NLP 2024
Homework 1a instructions

Evaluation dataset creation for Generative Language Models
FINAL - Updated on March 27th

From EVALITA and SemEval datasets to a LLM evaluation suite

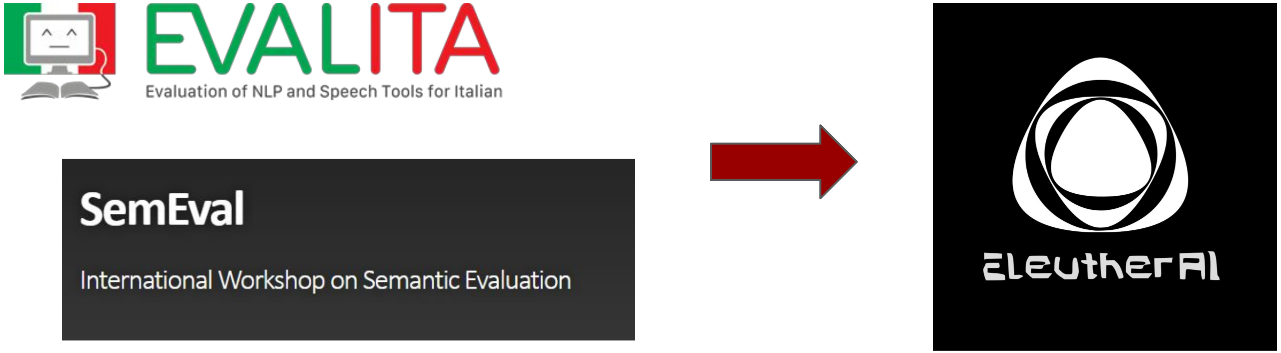
Slides provided by:

- Roberto Navigli
- Luca Gioffrè
- Lu Xu
- Luca Moroni
- Tommaso Bonomo
- Alessandro Scirè

SAPIENZA NLP

Goal

- You will start from existing evaluation datasets
- You will convert them into a format useful to evaluate the linguistic skills of Large Language Models (LLMs)



EVALITA

Evaluation of NLP and Speech Tools for Italian

SemEval

International Workshop on Semantic Evaluation

EleutherAI

Data source 1: EVALITA

- EVALITA is a periodic evaluation campaign of Natural Language Processing (NLP) and speech tools for the Italian language
- https://www.evalita.it
    - We are not using speech datasets, only text/NLP datasets

Data source 2: SemEval

- An international competition for semantic evaluations held every year
- There are several datasets coming from different SemEval editions
  - Word Sense Disambiguation
  - Multilingual and Cross-lingual Word-in-Context
  - Lexical substitution
  - More...
- Checkout the tasks here (https://en.wikipedia.org/wiki/SemEval) and for recent tasks here (https://semeval.github.io/)

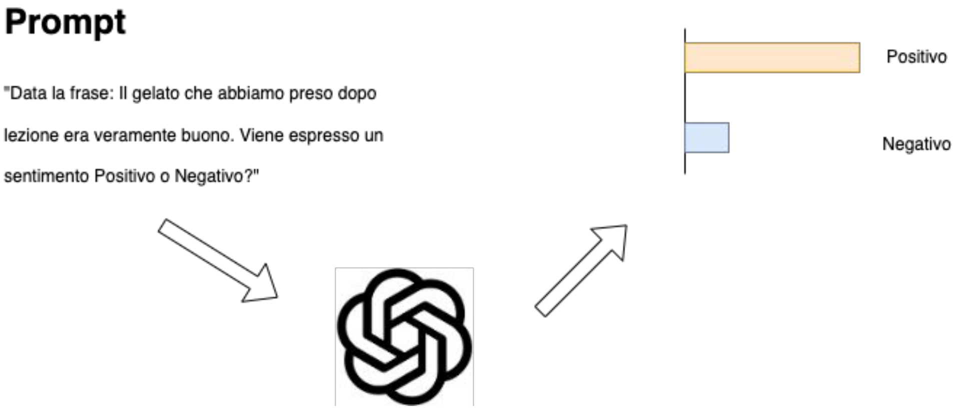
From different sources to multi-choices

- The datasets that you will find in EVALITA and SemEval campaigns come in various formats and are not suited to test Generative LLMs
- Your assignment is to reframe each task so that an LLM can receive a **sentence and a prompt as input** and produce a **single answer as output**

