Dauphine | PSL
UNIVERSITÉ PARIS

Natural Language Processing

Medical Abbreviation Disambiguation

Nathan Bigaud, Germain Bregeon & Emma Covili

Project report

December 20, 2022

1 Overview

In this project, we investigate pre-training a BERT-like model on medical abbreviation disambiguation for use on medical text. For pre-training, we use the MeDAL dataset, and the accompanying paper Wen, Lu, and Reddy 2020. We then compare this custom pre-trained model to the standard model by training both on a classification task on the MIMIC-III dataset. We compare the results, and, using Kovaleva et al. 2019, we explore the differences between the two models - focusing on attention patterns and influence of pruning.

2 Approach and data preparation

2.1 Approach

Our initial goal was to simply train a model on medical text. As we researched, we broaden our scope a little by trying to investigate some specificities of a model dealing with technical vocabulary.

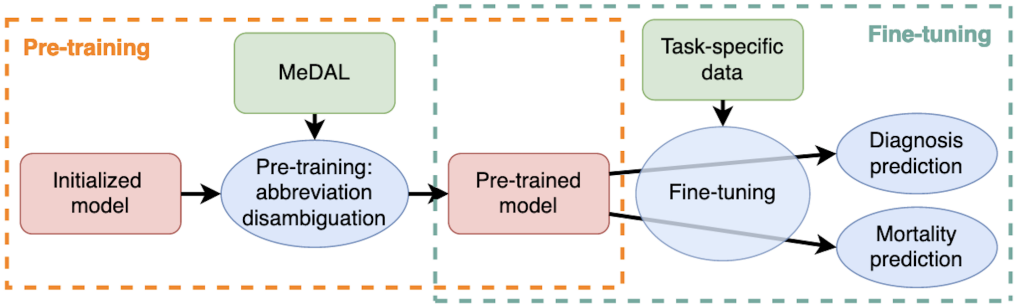


Figure 1: Overview of model training

To train our model, we used the paper *MeDAL: Medical Abbreviation Disambiguation Dataset for Natural Language Understanding Pretraining* by Wen, Lu, and Reddy 2020 and adapted elements of the accompanying github.

- **The article focuses on optimizing BERT-like models to medical text**. It uses models already pre-trained using MLM and NSP on large corpuses, and adds a new pre-training task using only medical text.
- **The medal dataset** is built by taking data from PubMed abstracts and replacing medical terms by their abbreviation. The pre-training task is then to correctly identify the proper term, being given only the abbreviation. For example, the abbreviation DHF can either mean dihydrofolate or diastolic heart failure or dengue hemorragic fever or dihydroxyfumarate, and we train a classifier to be able to decipher between the meaning based on context. In the following, we will refer to the model not trained on the MeDAL dataset as the *base model* and the pre-trained one as the *fine-tuned model*.

- **The model is then trained on two classification tasks** using the MIMIC-III dataset. We focus on the mortality prediction, a binary classification task.

We used the Hugging face implementation of the ELECTRA model from Google. The original paper uses LSTM, LSTM with self-attention and ELECTRA pre-trained transformer. We focus on ELECTRA, a small (14M parameters) BERT-like model optimized to be sample efficient, an important feature given our limited resources. Under the hood, ELECTRA models are trained to distinguish "real" input tokens vs "fake" input tokens generated by another neural network, similar to the discriminator of a GAN.

We then turned to probing the models using the methodologies and visualizations proposed by Kovaleva et al. 2019. In particular, the paper identifies five distinct types of attention patterns in tokenwise attention maps, and investigates in the general case the influence of pre-training on attention heads. Finally the authors explore the effects of disabling different heads in BERT and the resulting effects on task performance. We tried to explore those three aspects on our own model, focusing on the pre-training task rather than teh downstream one.

