Language Models (LMs) have advanced a range of semantic tasks, and the recent prompt-based learning paradigm has enabled knowledge extraction from these large models. As mentioned in the seminal LAMA paper [1], LMs have many advantages over structured knowledge bases: they require no schema engineering, allow practitioners to query about an open class of relations and can be extended to more data. Knowledge bases consist of factual triples of the form (`subject_entity, relation, object_entity`). We can formulate a query or prompt as a "fill-in-the-blank" type cloze statement for a given `subject_entity` and `relation` of interest.

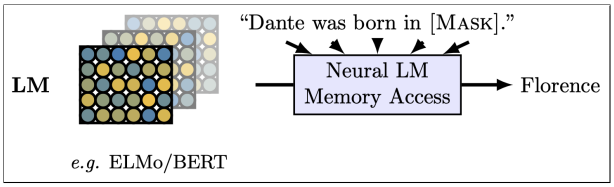

Figure 1: Here Dante is the `subject_entity` and born-in is the `relation`. Using a BERT type masked LM, one can obtain Florence as the `object_entity` in the [MASK] token position.

In this assignment, you will use existing pre-trained LMs to construct a knowledge base. The given dataset has three columns, the first two, `subject_entity` and `relation` are part of the input, the third, `object_entity` stores ground truth values. Your task is to design a solution that formulates an input prompt containing the `subject_entity` and `relation` and generates all possible `object_entities` by LM probing. For example, when Germany is the `subject_entity` and the `relation` is shares-border, the output should ideally contain nine triples where the `object_entities` are the names of countries with which Germany shares a land border.

```
Input: Germany shares a border with [MASK].
Output:
[
    ("Germany", shares-border, "France"),
    ("Germany", shares-border, "Denmark"),
    ("Germany", shares-border, "Poland"),
    ("Germany", shares-border, "Belgium"),
    ("Germany", shares-border, "Netherlands"),
    ("Germany", shares-border, "Luxembourg"),
    ("Germany", shares-border, "Switzerland"),
    ("Germany", shares-border, "Austria"),
    ("Germany", shares-border, "Czechia"), # Czech Republic is also possible
]
```

Dataset and relations. Download the dataset and supporting scripts from the GitHub repository[1]. There are a total of 5 relations: `CountryBordersWithCountry, RiverBasinsCountry, PersonLanguage, PersonProfession`, and `PersonInstrument`. For each relation, 100 unique subjects are provided in the train split and 50 in the test split.

Evaluation. The generated output on the given test dataset will be evaluated using precision, recall, and f1-score metrics, macro-averaged across all the subjects, and then macro-averaged across all relations. Check `evaluate.py` script for more details.

Baseline. A baseline solution is released which uses the BERT language model, and sample prompts like "Germany shares border with [MASK]", and selects object-entities predicted in the [MASK] position with greater than or equal to 0.5 likelihood as outputs. Follow the instructions in the `README.md` file to run the baseline script. The baseline achieves 8.4% precision, 5.61% recall and 6.32% f1-score macro-averaged scores.

[1] `https://github.com/snehasinghania/akbc-lab08/`

Approach. Possible approaches for building upon the given baseline script include:

1. *Prompt Engineering*: The output given by an LM is highly susceptible to the given input prompt. You can manually figure out better prompts using trial-and-error or refer to existing implementations in this space (e.g., AutoPrompt[2], LPAQA[3], etc.)

2. *Selection Criteria*: Other than picking the generated output with greater than 0.5 probability, better selection mechanisms could be used. A lower probability threshold could help improve the recall as more entities will get selected, but it might hurt precision. Similarly, a higher threshold could help with precision but might affect the overall recall. You can devise your selection algorithm that produces an optimal (possibly subject-specific) threshold.

3. *Model Selection*: You can choose any LM that can perform masked token prediction from the HuggingFace Library[4]. **NOTE**: Current research trend shows that larger LMs give better performance. Depending on the computational resources available to you, selecting larger models other than BERT or fine-tuning of a LM is also possible. Please make sure to include only the train dataset during the fine-tuning process. If required, you can split the train dataset into train and dev as well. Any submission using the information from test dataset will not be passed.

Submission

Your submission **should include** a short description explaining your method (maximum 1 page) in **MatriculationNumber_methodology.pdf** file. Please implement your algorithm in the `your_solution` function in the `solution.py` file. Your submission should only contain `solution.py` and other supporting files (e.g., Python files imported in solution.py).

Your submission **should not include** the train and test dataset folders or other provided files (e.g., `baseline.py`, README.md, etc.).

Please submit all necessary files, which are compressed into a zip file named:
Lab08_MatriculationNumber_Name.zip
to the email address: **akbc-assignments@mpi-inf.mpg.de** with title of the email: **[AKBC]Lab08_MatriculationNumber_Name**

Deadline: 23:59 21.06.2022 (Tuesday)

References

[1] F. Petroni, T. Rocktäschel, S. Riedel, P. Lewis, A. Bakhtin, Y. Wu, and A. Miller, "Language models as knowledge bases?" Association for Computational Linguistics, 2019, pp. 2463–2473.

[2] `https://github.com/ucinlp/autoprompt`
[3] `https://github.com/jzbjyb/LPAQA`
[4] `https://huggingface.co/models`