NLP course 2022

Homework 1

Named Entity Recognition

Prof. Roberto Navigli

Teaching assistants:
Andrea Bacciu, Stefan Bejgu,
Abelardo Carlos Martinez Lorenzo,
Valerio Neri, Riccardo Orlando,
Alessandro Scirè, Simone Tedeschi

SAPIENZA
UNIVERSITÀ DI ROMA

Named Entity Recognition

An introduction

What is NER?

- NER is the task of **locating** and **classifying** named entities mentioned in text.
  - It is a sequence labeling task, a.k.a. every token must be tagged with some label
  - Six labels (plus **O** tag for no named entity)[1]:

PER: Person	**CORP**: Corporation
LOC: Location	**PROD**: Product
GRP: Group	**CW**: Creative Work

[1]The number of classes depends on the dataset.

Model

Model: possible approaches

You can start from the bonus exercise model!

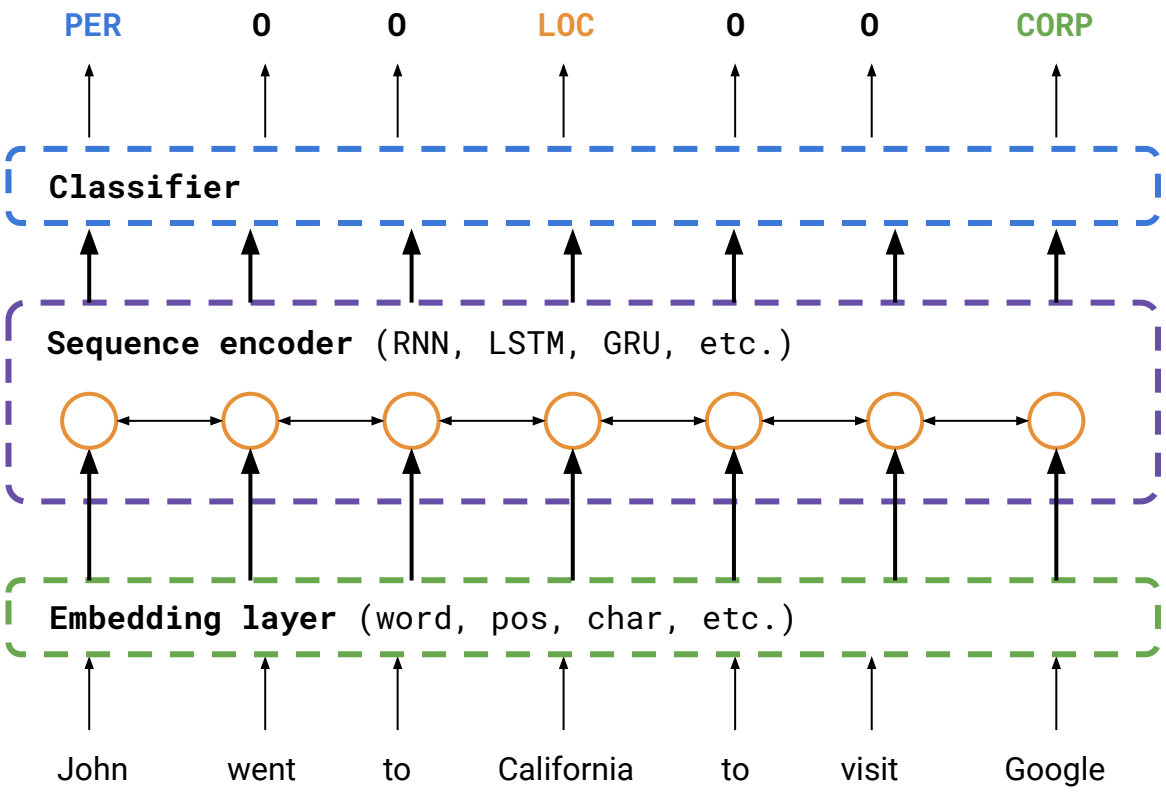
- You probably need to **encode the sequence**
  - RNNs 👀
  - ...

Hint: Plain accuracy probably is not the best measure to use... Why?

- **Look at the data**: things are going to be clearer!

Note: for this homework, **Transformer-based** architectures **are not allowed**

Model: possible approaches



Submission

What you will receive

- We will provide you with a folder organized as follows (some files are omitted):

```
- nlp2022-hw1/
  - data/
  - hw1/
    - model.py
    - stud/
  - model/
  - requirements.txt
  - test.sh
```

- You are allowed to edit **only** the items in bold!