2.2 Data preparation

The MeDAL dataset consists of 14 million articles and on average 3 abbreviations per article. It is a large medical text dataset (14Go) curated to 4Go for abbreviation disambiguation, designed for natural language understanding pre-training in the medical domain. The dataset is divided into 3 csv files: train, test and valid. Each of them is a table consisting of three columns:

- `text`: The normalized content of an abstract as a string
- `location`: The location (index) of each abbreviation that was substituted as an integer
- `label`: The word at that was substituted at the given location as a string

The MIMIC-III dataset included deidentified health-related data associated with over forty thousand patients who stayed in critical care units of the Beth Israel Deaconess Medical Center between 2001 and 2012. We used only specific subsets of it, only processing the `ADMISSIONS, PATIENTS, DIAGNOSES_ICD,PROCEDURES_ICD`, and `NOTEEVENTS` files. We then build a simple dataset with two columns:

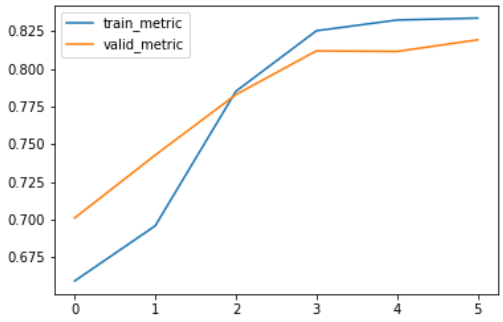
- `text`: The admission notes for a given patient
- `label`: Its survival, a binary variable

3 Model training

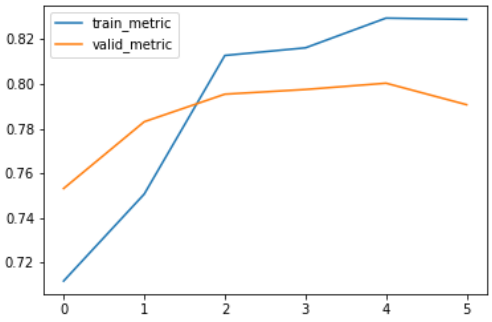
Our initial goal was first to reproduce in full the methodology shown in Wen, Lu, and Reddy 2020, in two main steps:

- **First, we focused on reproducing pre-training** on the task of abbreviation disambiguation in medical text. We faced a practical challenges, as the process was very resources intensive, and would have taken about 200h of continuous training. We trained for a single epoch, instead of the ten needed, simply to experiment with the process. For our experimentations, we then used the checkpoints provided by the authors.
- **Second, we implemented fine-tuning** for one of the two downstream tasks, using the MIMIC dataset. We detail below the results of this training.

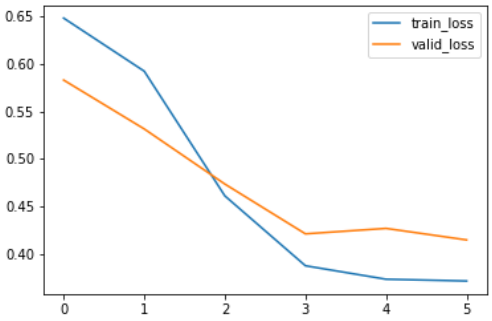
We trained the model on mortality prediction, mostly using the parameters suggested by the paper. We used a batch size of 8, 0.1 of dropout, Adam optimizer and learning schedule decreasing on plateau. One important difference is that we only trained for three epochs, each taking about 3h, instead of the ten used in the paper (2). Each epoch was divided in two batches in the graph below.



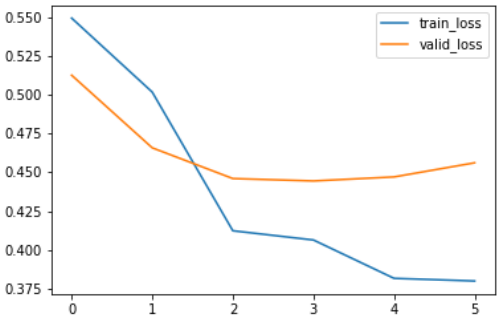
(a) Medal-trained accuracy



(b) Base model accuracy



(c) Medal-trained loss



(d) Base model loss

Figure 2: Learning curves

We observe a better generalization of the model pre-trained on medal. While the train metrics for both converge at comparable speed, the validation metric quickly drops for the base model. This is also observed if we validate our finding on test data, unseen by the model (3). Here we plot the standard deviation and average accuracy for the two models over five random subsamples of the test data. We observe a small advantage of the pre-trained model, of around 0.5 percentage point.

