**Generating Scores From User Reviews**

Welcome to task #4, the last project of this year’s Introduction to Machine Learning course at ETH!

When customers shop online, they can leave reviews for the purchased product. In this task, given a dataset containing the review's title and its content, you will try to infer the score of the product, which is a continuous value between 0 and 10.

**DATA DESCRIPTION**

You can download the project material here:

(4.4 MB)

In the handout, you will find the following material:

* **train.csv** - a csv containing the columns "title", "sentence", and "score". The score is always within the range [0, 10].
* **test\_no\_score.csv** - csv containing the columns "title" and "sentence".
* **sample.txt** - a sample submission file.
* **template\_solution.py** - a template file that will guide you through the implementation of the solution.
* **template\_solution.ipynb** - the template solution in an interactive notebook form.

**template\_solution.py** provides a starting template structure for how you can solve the task, by filling in the TODOs in the skeleton code. It is not mandatory to use this solution template but it is recommended since it should make getting started on the task easier. The template solution uses the [PyTorch](https://pytorch.org/) deep learning framework.

**SUGGESTED APPROACH**

The recommended approach is to use the [transformers](https://huggingface.co/docs/transformers/index) library, which contains the necessary network architectures and pretrained transformers. It is recommended to look into the following transformers:

* [DistilBERT](https://huggingface.co/docs/transformers/en/model_doc/distilbert#transformers.DistilBertModel)
* [ALBERT](https://huggingface.co/docs/transformers/en/model_doc/albert#transformers.AlbertModel)
* [GPT-2](https://huggingface.co/docs/transformers/en/model_doc/gpt2#transformers.GPT2Model)

Note that while traditional approaches using pretrained transformers requires fine-tuning them for the desired task, it is **NOT** required to fine-tune the transformers to pass the hard baseline. Using frozen weights are sufficient to pass the hard baseline. To do so, one can run the transformer separately and save the results to a stand alone file. When needed, the results are loaded without additional computation. It is recommended to refer to previous tasks for examples on how this is done. Interested students with sufficient computational resources are encouraged to fine-tune the models. This best mimics real world use cases of pretrained transformers.

Due to the heavy computational cost associated with transformer inference, it is recommended to use a modern GPU for this task. Students are invited to use Google Colab or the ETH Euler cluster. If no GPU can be obtained, it is possible to pass the hard baseline using CPU-based computation on the ETH Euler cluster within a reasonable time limit.

A screenshot of a computer

Description automatically generated

Loss on public data 3.35