

# NLP 2024

## Homework 3 instructions

---

### Fit and Evaluate Generative Models

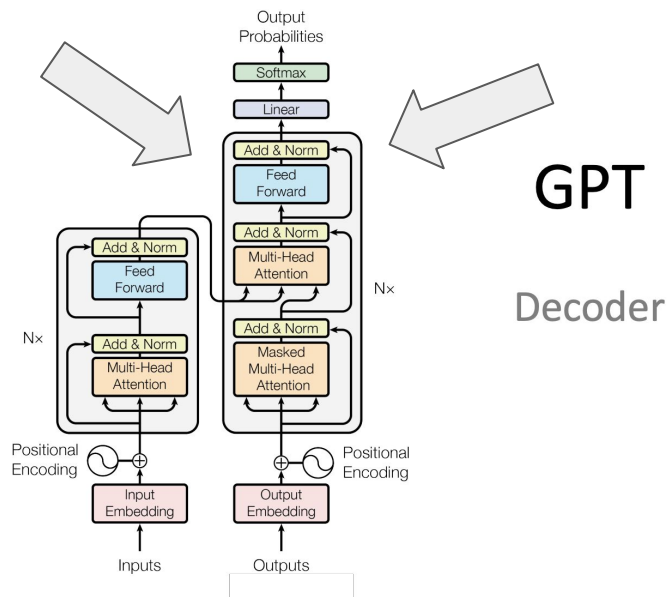
Slides provided by:

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



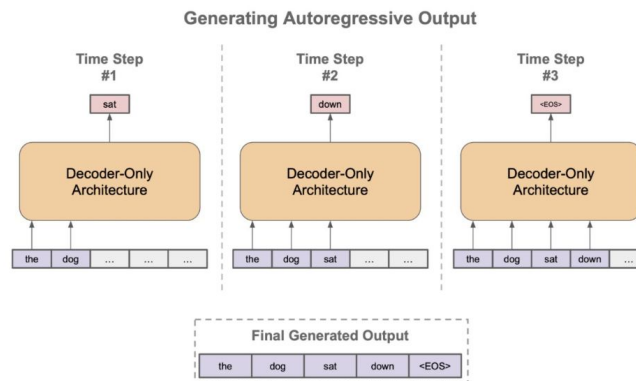
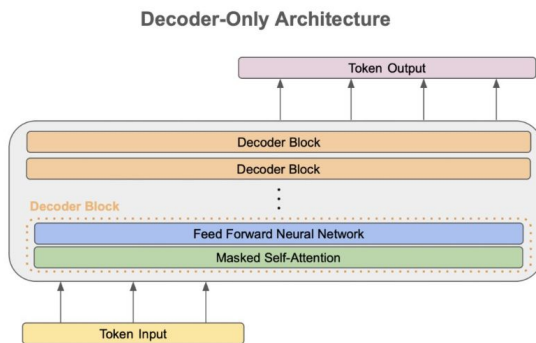
# Generative Models

In this Homework you will be introduced to the Natural Language Generation (NLG) world. We will face decoder models for the first time ...



# How to train Generative Models

During training, generative models take as input a sequence of tokens (a sentence) and return as output the same sequence shifted by 1 (next token prediction)



# Problem of teacher forcing

The setting differs a lot between **training time** and **test time**.

**Training time:** due to optimization reasons, the output sequence is predicted in a single shot, applying a **teacher forcing** setting, where gold target token is passed as the next input to the decoder.

**Test time:** at test time we don't have gold target tokens so the input of next token is the previous predicted token.

Evaluating the tokens decoded at training time are not the optimal way to test Generative Models.

# Summarization problems and possible metrics

During the lecture about Text Summarization, you already analyzed the problems of evaluation metrics used to test summarizer models ->

<https://classroom.google.com/w/NjY1NTk0NzUwNjY5/t/all>

Standard metrics (e.g. Blue, Rouge1, RougeL, ...) cannot take into account the real meaning of the summary generated by a model, but look only at sequence overlapping...