SAPIENZA NLP

What you will receive

- We will evaluate your work using Docker
  - You should be fine even if you don't know anything about it
- If **`test.sh`** runs on your side, it will run on ours as well
  - Just keep in mind: do not change any file but those we marked in bold as editable in the previous slide
- Additionally, we wrote a **`README.md`** to get you everything up and running
- You can find the code repository here!

What we expect from you

- The zip folder we gave you (but populated :))
- Put your training code (if you used Colab, download the notebook `.ipynb` and place it) in **`hw1/stud/`**
- If you use any additional library, modify the **`requirements.txt`** file as needed (click here for info)
- Use the data (train and dev) in the data folder
  - use each file as defined in the **standard ML conventions** (*train for training*, *dev for model selection*)

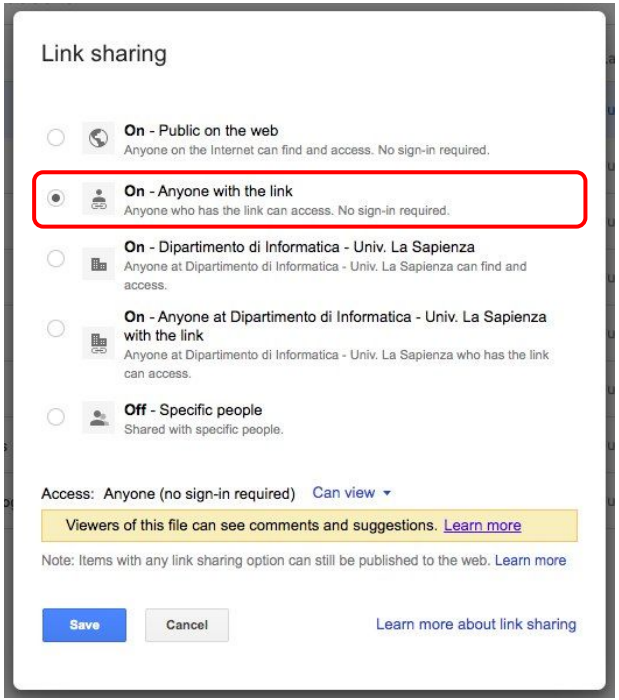
What we expect from you

- Put **everything** your model needs (vocabulary, weights, ...) inside the **`model/`** folder, and **be sure to properly load them** in your model
- In **`hw1/stud/implementation.py`** implement the **`StudentModel`** class
  - Load your model and use it in the **`predict`** method
  - You **must respect** the signature of the predict method!
  - You can add other methods (i.e. the constructor)
- In **`hw1/stud/implementation.py`** implement the **`build_model`** function
  - It should initialize your **`StudentModel`** class.

What we expect from you

- Use **`test.sh`** to check that everything works
- Add your **`report.pdf`** to the folder (yes, export it in PDF even if you are using Word!)
- Name the zip folder **`lastname_studentid_hw1.zip`**:
  - Ex: Luigi D'Andrea will submit a file named **`dandrea_1234567_hw1.zip`**
  - If you are unsure which name to put, use the one in your institutional email account

Submission Instructions



- Upload the zip on your **institutional** Drive and make it **link-shareable** and **public** to anyone (an automatic script will download it).
- Make sure it is accessible via an incognito page of your browser!
- Do **NOT modify** the folder structure
- You have to submit the homework through the submission form on Google Classroom. You will be asked to fill a form with the requested information and the **link** to the zip you uploaded on Drive.

SAPIENZA NLP

Dataset and Evaluation

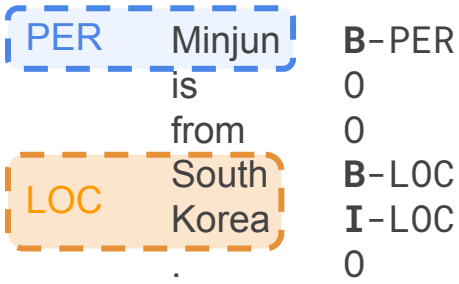
The Dataset

- The dataset focuses on detecting **semantically ambiguous** and **complex entities** in short and low-context settings
- It contains
  - **14535** training examples
  - **765** development examples
- A label can cover a **span** of words
  - Usually it is addressed using the `BIO` tags
  - So there are actually **13** classes: `B-PER`, `B-LOC`, `B-GRP`, `B-CORP`, `B-PROD`, `B-CW`, `I-PER`, `I-LOC`, `I-GRP`, `I-CORP`, `I-PROD`, `I-CW` and `O`

What is BIO format?

