Institut für Informatik

Reproducibility of SciBert

Jan-Niklas Weder

Modul : Data Mining

Contents

Introduction

- Als pretrained model hängt bert stark von dem korpus ab
- Genauso ist das vocabular sehr wichtig
- Andere arbeiten zeigten den einfluss eines erweiterten trainings/besser passenden korpus? referenz suchen
- $\Longrightarrow$ Roberta zum Beispiel zeigte, dass allein weiteres training die ergebnisse verbessern kann
- Wissenschaftlöiche texte unterschieden sich allgemein sehr stark von "normalen"
- $\Rightarrow$ Somit ist ein auf ähnliche weise trainiertes modell als bessere grundlagen für NLP aufgaben im wissenschaftlichen bereich sinnvoll
- Gab es in dieser form noch nicht
- $\Rightarrow$ besitzt daher potential (annahme das BERT gut sei)
- insbesondere als grundbaustein für unterschiedlichste aufgaben im !!! wiss. bereich
- Allgemein stellt sich das Problem von Datensätzen insbesondere da diese annotiert werden müssen (im wiss. bereich teuer da hochqualifizierte experten notwendig sind)

State of the art

2.1 BERT

- BERT as revolution
- pretrained-models
- usefull even without finetuning
- $\Rightarrow$ unexpected precision
- nowadays used for many different NLP tasks
- Architecture of BERT
  - explain corpora
  - explain vocabulary
  - explain tokenizer
- extensions of BERT like roBERTa

2.2 BioBERT

BioBERT is one of the extensions of BERT which were created because BERT by itself did not yield the desired results in the biomedical landscape. This observation was oftentimes explained with the differences in word distributions between a general domain such as Wikipedia on which BERT was trained and the highly specialized words which are oftentimes only or with this meaning only used in the corresponding domain. This difference in the underlying corpora not only produces differences in the architecture of the model itself but implies a possible need for adjustments to the used vocabulary and the tokenizer. [3]

BioBERT itself is a further trained version of BERT. This means the original BERT model was used as basis and further trained on either PubMed abstracts, PubMed central full-text articles or on both. Therfore the authers chose to keep the original BERT vocabulary to be able to use the pretrained version of BERT as basis. This had the advantage that the original model only needed to be further trained on the new corpus and the needed training time could be reduced.

2.3 AOG-BERT

- als kontrahend zu SciBERT [4]

2.4 Datasets

- zum Beispiel NCBI-Disease (versuch einen goldstandard für corpora zu erstellen)
- sehr günstig um darauf entsprechende modelle zu trainieren [2]
- SciERC /sciie im repo [5]

Due to the availability of the Datasets used by the original Authors. We will use their prepared Datasets, which are already prepared so that it is easier to use for training of neural nets and still only vary slightly from the original Datasets. The Datasets which we will use are directly retrieved from the SciBERT GitHub page and made available through the DataDeps package which provides an easy way to retrieve data that may or may not be locally available.

If it is not already stored locally it will be cached in the local Julia path and inside Julia, DataDeps provides the corresponding paths to the Data and retrieves it from the defined Source if needed. Furthermore, a hash can be defined as well to ensure that the provided data is identical to the expected one.[7] In the following paragraph, we will take a closer look at the original data and the individual changes that have been made to use those Datasets for the training process.

Chemprot

Chemprot is in a JSON lines file format provided. More precisely every line consists of a text and the corresponding label. A field for metadata exists as well but is most of the time not used. In its original format, the Chemprot corpus consists of a develop, test, and train Set of which the develop, test, and train folder correspond to the identically named files inside the chemprot folder provided on the GitHub site of sciBERT. The difference arises from a database-like structure in which the chemprot corpus is originally provided, in contrast to those subdivided information sets where for example the text itself is in another file than the positions and annotations. Those divided pieces of information were combined and are provided in a single file in the already mentioned format. [1, 6]

SciBert

3.1 Corpus

Comparison

3.2 Vocabulary

BaseVocab vs. SCIVocab
! $\approx 42\%$ overlap

SentencePiece

Experiments

- kurze einführung in die test fälle mit einer erklärung, was f1 scores sind
- Alles NLP Aufgaben bei denen Bert "überraschend" gut abschneided
- Jetzt mit der erweiterung zu scibert erneut betrachtet
- Einföuss des vocabulars und des corpus genau gegenübergestellt
- all got dropout of 0.1
- loss cross entropy
- optemizer adam
- finetuning for 2 to 5 epochs

In the following, we will take a closer look at how the already conceptually described tasks are implemented. Since some tasks utilize the same architecture for the model more precisely the identical architecture for the last layer, we will first explore the classification tasks and then the labeling tasks in more detail. So the order in which the tasks will follow stays the same as earlier.