# News Summarization

In this homework you will use a Newspaper summarization dataset: **CC-News**:

[https://huggingface.co/datasets/vblagoje/cc\\_news](https://huggingface.co/datasets/vblagoje/cc_news)

Input schema:

```
{  
    "Text": the actual article text in raw form,  
    "Description": description or a summary of the article  
}
```

# Minerva-350M as base model

You will work with our Minerva model!!!

Specifically you will use the small one:

[Minerva-350M-base-v1.0](#)



**Minerva-350M-base-v1.0**

This compact model is **fast and agile**, making it ideal for applications requiring quick responses and lower computational resources. With dual-language training in Italian and English, it's perfectly suited for fine-tuning tasks in specialized domains where tailored responses are crucial.

# SFT Training Colab Notebook

Here you can find the colab that you can use to train for some steps a generative decoder-only model:

<https://colab.research.google.com/drive/1Q3Jav6HeNRWVw3nkvW6y0AqoZgUAEA>  
[SX](#)



# Evaluation Colab Notebook [SKELETON]

Here you can find the colab that you can use to test a generative model:

There you have to implement the pipeline to generate text with your tuned minerva model, following this documentation: [LINK](#)

[https://colab.research.google.com/drive/1kBr\\_tFQtCUPz9AoqoaI8UJOx-9pQLFaXI#scrollTo=a5eoG3QjhoCi](https://colab.research.google.com/drive/1kBr_tFQtCUPz9AoqoaI8UJOx-9pQLFaXI#scrollTo=a5eoG3QjhoCi)

# What we expect from you

## Mandatory (up to 27/30)

1. Train a **Minerva-350M** model to solve a summarization task on the provided dataset, getting out two checkpoints, the number of steps between each checkpoint is up to you, Colab GPU is limited... (we suggest to save 600 and 1200, with the configuration given to you is about 1h of computation)
2. Evaluate with Rouge metrics the two checkpoints (over a subset of test set)

## Qualitative Analysis (+3 points)

3. From the test set, get out the prediction of three samples (the same between the two checkpoints), do a qualitative analysis between the checkpoints, reporting your chosen metrics. Evaluation criteria:
  - a. **Factual Consistency:** whether the generated summary is factually-correct wrt the input document, i.e., does not contain hallucinations.
  - b. **Fluency:** Whether the output is readable and grammatically correct,
  - c. **Coherence:** The generated summary follows a clear logical flow, without interruptions in the narrative or jumps.
  - d. **Relevance:** The fact featured in the output summary are salient (important) wrt the input document.

**Extra (+3 points):** Try at least other two different evaluation metrics over the validation set, explaining them properly, comparing them w.r.t. Rouge

# Report Delivery

- You will write an **individual report**, up to **2 pages long** excluding tables, figures and references.
- Delivery deadline: 23:59ish June 18th 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 June 23rd (CET)

# Report format

- Use **Latex** to write the report, we suggest [Overleaf](#).
- Use the current **ACL template** (available as [Overleaf](#) template or in [GitHub](#))
  - We penalized some submissions in HW1 and HW2 for not using this template...
- Each report should consist of **up to 2 pages**; tables, figures and references are not counted in this limit, but they **have to be at the end** of the report
  - Again, we penalized some submissions in HW1 and HW2 for putting figures and tables between the text, resulting in too many pages

# Report format

- We are interested in your reasonings and why you did X instead of Y – this must be your priority when writing the report
  - Any unsupported claim (e.g., saying that the dataset is unbalanced without providing stats) is considered an **error**
  - Technical details are important, but less important than your reasoning, which is the **priority**
- **Remember:** a well-written report with okay results will give you more points than a poorly-written report with stunning performances

# Report content

In the report you have to specify:

- **Number of steps** used to tune the model.
- Brief description of the **created code to evaluate** the models.
- Average of the **chosen metrics** over a reasonable amount of instances of the test set for two **selected checkpoints**.
- **For each model checkpoint**
  - Of the 3 test samples highlight: hallucinations, quality of the generated text, factual references on the input, repetitions and never ending generation.
  - Of the 3 test samples report the chosen metrics.