BIO stands for **Beginning**, **Inside** and **Outside** of a text segment that compose a Named Entity

- **B-** prefix before a tag indicates that the tag is the beginning of a chunk
- **I-** prefix before a tag indicates that the tag is inside a chunk
- **O-** tag indicates that a token belongs to **NO** entity/chunk



The dataset

```
#   id  0
ease    B-CW
on  I-CW
down    I-CW
the I-CW
road    I-CW
–   O
charlie B-PER
smalls  I-PER
(   O
diana   B-PER
ross    I-PER
and O
michael B-PER
jackson I-PER
in  O
the B-CW
wiz I-CW
)   O

#   id  1
it  O
was O
republished O
by  O
mit B-CORP
```

- The dataset is a TSV where each line is a word tab-separated with its label
  - There is a header at the beginning with the sentence ID
  - The end of a sentence is marked with an empty line

Evaluation

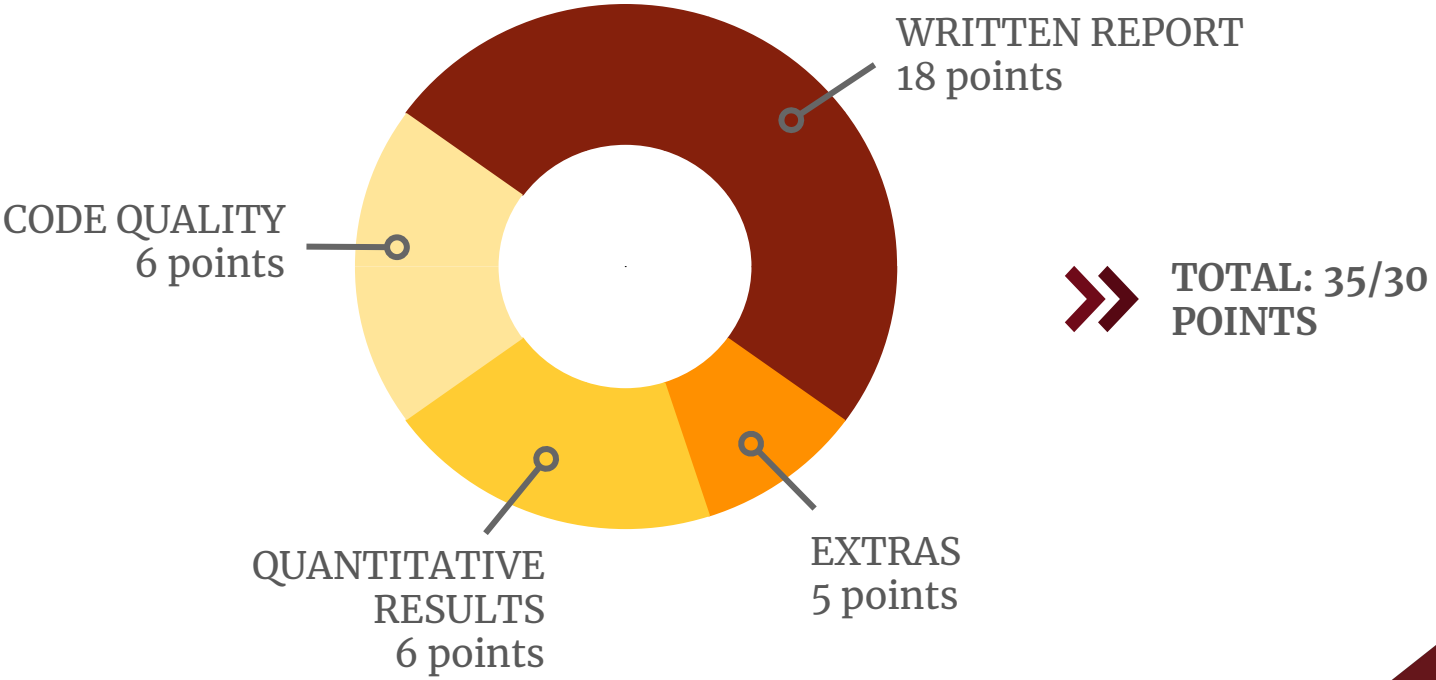
- The evaluation will be conducted on a **SECRET** test set
- We will use **`seqeval`** to compute the F1 of your systems:

$$F_1 = 2 \cdot \frac{\text{recall} \times \text{precision}}{\text{recall} + \text{precision}}$$