How an LLM works

Take a simple task, **Sentiment Analysis of a sentence**

- `sentence`: *“Il gelato che abbiamo preso dopo lezione era veramente buono”*
- `sentiment`: *“positivo”*
- `prompt`: *“Considera la frase: {{input_sentence}}. Viene espresso un sentimento positivo o negativo?”*



Prompt

"Data la frase: Il gelato che abbiamo preso dopo lezione era veramente buono. Viene espresso un sentimento Positivo o Negativo?"

Reframe as multi-choice

You have to **reframe the tasks as multi-choice Question Answering (QA)**

There are several possible tasks:

- **Word or sentence** level tasks
- **Binary or multi-categorization** tasks
- **Natural Language Generation (NLG)** tasks

For each task you have to define a suitable **prompt!**

Prompt Engineering

- The meticulous design of the input provided to an LLM to solve a task.
- The aim is to **improve the performance and manage the behavior of LLMs.**
- Along with the dataset reformatting **you have to provide different prompts** for each of defined task.

Here are some examples

Sapienza NLP

Binary sentence level task: BoolQ

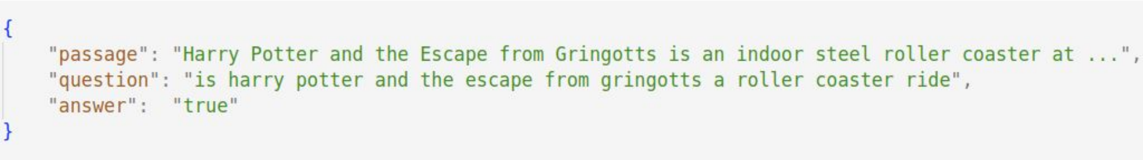
EXAMPLE

A binary sentence-level task with an input sentence and a binary output

BoolQ: Hugging Face link | BoolQ is a QA dataset for yes/no questions.

Step 1: transform the input item into a JSON containing passage, question, the possible answers and the correct one.

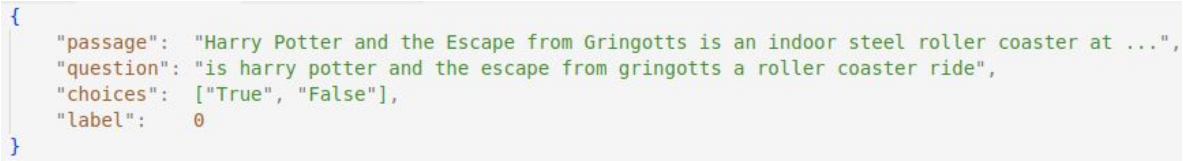
From this…



```
{
    "passage": "Harry Potter and the Escape from Gringotts is an indoor steel roller coaster at ...",
    "question": "is harry potter and the escape from gringotts a roller coaster ride",
    "answer":  "true"
}
```

This example is in English, but you will use Italian in all fields (also `choices`)!

… to this



```
{
    "passage":  "Harry Potter and the Escape from Gringotts is an indoor steel roller coaster at ...",
    "question": "is harry potter and the escape from gringotts a roller coaster ride",
    "choices":  ["True", "False"],
    "label":    0
}
```

The "label" field is the position of the correct answer in the "choices" list

BoolQ: possible prompt

Step 2: come up with an effective prompt. For BoolQ, it could be the following:

Prompt: `Given the passage {{passage}}, is the following statement, {{question}}, true or false?`

Follow the jinja template to create the prompt, put the JSON features inside curly brackets ({{...}}).

There can be multiple valid prompts for any given task.