# What to deliver...

You must also deliver the code created to evaluate the model, following the second notebook that we shared with you, and two jsonl files with the chosen test samples.

You have to deliver:

- Modified evaluation notebook
- **Jsonl with:** “text”, “gold\_summary”, “generated\_summary”
  - One file for each model checkpoint (so 2 files in total)
  - Each line is a json object (3 lines, one for each chosen sample). The samples have to be the same across the two files, only change the “generated summary”

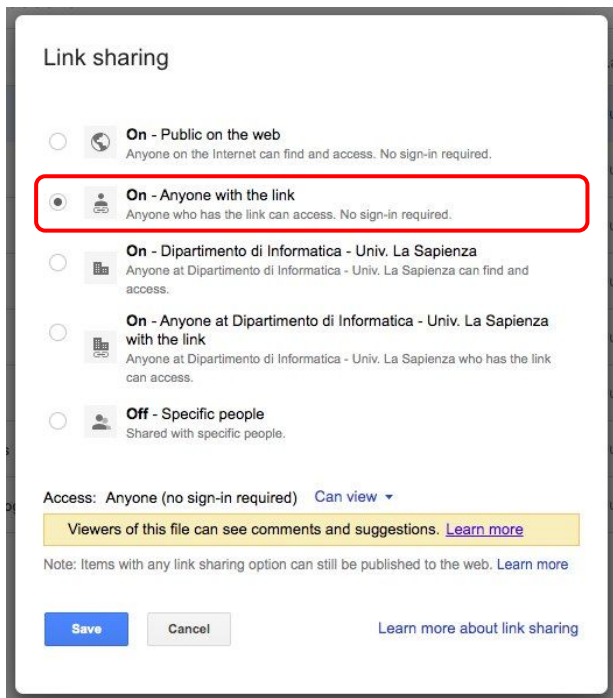
# What to deliver...

Folder schema of your submission:

- HW3\_<surname>-<university\_id>
  - Generative\_evaluate.ipynb
  - Report.pdf
  - Checkpoint\_1\_selected\_samples.jsonl
  - Checkpoint\_2\_selected\_samples.jsonl





# Submission Instructions





The screenshot shows the 'Link sharing' settings for a Google Drive file. The 'On - Anyone with the link' option is selected and highlighted with a red rectangle. Below it, the 'Access' is set to 'Anyone (no sign-in required)' with a 'Can view' dropdown. A yellow banner indicates that viewers can see comments and suggestions. At the bottom, there are 'Save' and 'Cancel' buttons, and a link to 'Learn more about link sharing'.


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](#)

[Save](#) [Cancel](#) [Learn more about link sharing](#)

- 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!
- As this homework is reserved to **non-attending** students, you should submit it through the [non-attending form](#).

# Plagiarism

We will check for plagiarism both manually and automatically.

It is **not allowed** to:

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

Projects violating any of the above conditions will be desk-rejected

# Any doubts?

Use the [Classroom group](#) if you have any questions. Only after you cannot solve your issues in the group, write in English to **ALL the TAs**, so to have a higher reply rate ;)

- Luca Moroni: [moroni@diag.uniroma1.it](mailto:moroni@diag.uniroma1.it)
  - Luca Giofrè: [gioffre@diag.uniroma1.it](mailto:gioffre@diag.uniroma1.it)
  - Lu Xu: [xu@diag.uniroma1.it](mailto:xu@diag.uniroma1.it)
  - Alessandro Scirè: [scire@diag.uniroma1.it](mailto:scire@diag.uniroma1.it)
- Tommaso Bonomo: [bonomo@diag.uniroma1.it](mailto:bonomo@diag.uniroma1.it)