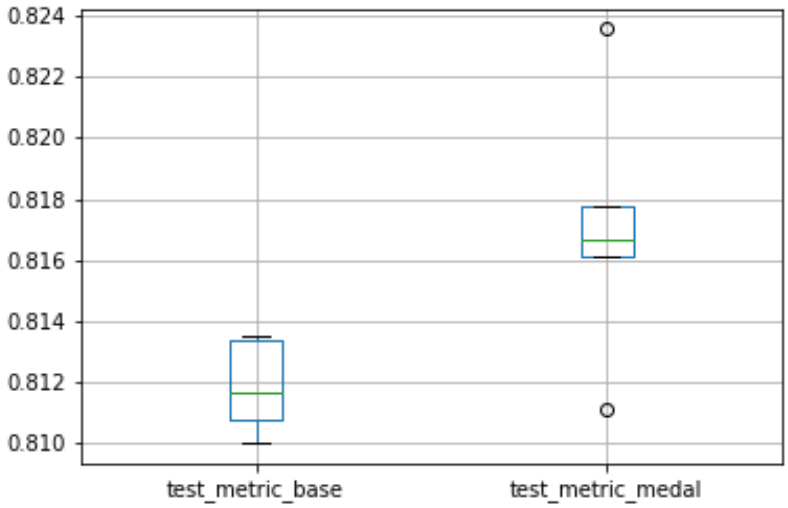


Figure 3: Overview of model accuracy

4 Models comparison

We now compare the two models using methodologies proposed by Kovaleva et al. 2019, focusing on attention patterns and influence of pruning. We first look at token-wise attention maps at the pre-training stage, then explore the attention pattern changes with the medal pre-training, and we finally look at models robustness to random pruning.