Your job is to **provide between three and five different prompts for each task**. Motivate your design choices.

Categorical sentence-level task: ANLI

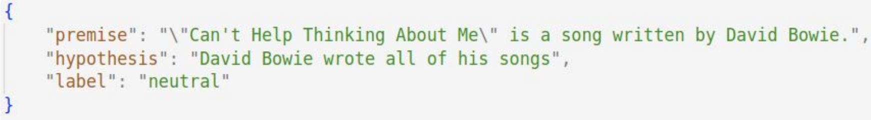
EXAMPLE

A categorical sentence-level task consists in assigning a category to a given input sentence

ANLI: Hugging Face link | The Adversarial Natural Language Inference (ANLI) is a large-scale NLI dataset

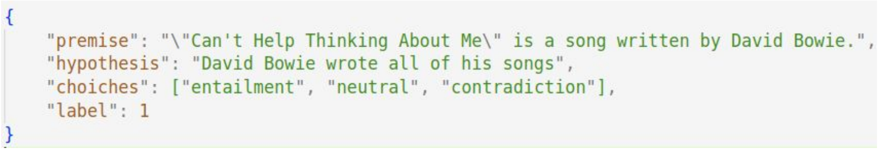
Step 1

From this…



```
{
    "premise": "\"Can't Help Thinking About Me\" is a song written by David Bowie.",
    "hypothesis": "David Bowie wrote all of his songs",
    "label": "neutral"
}
```

… to this



```
{
    "premise": "\"Can't Help Thinking About Me\" is a song written by David Bowie.",
    "hypothesis": "David Bowie wrote all of his songs",
    "choiches": ["entailment", "neutral", "contradiction"],
    "label": 1
}
```

The “label” field is the position of the correct answer in the “choices” list

Step 2

- Given a premise {{premise}} and a hypothesis {{hypothesis}}, determine whether the hypothesis logically follows from the premise (entailment), contradicts the premise (contradiction), or is unrelated to the premise (neutral).
- ...

This example is in English, but you will use Italian in all fields (also choices)!

Word-level classification task

EXAMPLE

In word-level classification tasks you have to classify each word of an input sentence

Those particular cases **have to be reframed as multi-choice classification tasks**

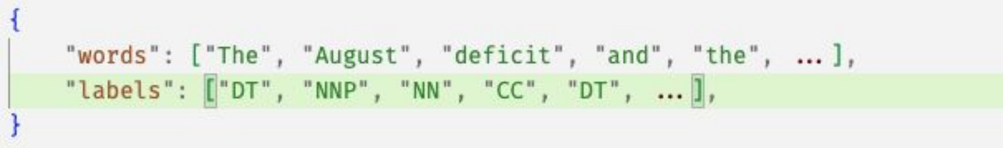
For example, in the Part-of-Speech (POS) tagging task, each word of the input sentence is labeled with a specific tag, such as NN (noun), ADV (adverb), and so on.

You have to reframe the task by **creating a new sample for each word**.

Word-level classification task: POS tagging

Practical example, POS tagging: Hugging Face link

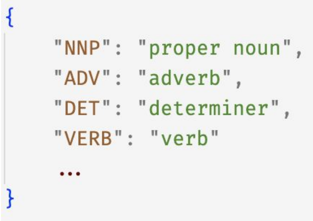
For each token in the input sentence, you are asked to select 4 different tags to choose from (one correct and three incorrect)



```
{
    "words": ["The", "August", "deficit", "and", "the", ... ],
    "labels": ["DT", "NNP", "NN", "CC", "DT", ... ],
}
```