- Use the validation split to select the best model/hyperparameters configuration

Evaluation

We will take into account the following criteria:



TOTAL: 35/30 POINTS

SAPIENZA NLP

Report: dos and don’ts

- **ACL 2021 paper template**
  - Available here (Word and LaTeX direct download) or here (Overleaf LaTeX template)
  - You can use either the LaTeX or the Word template, your choice
  - **DO NOT MODIFY** the template (margins, spacing, font size)
  - Use the non-anonymous flag, so you can enter your name
- **Max 2 pages**
  - For the report, including title, subtitles, etc.
  - This is a **STRICT RULE!**
- **Unlimited extra pages for images, tables and references**
  - Be sure to **include** and properly **comment** a confusion matrix, visualized as heat map
  - Every image and table must have a caption (don’t abuse them please :))
  - Tables and images must be referenced in the report

Report: what you are expected to do

We expect a good report to be:

- **Readable** and **understandable**
  - We will not give penalties for English errors, but we expect the report to follow a clear flow. We don't want to read just a sequence of statements on what you did without showing the reasoning behind your choices
- **Well-structured** and **organized**
  - Take inspiration from the many papers available online and organize your report in well-defined sections (e.g. method, setup, experiments, results...)

Report: what you are not expected to do

We expect a good report **NOT** to include:

- Unnecessary **task** or **dataset descriptions**
  - just focus on your solution to the problem.
- **Code** copy-paste
  - Your code should be self-explanatory, so no need to show it in the report. You can add **pseudocode** to show some particular algorithm, but **no code or screenshots,** please!

Report: what you are not expected to do

We expect a good report **NOT** to include:

- Unnecessary **low-level implementation details**
  - Avoid any **low-level implementation/technical details** like “I used a dictionary to store these values”, “I had to use configuration X to solve this exception”, “I could not use Y because there was a dependency issue with Z”, etc.
  - Instead, **we are interested in high-level abstractions/strategies** you decide to use to tackle the homework, as well as the **intuitions behind your choices**.

    E.g. use and description of a particular model, explanation of how and why an architecture works, etc.

Application: what you are expected to do

Your project should conform to the following rules:

- You **MUST** use PyTorch.
  - TensorFlow and other deep learning frameworks are **NOT** allowed.
  - PyTorch Lightning is **NOT** allowed (at this stage)
- **Frameworks** that use PyTorch (e.g. AllenNLP, torchtext…) are **NOT** allowed.
- Libraries (such as tqdm, sklearn, NLTK) are fine, but since the line between a framework and a library is sometimes blurred, please ask in the Google Classroom group before using any external library: **any other** library **MUST be agreed with the TAs**.

Application: what you are not expected to do

Your project should conform to the following rules:

- **You are not allowed** to use tools/architectures that have not been explained **yet** in the course, in particular:
  - word embeddings (Word2Vec, GloVe, etc.) **are allowed**,
  - contextualized word embeddings (ELMo, etc.) **are NOT allowed**,
  - Transformer-based models (BERT, BART, RoBERTa, etc.) **are NOT allowed**.
- For any doubt, please ask the TAs on Google Classroom.
- **Comment** your code, please!

Quantitative Results

We will evaluate the **performance of your model** on a SECRET test set.

You can get **from 0 to 6** points according to the following **thresholds**:

- `P < 0.15 => FAIL`
- `0.15 < P < 0.30 => 0`
- `0.30 < P < T2 => 1`
- `T2 < P < T3 => 2`
- `T3 < P < T4 => 3`
- `T4 < P < T5 => 4`
- `T5 < P < T6 => 5`
- `P > T6 => 6`

Quantitative Results

We will evaluate the **performance of your model** on a SECRET test set.

You can get **from 0 to 6** points according to the following **thresholds**:

- `P < 0.15` => `FAIL`
- `0.15 < P < 0.30` => `0`
- `0.30 < P < T2` => `1`
- `T2 < P < T3` => `2`
- `T3 < P < T4` => `3`
- `T4 < P < T5` => `4`
- `T5 < P < T6` => `5`
- `P > T6` => `6`

Thresholds will be defined based on an internal reference model and the **normalized distribution of YOUR scores**!