4.1 Token-wise attention maps

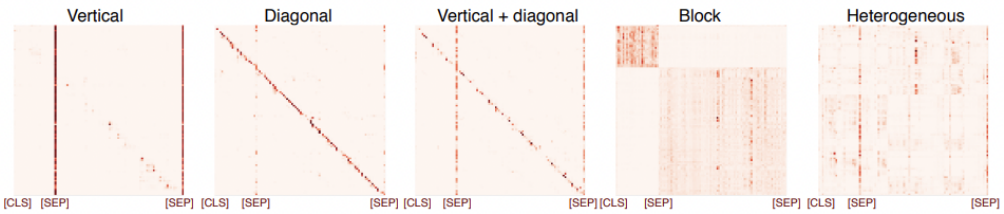
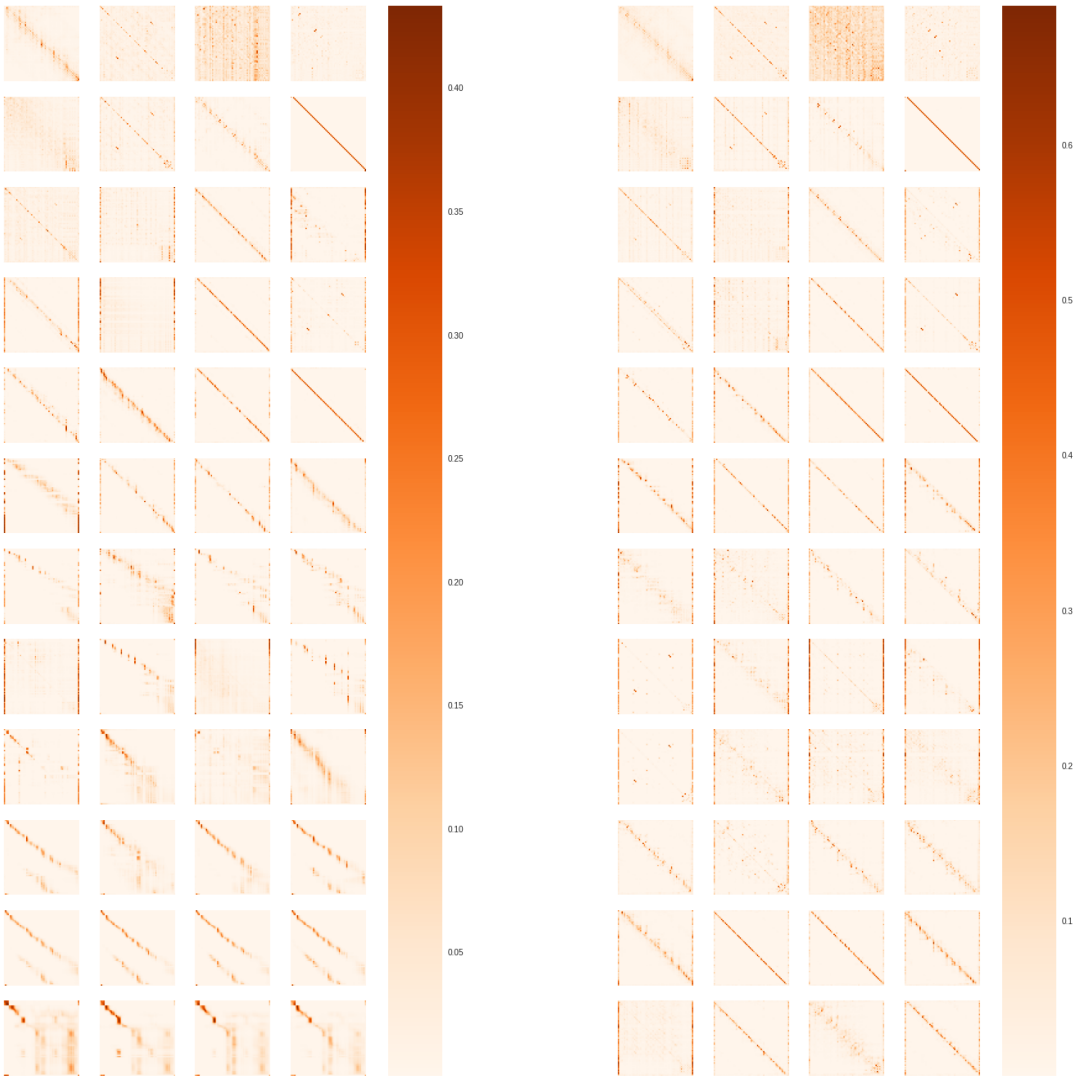


Figure 4: Types of head-specific token-wise attention maps proposed by Kovaleva et al. 2019



(a) Medal-trained accuracy (b) Base model accuracy

Figure 5: token-wise attention maps for an intubated patient in respiratory care

In the original paper by Kovaleva et al. 2019, the authors distinguish five 'types' of heads (4). This is done by taking the token-wise activation map of the different heads of the model and classifying them according to the observed pattern.

1. Vertical: mainly corresponds to attention to special tokens [CLS] and [SEP] which serve as delimiters between individual chunks
2. Diagonal: formed by the attention to the previous/following tokens;
3. Vertical+Diagonal: a mix of the previous two types,
4. Block: intra-sentence attention for the tasks with two distinct sentences (such as, for example, RTE or MRPC)
5. Heterogeneous: highly variable

We reproduce this analysis to compare our base model to the medal pre-trained model (5). We picked a few short example tokens, and plotted the token maps. We here show a specific example (index 4 in our dataset, with a 62 tokens lenght after processing)

- We observe the first six layers remain largely unchange and are dominated by vertical and diagonal layers, which is coherent with the widely accepted fact that the last layers of a model are more task specific. Layers seven to nine change to less easily categorized patterns.
- The last three layers are the most interesting, with layer eleven and twelve changing from almost exclusive diagonal layers to an uncategorized yet repeating pattern of double diagonal. The layer layer switches to mostly diagonal blocks, suggestion increased attention to nearby tokens.