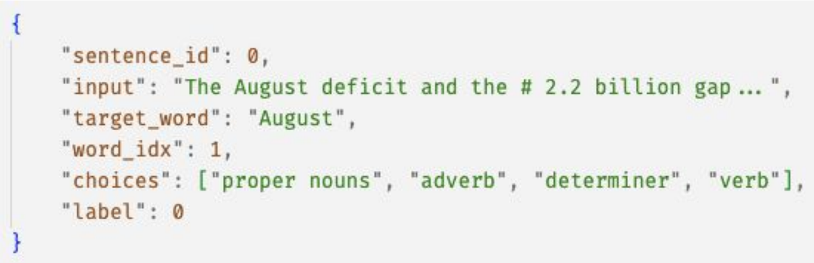
From this…

Categories mapping



```
{
    "NNP": "proper noun",
    "ADV": "adverb",
    "DET": "determiner",
    "VERB": "verb"
    ...
}
```

… to this



```
{
    "sentence_id": 0,
    "input": "The August deficit and the # 2.2 billion gap ... ",
    "target_word": "August",
    "word_idx": 1,
    "choices": ["proper nouns", "adverb", "determiner", "verb"],
    "label": 0
}
```

This example is in English, but you will use Italian in all fields (also `choices`)!

The “label” field is the position of the correct answer in the “choices” list

The “index_word” indicates the position of word in the input sentence, necessary to avoid ambiguities

POS Tagging: Possible Prompt

- `input_sentence`: *“I like to read books”*
- `target_word`: *“books”*
- `word_idx`: *4*

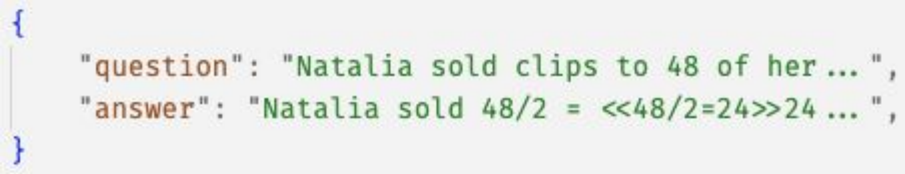
Possible prompt: `“In the sentence {{input_sentence}}, which is the part of speech tag of the word {{target_word}}?”`

SAPIENZA NLP

Practical example: GSM8k – Generative task

EXAMPLE

GSM8K: Huggingface link | (Grade School Math 8K) is a dataset of high quality linguistically diverse grade school math word problems

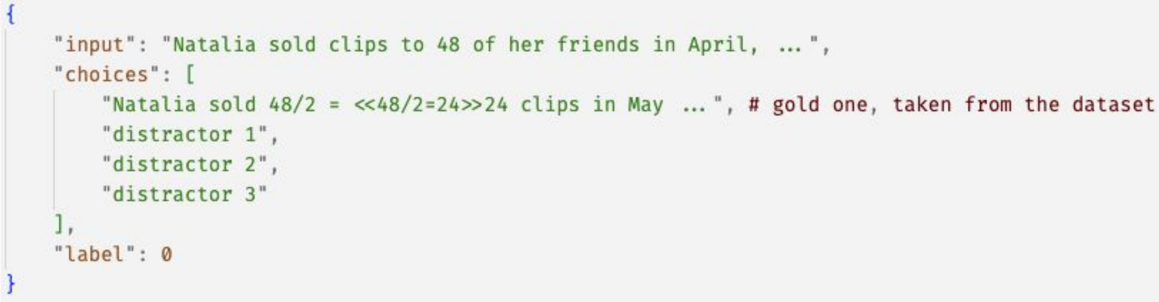


```
{
    "question": "Natalia sold clips to 48 of her ... ",
    "answer": "Natalia sold 48/2 = <<48/2=24>>24 ... ",
}
```

From this…

This example is in English, but you will use Italian in all fields (also `choices`)!