Nevertheless, both task types will use a dropout of 0.1, cross-entropy for the loss, and Adam as the optimizer, all of this follows the parameter given in the SciBERT paper.

4.1 Classification tasks

Both classification tasks will utilize the same architectural structure. This means that the final BERT or SciBERT layer will be followed by a dense or fully connected linear layer. This dense layer then will act as the linear classification layer described in the original paper.

Text Classification (CLS)

Relation Classification (REL)

4.2 Sequenz labeling tasks

Named Entity Recognition (NER)

Pretrained model -¿ linear classification layer with softmax output

PICO Extraction (PICO)

4.3 DEP

dependency tag and arc embedding of size 100 and biaffine matrix attention

4.4 Finetuning

4.5 Frozen embeddings

4.6 Influences of different platforms

This section will take a short look at the usability of different hardware platforms for the creation of trans-

former models and in the training or testing of those. More precisely we will compare the google-colab environment with an Nvidia GPU and an AMD GPU. Due to the randomness of the allocation of hardware on the google-colab site, I cannot further define the GPU that was used on this platform. The Nvidia GPU that was used is a GeForce 940MX with approximate 2GB VRAM, the AMD GPU on the other side was an RX580 with approximate 8GB of VRAM. At this point, I will shortly describe how far the ROCM stack of AMD is usable because surprisingly I was able to define the model and make predictions with it in a newly created state. Sadly due to the instability in the ROCM stack, the Linux kernel wasn't capable of using the GPU anymore after an update which probably broke some intern dependencies on which the kernel and the ROCM driver relay and therefore the video output of the computer was not usable anymore. Even though this shows that in fact, an AMD GPU is capable of running the Transformer package and at least load a defined model. Even though I cannot disclose whether the model could be trained or in any other way further used. This fact in itself is surprising since AMD itself describes the support status of the RX580 as it "may or may not work" and Julia describes the support of AMD GPUs as level 3 which corresponds to the lowest level of support. (verweis zu der aussage)[belege und verweis zu ROCM]
Still, I would discourage anyone from installing the ROCM stack on any productive system since it is still unstable and therfore I would recommend experimenting only in some form of a virtualized environment. Of course, this warning only applies to systems that rely on a working video output.
Due to the failure with the ROCM stack, the following part will only consider the MX940 and the google-colab environment.

Discussion

Further development

Appendix

Data availability

Code availability

Bibliography

[1] Iz Beltagy, Kyle Lo, and Arman Cohan. "SciBERT: A Pretrained Language Model for Scientific Text". In: *EMNLP 2019* (Mar. 26, 2019). arXiv: 1903.10676 [cs.CL].

[2] Rezarta Islamaj Doğan, Robert Leaman, and Zhiyong Lu. "NCBI disease corpus: A resource for disease name recognition and concept normalization". In: *Journal of Biomedical Informatics* 47 (Feb. 2014), pp. 1–10. DOI: 10.1016/j.jbi.2013.12.006.

[3] Jinhyuk Lee et al. "BioBERT: a pre-trained biomedical language representation model for biomedical text mining". In: *Bioinformatics* (Sept. 2019). Ed. by Jonathan Wren. DOI: 10.1093/bioinformatics/btz682.

[4] Xiao Liu et al. "OAG-BERT: Pre-train Heterogeneous Entity-augmented Academic Language Model". In: (Mar. 3, 2021). arXiv: 2103.02410 [cs.CL].

[5] Yi Luan et al. "Multi-Task Identification of Entities, Relations, and Coreferencefor Scientific Knowledge Graph Construction". In: *Proc. Conf. Empirical Methods Natural Language Process. (EMNLP)*. 2018.

[6] Qinghua Wang et al. "Overview of the interactive task in BioCreative V". In: *Database* 2016 (2016), baw119. DOI: 10.1093/database/baw119.

[7] Lyndon White et al. "DataDeps.jl: Repeatable Data Setup for Reproducible Data Science". In: *Journal of Open Research Software* 7 (2019). DOI: 10.5334/jors.244.