While at this stage this is simply interesting heuristics, more in depth study of those patterns could provide important information on head specialization patterns. In particular, looking at multiple examples with comparable grammatical structures could help investigate possible semantic specialization.

4.2 Changes in attention patterns

To illustrate the changes in attention patterns, we plotted the per-head cosine similarity self-attention maps between base and fine-tuned models over sampled dataset examples (6). So over several examples, we quantify the similarity in actiavtions patterns for both models, with lighter colors correspond to greater similarity.

We here confirm that the main differences between the two models are in the last heads, from head eight to head twelve, so we can infer the pre-training on the MeDAL dataset really has an on the specialization of the model.

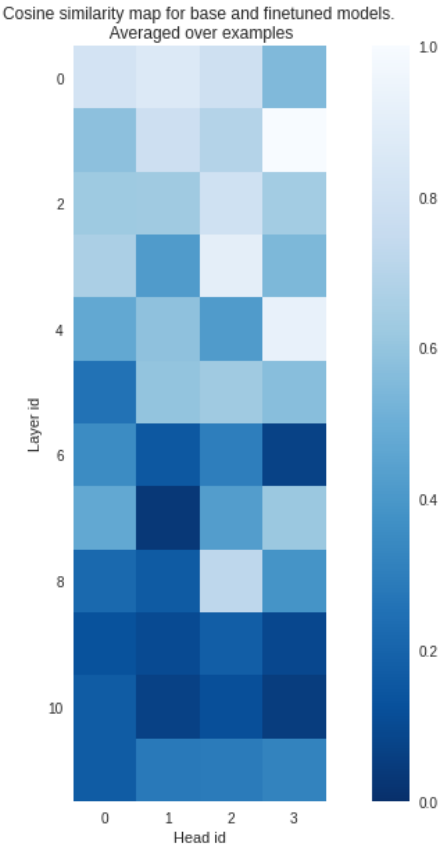


Figure 6: Per-head cosine similarity between base and fine-tuned model self-attention maps averaged over sampled dataset examples.

4.3 Robustness to pruning

To show the influence of pruning, we focused on the loss and accuracy of several pruned models. First, we compared the results of the base model pruned and unpruned, and finally the ones of the medal pre-trained model pruned and unpruned.

The pruned models were randomly pruned, meaning we randomly selected which heads of which layers were to be pruned. We evaluate both our models on the same examples and repeat the experiment 100 times.

The tables show gives the results of the first experiment on the base and pre-trained models. We can see the results degrade with pruning model, which is expected. What is more surpising is the significant loss of performance of the pre-trained model, as compare to the base model, which suggest a decreased robustness of the model to pruning.

An other observation we made during the runs is that when the pruning happens on early layers, such as layers 1 to 3, the results are really low comparing to pruning on latter layers. This shows the importance of the heads in the first layers of the model.

	mean loss	mean accuracy
base model	0.42	0.81
pruned model	0.50	0.76

Table 1: Mean loss and accuracy over 100 random head prunings for the base models

	mean loss	mean accuracy
base model	0.42	0.82
pruned model	0.72	0.66

Table 2: Mean loss and accuracy over 100 random head prunings for the MeDAL pretrained models

References

Kovaleva, Olga et al. (2019). "Revealing the dark secrets of BERT". In: *arXiv preprint arXiv:1908.08593*.

Wen, Zhi, Xing Han Lu, and Siva Reddy (2020). "MeDAL: Medical abbreviation disambiguation dataset for natural language understanding pretraining". In: *arXiv preprint arXiv:2012.13978*.