… to this



```
{
    "input": "Natalia sold clips to 48 of her friends in April, ... ",
    "choices": [
        "Natalia sold 48/2 = <<48/2=24>>24 clips in May ... ", # gold one, taken from the dataset
        "distractor 1",
        "distractor 2",
        "distractor 3"
    ],
    "label": 0
}
```

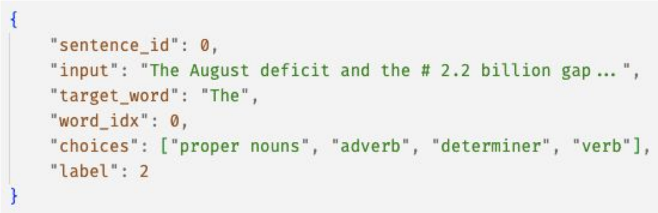
The "label" field is the position of the correct answer in the "choices" list

SAPIENZA NLP

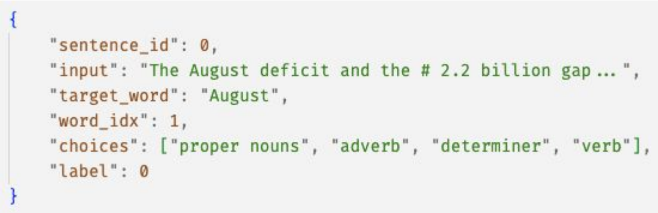
Sentence Identifier (only for word-level tasks)

When you create different samples from the same sentence, as in word-level classification, we ask you to add a sentence identifier (through a **`sentence_id`** field) for each sample created from the same sentence.

In the previous example you will have the same **`sentence_id`** for each sample derived from the same starting sentence



```
{
    "sentence_id": 0,
    "input": "The August deficit and the # 2.2 billion gap ... ",
    "target_word": "The",
    "word_idx": 0,
    "choices": ["proper nouns", "adverb", "determiner", "verb"],
    "label": 2
}
```



```
{
    "sentence_id": 0,
    "input": "The August deficit and the # 2.2 billion gap ... ",
    "target_word": "August",
    "word_idx": 1,
    "choices": ["proper nouns", "adverb", "determiner", "verb"],
    "label": 0
}
```

Word level tasks: same word, different labels

Important!!! There could be cases where you can encounter *ambiguous* words. For example, in a **Named Entity Recognition**, task you can have:

"La Roma ha giocato allo stadio Olimpico a Roma"

where the first occurrence of "Roma" is labelled as **ORGANIZATION** and in the second case as **LOCATION**.

You must remove all such sentences from the dataset.

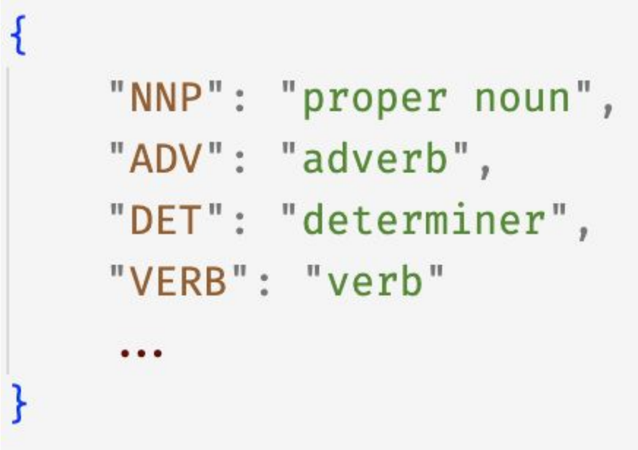
As a general rule, if you encounter an ambiguous sample where the same word appears **with different labels,** you have to **remove the entire sentence** during the dataset creation process.

Category Mapping

In some cases the labels of a classification task are expressed as acronyms or simply are not expressed in a natural language string.

We ask you to create a mapping and **use always a natural language label** to identify those categories in the generated dataset.

For example in POS tagging →
(remember to use Italian!)



```
{
    "NNP": "proper noun",
    "ADV": "adverb",
    "DET": "determiner",
    "VERB": "verb"
    ...
}
```

Other kinds of tasks

- EVALITA and SemEval provide different kinds of task, some of them not following the categorizations that we showed up to now

- As a general guideline, we ask you to try to **reframe each task as multiple-choice QA**

Generation of Distractors

- There are tasks where the set of possible answers is huge or undefined
- These datasets can still be cast to a multi-class classification task by selecting 3 *distractor* answers, i.e.: incorrect answers that can be confused with the correct one
- **You have to motivate and explain your choice of distractors**, we will reward strategies that generate challenging distractors with up to +3 points
- **There is always a simple way to generate distractors**, for example by random sampling from the space of all possible answers. Strategies like this **will NOT be awarded extra points**

Here's an example...