Extras

You can achieve **up to 5 points with some extras!**

An “extra” is whatever you decide to add to your model to make it better. For instance:

- use of pre-trained embeddings, PoS embeddings, N-grams, char embeddings, CRF, ...
- use of NLP best practices,
- comparative analysis of results in your report,
- informative plots in your report,
- **new ideas** (including using external resources in a clever way)

and more, according to internal baselines. Don’t forget to **explain your choices** in the report! Extras that are not explained in the report will not be considered for evaluation.

Evaluation

- `test.sh` is identical to what we will be using
- **If it does not run on your side, we will not correct your homework**
- Note that, if you use **any kind of hard-coded paths**, this script **won't work**
- Use **paths relative to the project root folder**, e.g.:
  - **NO:** `/home/pincopallino/my_folder/model/weights.pt`
  - **OK:** `model/weights.pt`

Warnings

Things you should be aware of

Please be aware that

This is an **individual exercise!** Collaboration among the students is **not** allowed.

We will check for **plagiarism** both manually and automatically.

It is **not allowed** to:

- Copy from other students
- Share your code with other students
- Copy from online resources (StackOverflow, GitHub, Medium, Kaggle and so on).

However, you are allowed to use material from **external sources** as long as it is **not central** to the homework.

You are also allowed to use the **SOME** parts of the presented class notebooks. However, you **MUST** explicitly specify these parts in your code comments.

WARNING if a notebook uses torchtext or pytorch-lightning you cannot use it

Use of external data

- For your experiments, **use the provided data** (train and dev) in the data folder; use each file as defined in the standard ML conventions (train for training, dev for model selection).
- If you train it using external data, include it in the data folder.
- If you train it on dev set, it will be a **FAIL**.

Tips

A few tips to organize your work:

- Start **as soon as possible**!
  - Training a neural network requires time, possibly hours, depending on your hardware
- Start **small**!
  - If you don't get decent results with a very simple neural network, there is a good chance that adding other things won't make your model perform better
- Leave some time for **hyperparameter** tuning!
  - Sometimes good hyperparameter combinations can do wonders for your neural network
- Use **Google** Colab (free GPUs!)

Deadline

When to deliver what

Deadline

Submission date: **April 17th, 2022 (Sunday)**
23:59:59 Italian time (UTC + 1)

Submit the homework through the submission form on Google Classroom. You have to fill the form with the requested information and a link to the zip folder of the homework on Google Drive.

Late submission policy

Late Submission date: until **April 23rd, 2022 (Sunday)**
23:59:59 Italian time (UTC + 1)

1 point penalty will be applied for each day of delay, e.g.:

- A student delivers their homework on April 20th -> max possible grade 35 - 3 = 32
- A student delivers their homework on April 23rd -> **FAIL**!

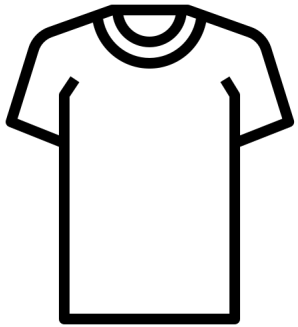
Awards 

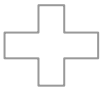
Get a **Sapienza NLP™** t-shirt

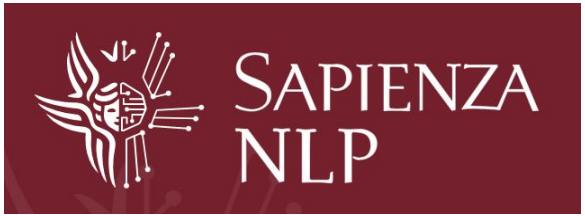
Win a Sapienza NLP t-shirt!

We will hand out amazing Sapienza NLP t-shirts to the **overall top-5** students!

The final ranking will be computed according to the scores on our **secret** test set.







SAPIENZA
NLP

That’s not all

If your work is novel, interesting and original, we will gladly invite you to work together with us to extended on a fully-fledged paper for **TOP-TIER INTERNATIONAL CONFERENCE**!

Just over the last 12 months, the Sapienza NLP group published more than a dozen of papers!

Questions?

If you have a question that may interest your colleagues, **please ask it on Google Classroom.**

Otherwise, for personal or other questions, email **ALL** of us (but please, only reach for things that can't be asked on the Google Classroom).

Our emails are:

{bacciu, bejgu, neri, orlando, scire, tedeschi}@diag.uniroma1.it

Good Luck!!