Assignment 5: Classify Math Publications

AI-2 Systems Project (Summer Semester 2024)

Jan Frederik Schaefer

Friedrich-Alexander-Universität Erlangen-Nürnberg, Department Informatik

Topic:	Natural language processing, artificial neural networks
Due on:	September 15, 2024
Version from:	June 28, 2024
Author:	Jan Frederik Schaefer

1 Task summary

Train a neural network to suggest classifications for mathematical publications based on their titles. This could, for example, be used in a web interface where authors submit publications. We suggest to follow a pytorch tutorial for text classification [TC], but you can also try out something else.

Didactic objectives

1. Learn the basics of a modern machine learning framework (pytorch),
2. get some hands-on experience with training a simple neural network for a natural language processing task.

Prerequisites and useful methods

1. Basics of neural networks,
2. basics of natural language processing.

2 Detailed task description

In recent years, natural language processing has advanced a lot through the use of sophisticated artificial neural networks (e.g. transformer models). These networks require a lot of computational power and are out of the scope of the AI lecture and the systems project. But simpler models can yield very good results as well! The goal of this assignment is to learn the basics of a modern machine learning framework, by solving a text classification problem.

You can base your implementation on a pytorch tutorial for text classification [TC] and use the matrix rooms if you have any questions about the tutorial or pytorch in general. It might also make sense to skim through the basic introduction tutorial for pytorch [PT]. It is possible to train a sufficiently good model within a few minutes (and without using a GPU).

2.1 The dataset

The dataset for this assignment is based on a publicly accessible dataset [ZBMd] from zbMATH [ZBM]. zbMATH collects abstracts and reviews of publications in the area of mathematics and its applications. For this assignment, we use the title of publications and their classifications, which are based on the Mathematics Subject Classification [MSC]. For simplicity, we only use the top-level classifications.

The training data consists of 1 000 000 titles, each of which has up to 5 classifications. They are stored in one file, where each line is a JSON object of the form

```
{"title": <TITLE>, "classifications": [<CLASSIFICATION 1>, ...]}
```

Here is an example entry:

```
{"title": "An extension of Chevalley's theorem to congruences modulo prime powers",
    "classifications": ["Number theory", "Group theory and generalizations"]}
```

There are also a test and a validation dataset that each list 10 000 publication titles (without classifications). The validation dataset is also available in the format of the training dataset, so you can use it for checking your solution.

The datasets are in the assignment repository [AR].

2.2 Evaluation

Your model should propose a set P_t of 5 classifications for every title t in the test dataset D. Your agent then gets the evaluation score

$$\frac{\sum_{t \in D} |P_t \cap C_t|}{\sum_{t \in D} |C_t|}$$

where C_t are the real classifications of t.

The proposed classifications should be stored in the same format as the training dataset:

```
{"title": <TITLE>, "classifications": [<CLASSIFICATION 1>, ...]}
```

You can use the validation dataset to evaluate your model (the assignment repository [AR] also has a script for that). Unless you use the validation dataset for training, the evaluation score on the validation dataset and the test dataset should be very similar.

3 What to submit

Your solution should be submitted to your repository for this assignment. It should contain:

1. all your code,
2. the proposed classifications for the test dataset (see Section 2.2),
3. a solution summary (see [SoS] for more details – it should describe the main ideas, not document the code).
4. a README.md file explaining
   i. dependencies (programming language, version, external libraries and how to get them),
   ii. how to run your code on different environments,
   iii. the repository structure,
   iv. anything else we should know.

Please **do not commit large files** (e.g. a large model file or a Python virtual environment) to the repository. Rather, tell us how to produce them in the README.md file.

4 Server

For the first time, we also use the AISysProj server for this assignment. For now, it is optional to use the server and the rating of your agent will be ignored while grading. However, you can use the server to test your agent and compare it to others. The assignment repository [AR] contains an example implementation for communicating with the server. Please let us know what you think about this (is it useful, what could be improved, ...).

5 Points

You can get up to 80 points for the evaluation score (see Section 2.2) of your model. This is **not based on the server**. Concretely, you will get the following points:

- 30 points if the evaluation score is ≥ 0.3
- 50 points if the evaluation score is ≥ 0.5

- 70 points if the evaluation score is ≥ 0.6
- 80 points if the evaluation score is ≥ 0.7

Note that it is possible to get a significantly better evaluation score than 0.7. Assuming you have at least a partial solution, you can additionally get up to 20 points for the quality of the submission (README, evaluation, ...). The maximum number of points is therefore 100. If the grading scheme does not seem to work well, we might adjust it later on (likely in your favor).

References

[AR] *Repository for Assignment 5: Classify Math Publications*. URL: https://gitlab.rrze.fau.de/wrv/AISysProj/ss24/a2.5-classify-math-publications/assignment.

[MSC] *Mathematics Subject Classification – MSC2020*. URL: https://zbmath.org/classification/ (visited on 02/04/2022).

[PT] *PyTorch – Learn the Basics*. URL: https://pytorch.org/tutorials/beginner/basics/intro.html (visited on 08/01/2022).

[SoS] *Solution Summary*. URL: https://gitlab.rrze.fau.de/wrv/AISysProj/admin/general/-/blob/main/solution-summary.md.

[TC] *Text classification with the torchtext library*. URL: https://pytorch.org/tutorials/beginner/text_sentiment_ngrams_tutorial.html (visited on 08/01/2022).

[ZBM] *zbMATH Open*. URL: https://zbmath.org (visited on 02/04/2022).

[ZBMd] *zbMATH Open Dataset*. URL: https://gl.kwarc.info/SIGMathLing/dataset-zbmath-open-2021 (visited on 02/04/2022).