Generation of Distractors

EXAMPLE

In a Question-Answering task

- **Question:** “Who was the Mexican leader who was supported by the US during Mexican civil war?”
  - **Correct Answer:** “Benito Juárez”
  - **Distractor 1:** “Donald Trump”
  - **Distractor 2:** “Pablo Picasso”
  - **Distractor 3:** “Albert Einstein”

In this case the distractors are incorrect person names, so we have maintained the same “category” of the correct answer. Try to do the same, motivating your choice, and keep some relation between the correct answer and the distractors.

Up to 4 answers

The set of possible answers/categories for a task could be more than 4 (e.g., in POS tagging there are ca. 27 tags).

You have to make sure that we always have at most 4 different possible answers, in Italian.

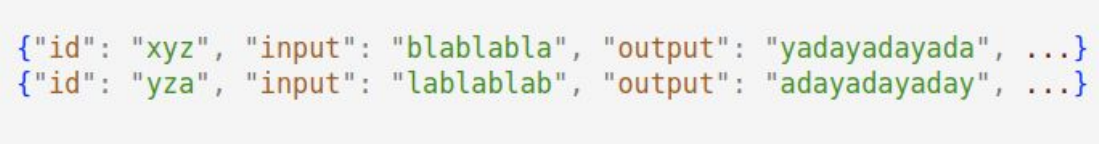
So, for each sample of a dataset, there will be one correct answer and up to three non-correct possibilities.

Dataset Format

You have to format your data using JSON Lines standard.

JSONL is very similar to JSON standard, but in this case *each line is a json object.*

Here is an example:



```
{"id": "xyz", "input": "blablabla", "output": "yadayadayada", ...}
{"id": "yza", "input": "lablablab", "output": "adayadayaday", ...}
```

Remember to also add an ID (e.g., <dataset>_<task>_xxx) to each json object!

Note on the prompt definition

- There must be exactly one prompt file for each task/sub-task.
- You have to define **at least 3 and at most 5** prompts for each task, **in Italian**.
- Hence, the prompt file will be composed by at most 5 JSON objects (i.e., 5 prompts), **one JSON object per line**.

… The EVALITA|SemEval format is not standard

- So far, we showed you source samples in clean json format, but during this first assignment you will encounter datasets in several format (e.g., csv, tsv, conll)
  - Some of the dataset may also violate the standard, e.g., by putting too many commas in a csv file – that’s data in the wild…
- It is up to you to understand the source format and generate a JSONL dataset, taking into account the particularities of each task.
  - Before contacting the TAs because something is not clear to you, do check the original guidelines for the task (you can find them on the task website/GitHub)
  - Before contacting the TAs for “broken datasets”, please try to solve the problem yourselves (e.g., if the pandas package can’t read that file, maybe you can write a simple scripts that does so)

Recommendations

1) Ensure that your strings are valid (e.g., using escaping sequences)
2) Check that your JSONL is valid (use the official python library)
3) Distractors have
   a) **to change**: don’t repeat the same three incorrect options
   b) **to be challenging**: in the POS tagging case, if a word has more than one possible tag in different contexts, it would be optimal to include them as distractors (e.g., for ‘run’ I want all its options like verb and noun, for ‘well’ I want noun, verb, adverb, adjective)
4) You should NEVER use as answers category labels that are cryptic or non-natural language (or even in English)
   a) E.g., in the POS tagging case, instead of DT, NN, NNP, etc., use “Determiner”, “Common Noun”, and so on (but remember to be consistent!)

