaussianPriorVariance=1.0 finished
Trial 0 Trainer MaxEntTrainer,gaussianPriorVariance=1.0 training data
 accuracy = 0.75
Trial 0 Trainer MaxEntTrainer,gaussianPriorVariance=1.0 Test Data
 Confusion Matrix
...
Summary. test recall(de) mean = 1.0 stddev = 0.0 stderr = 0.0
Summary. test recall(en) mean = 1.0 stddev = 0.0 stderr = 0.0
Summary. test f1(de) mean = 1.0 stddev = 0.0 stderr = 0.0
Summary. test f1(en) mean = 1.0 stddev = 0.0 stderr = 0.0
```

Additional options can be seen with `mallet train-classifier --help`

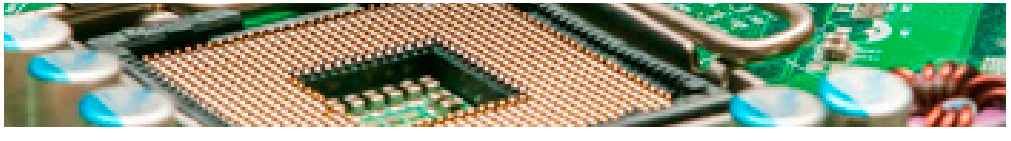


Creating a classifier with MALLET

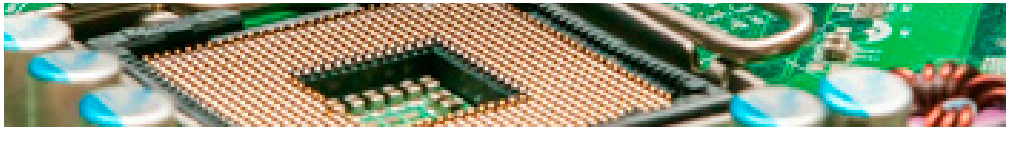
The classifier is stored in binary format, it can be converted to text format:

```
$ classifier2info --classifier web_MaxEnt.cl
FEATURES FOR CLASS de
 ...
 und 0.09942049183147898
 erste 0.0018041346889153833
 ...
 potential -8.087763378383716E-4
 given -8.087763378383716E-4
 ...
FEATURES FOR CLASS en
 ...
 und -0.09942049183146215
 erste -0.0018041346889152634
 ...
 potential 8.087763378392068E-4
 given 8.087763378392068E-4
 ...
```

Each line has the weight associated to each feature.



Outline



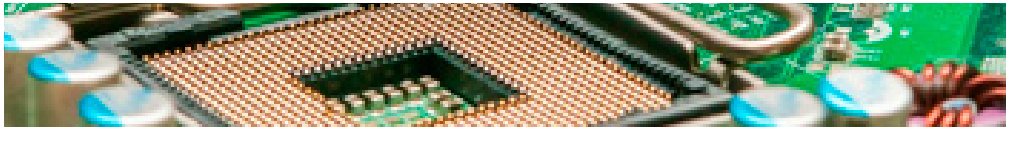
Chromosome classification

Let us create a ME classifier for the chromosome task (download `data.tgz` from PoliformaT)

```
$ tar zxvf data.tgz
$ cd data
$ csv2vectors --input cromosTr --token-regex '[A-Za-z=]' \
  --preserve-case --output cromosTr.vectors
$ mallet import-file --input cromosTr --output cromosTr.mallet \
  --use-pipe-from cromosTr.vectors --token-regex '[A-Za-z=]'
$ mallet train-classifier --input cromosTr.mallet --trainer MaxEnt \
  --output-classifier cromosTr.classifier
$ mallet import-file --input cromosTe --output cromosTe.mallet \
  --use-pipe-from cromosTr.vectors --token-regex '[A-Za-z=]'
$ vectors2classify --input cromosTr.classifier --training-file \
  cromosTr.mallet --testing-file cromosTe.mallet --trainer MaxEnt \
  --report test:confusion
```

Additional options can be obtained with:

```
$ vectors2classify --help
```

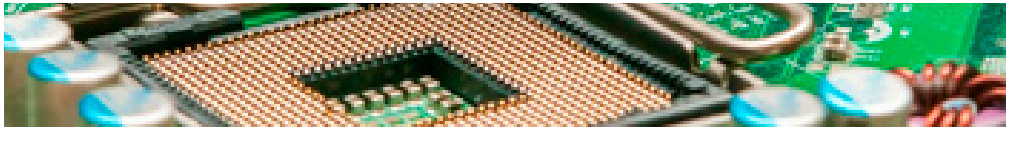



Chromosome classification

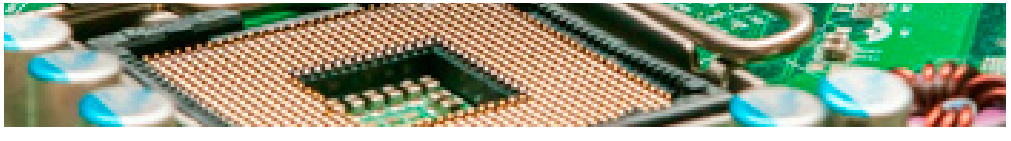
You can train a classifier and then to use it as follows:

```
$ mallet train-classifier --input cromosTr.mallet --trainer MaxEnt \
  --output-classifier cromosTr.classifier

$ mallet classify-file --input cromosTe --output - \
  --classifier cromosTr.classifier
```

But in such case, the test results are not reported directly and you have to compute them using a script

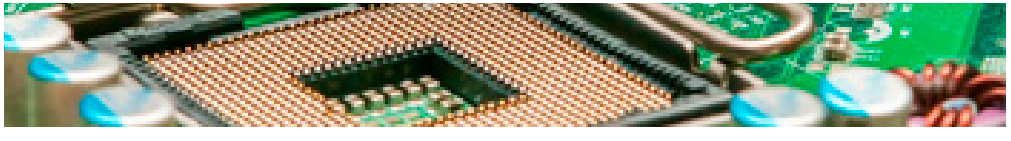


Outline



Student Projects (Exercise B2.3)

1. Download the `Stance-IberEval2017-training-20170320.zip` that is available in PoliformaT (the password is in the PoliformaT link)
2. Download evaluation tools as explained in the `overview-task-stance.pdf` that is available in PoliformaT (see modifications in PoliformaT as well)
3. Train ME classification models for stance classification for Spanish and for Catalan and upload the results on a blind test set that will be available on due time through a PoliformaT task. The task will open on Monday 5th June 2023 at 19:00 and will be closed on Friday 16th 2023 at 19.00
4. The task will describe the format for providing the results
5. Max mark: 25 out of 35



Student Projects

The final mark M would be according to the obtained F-score rate (FC_o and FE_o) for each task with respect to the best result reported in `overview-task-stance.pdf` ($FC_b = 0.4901$ and $FE_b = 0.4888$):

$$M = \min\left(25,5 + 10\frac{FC_o}{0.9FC_b} + 10\frac{FE_o}{0.9FE_b}\right)$$