Submission and Deadline

SAPIENZA NLP

Grading

- Each student will be evaluated on their individually assigned datasets.
- We will assign up to 15 points for each dataset, so including extras you can obtain up to 33/30 points:
- We will follow this breakdown:
  - Task comprehension (the student understood the task and how it works): ¼
  - Output file correctness (the files have been generated and are correct): ¼
  - Report validity: adequate explanation of the work done: ¼
  - Code provided and working: ¼
- **Extra points (as a group):**
  - We will assign up to +3 points based on the soundness of the proposed distractor generation approach
  - We expect this work to be carried out **JOINTLY** and defended by all group members
  - The extra points will be assigned to all members of the group

Extra points

- We will also assign **1 point** to each "best" dataset per task (if at least one is of adequate quality)
  - All the datasets for the same task will compete
  - Assignment based on performance evaluation carried out by Sapienza NLP
- The point will be assigned to the student who parsed the dataset (also in case of distractor datasets)
- So overall one student can be get up to 15+15 (2 datasets)+3 (group's smart distractor strategy)+1+1 (two best datasets for the corresponding tasks!)

SAPIENZA NLP

Assigned datasets

- Each student is assigned **two datasets** that you have to convert to JSONL format, see the assignment here

- There are **five tasks** where you are asked to select distractors:
    - Task 14 (Ghigliottin-AI)
    - Task 26 (Hypernym Discovery)
    - Task 12 (Tagit)
    - Task 20 (Itamoji)
    - Task 23 (PosTwita)

- Based on the datasets we assigned, there will be two cases (see next slide)

Assigned datasets

A. You may already be assigned a “distractor” dataset; in that case, you have to carry out the task in a simple manner (i.e., you can use straightforward approaches)
Then, only if you want to get the extra points, you will develop (**as a group**) a more effective selection approach

B. If you do not have a distractor dataset, you may swap one of your assigned dataset with it (you can find them in column F, “EXTRA”), and then, refer to case A
You may choose freely what already assigned dataset you swap

Report Delivery

- You will write an **individual report for each dataset** (up to 1 page, excluding tables, figures, and references). See next slide for more details.
  - You have 2 datasets, so you can deliver up to 2 pages excluding tables, figures, references, etc.
- A **group report** (up to 1 and a half page, excluding tables, figures, and references), in which you provide a detailed **description** of the proposed distractor generation approach and the **motivation** behind your choices.
- Delivery deadline: **23:59ish Tuesday April 9th Italian time (CET)**
- **Late submission penalty:** we will deduct 1 point for each day after the deadline, up to a maximum of -5 points.
- **No submissions are accepted beyond 23:59 Sunday April 14th (CET)**

What to Report

For **each given dataset you have to report <u>in short</u>**:

1. The description of the dataset (brief)
2. The format of the given dataset, give to us a sample of the input format.
3. Methodology used to reframe the dataset
4. Methodology and rationale behind the distractors generations (**this only applies to group reports, i.e., distractor reports**)
5. The list of suitable prompts
6. If needed by the task, a mapping for categorical natural language labels (see POS tagging slide)
7. The instructions to run your code and any other information you think it will be useful for us

Report Format

- Use **Latex** to write the report. If you don’t know how to use it, it is a good time to start learning :)
- I suggest you to write your report using **Overleaf** for the easiest possible experience
- **Use the current ACL template** (available as Overleaf template or in GitHub)
- Each report should consist of **up to 1 page**, excluding tables, figures and references (which should be at the end of the report). For distractor datasets, you are allowed to use **an extra column in a new page.**

What to deliver

- You have to create a data file **for each split** present in the dataset.
  - *If only train and test are provided, deliver only train and test splits.*
- You also have to create a prompt file, called *prompts.jsonl*, one for each task.
- **Important!!!** you have to submit all the code (for each task)
- **<u>It is mandatory</u>** to follow the example in the next slide

What to deliver - Example of folder structure

- Example of folder structure:

HM1_A-<student_id>/

- **<group-id>_<distractor_task_name>/**
  - train.jsonl
  - val.jsonl
  - test.jsonl
  - Report.pdf *# this identical for all the members of a specific group. All students must deliver a copy*
  - scripts.py
- **<task_name-1>/**
  - scripts.py
  - <subtask_name>-train.jsonl
  - <subtask_name>-valid.jsonl
  - <subtask_name>-test.jsonl
  - <task_name-1>_prompts[_subtask_name].jsonl
  - <task_name-1>_Report.pdf *# this instead is specific of a student.*
- …

Sapienza NLP

READMEs

Each dataset assigned to you will contain a `README.md` file.

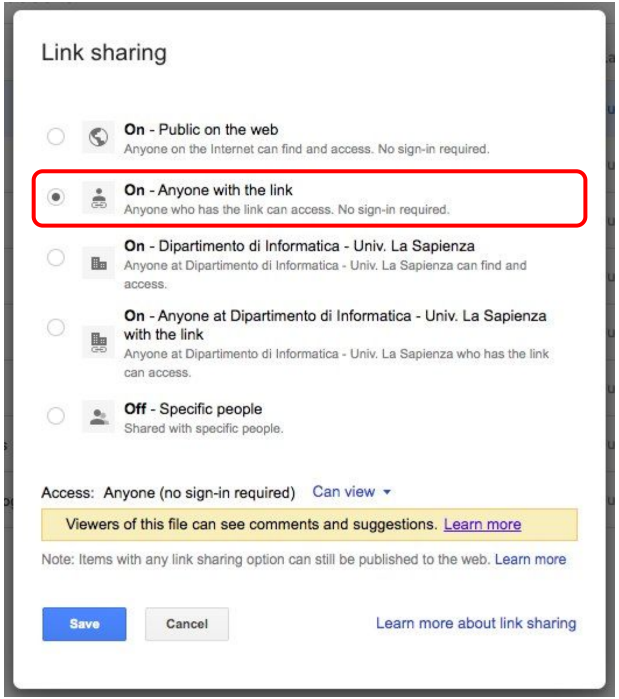
It contains:

- a simple description of the task,
- where to get the data from,
- some instructions regarding what we expect as output,
- the dataset licence.

These READMEs should be quite comprehensive, feel free to contacts the TAs if you have any doubts (but first do check the original website of the task).

Folder containing all datasets READMEs

Submission Instructions



Link sharing

○ **On** - Public on the web
Anyone on the Internet can find and access. No sign-in required.

● **On** - Anyone with the link
Anyone who has the link can access. No sign-in required.

○ **On** - Dipartimento di Informatica - Univ. La Sapienza
Anyone at Dipartimento di Informatica - Univ. La Sapienza can find and access.

○ **On** - Anyone at Dipartimento di Informatica - Univ. La Sapienza with the link
Anyone at Dipartimento di Informatica - Univ. La Sapienza who has the link can access.

○ **Off** - Specific people
Shared with specific people.

Access: Anyone (no sign-in required) Can view ▾

Viewers of this file can see comments and suggestions. Learn more

Note: Items with any link sharing option can still be published to the web. Learn more

- 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!

- 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.

Plagiarism

Collaboration among groups is **not** allowed.

We will check for plagiarism both manually and automatically.

It is **not allowed** to:

- Copy from other students.
- Share your code across groups.
- Use ChatGPT or similar systems **for report writing**

Projects under any of the above conditions will be **desk-rejected**

Any doubts? (hopefully not)

Use the Google group if you have any questions. Only **after you cannot solve your issues in the group**, write to all the TAs:

- Luca Moroni: moroni@diag.uniroma1.it
- Luca Gioffrè: gioffre@diag.uniroma1.it
- Lu Xu: xu@diag.uniroma1.it
- Alessandro Scirè: scire@diag.uniroma1.it
- Tommaso Bonomo: bonomo@diag.uniroma1.it

Start the mail subjects with “**[NLP 2024 HW1.A]**”

Sapienza NLP

Good Luck! (again)

Sapienza NLP