Applied Deep Learning VU

Medical Condition Extractor

Patrick Styll

WS23

1 Introduction

The main idea is free-text processing for extracting diagnoses and diseases from medical notes. This type of named entity recognition might be useful for e.g. converting unstructured data to specially constructed standards, suitable for deployment in Hospital Information Systems.
Admission notes are free-text notes, written by doctors when a new patient is admitted to the hospital. These notes include important patient details, such as gender, age, and several medical conditions. In the end, the doctor has to manually input this information into an Electronic Health Record (EHR). This process represent a bottleneck in the whole medical workflow, and Deep Learning, i.e. Named Entity Recognition, may be the solution to this problem.

1.1 Architecture

For this kind of NER project, BERT models are anticipated to be effective. As such, the plan is to harness BERT base models and adapt them for this specialized task.

1.2 Dataset

The primary dataset originates from the TREC CT topics, publicly accessible here.
Each topic has a similar structure, including several diagnoses in free text format. The topics represent admission notes - notes with the most important patient details, which a doctor takes as soon as a person is admitted to a hospital. This includes personal information and demographics, such as gender and age, but also and most importantly the current medical conditions, personal medical history and family medical history. For simplification purposes, the focus lies on detecting diseases/diagnoses present in the text, covering conditions such as diabetes mellitus or high blood pressure.
This dataset makes a total of 255 entries (topics). This includes:

- **topics2016.xml** - valuable information in note, description and summary. 30 topics in total. The fields could be processed individually, though, creating a total of 90 topics.
- **topics2021.xml** - 75 topics in total.
- **topics2022.xml** - 50 topics in total.
- **topics2023.xml** - preprocessed to free text in admission note style via LLM - 40 topics in total.

1.2.1 Additionals

Should the topic-dataset not be enough in case of inference (e.g. error metrics too high), I will include more data from the ClinicalTrials database. It contains information on clinical trials, including free text descriptions on said trials. This may be useful to further enhance the model's performance - given the complexity of annotating this kind of data, I would consider this only if the model's vocabulary does not suffice.

Since vocabulary in the medical world is complex and diverse, it might be incredibly useful to enhance the model's vocabulary with already existing medical thesauri. Some of which (such as ICD-10) are publicly available and continuously updated by medical professionals.
Language: All data (text) being used in this project will be in English.

1.3 Transfer Learning

In fact, there already exist Machine Learning models trained for similar purposes. These include, for instance, BioBERT [Alonso Casero(2021)]. This model is based on BERT architecture and is trained on free-text medical data in order to extract, among others, diseases. In addition, medSpaCy [Eyre et al.(2021)] is another ML-based technique to extract various forms of patient-related data from free-text. This also includes labelling medical conditions as family medical history, past medical history and current medical history.
Since BioBERT is based on the same model architecture which we are going to use, we are going to try to train BioBERT on the downstream-task of analyzing admission notes. We will then compare the transfer-training model and the newly trained uncased BERT model in terms of similarity and performance.

2 Dataset Preparation

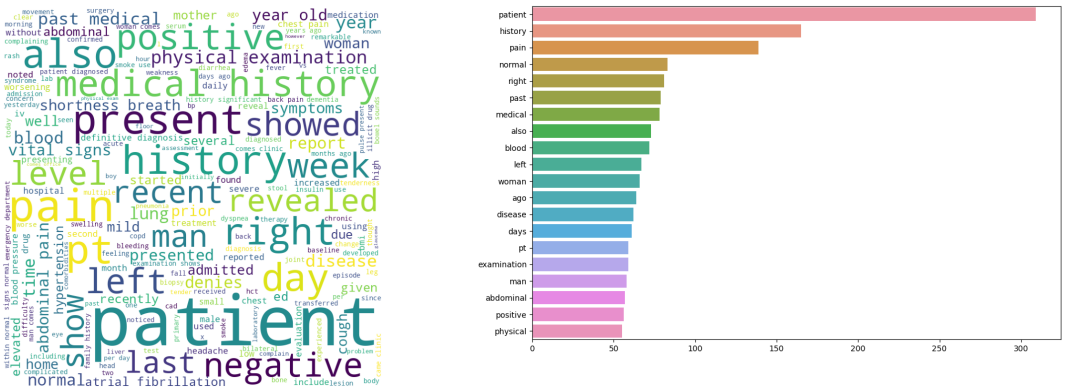
2.1 Labelling

The labelling of the dataset has been done via the open-source tool doccano [Nakayama et al.(2018)]. There were several issues while labelling data as *medical conditions*. The term *medical condition* is not entirely clear and subject to interpretation. For instance, we can observe the relationship between *medical condition* and *symptom*. E.g., a *fever* is not a medical condition - it is a response to medical condition or disease. The same goes for *dysuria*, being the subsequent response to e.g. UTIs (Urinary Tract Infections), a collection of various medical conditions. On the other hand arises the question - are injuries also medical conditions? In fact, injuries, such as a broken arm, are not considered medical conditions - surprisingly, injuries themselves are their own category in the medical lingo.
Another difficulty in working with the data was the extensive use of acronyms and abbreviations (e.g. *CAD* standing for *Coronary Artery Disease*). These are trivial for domain experts, but incredibly hard to work with when not being acquainted with the specialized medical lingo. However, the community-driven website All Acronyms provided great support in this task.

2.2 Understanding

2.3 Word Frequencies

As we can see in figure 1b, there are no entites, i.e. medical conditions, in the top words. However, as can be seen in the bottom of the wordcloud in figure 1a, *artrial fibrillation*, a medical condition, shows up.

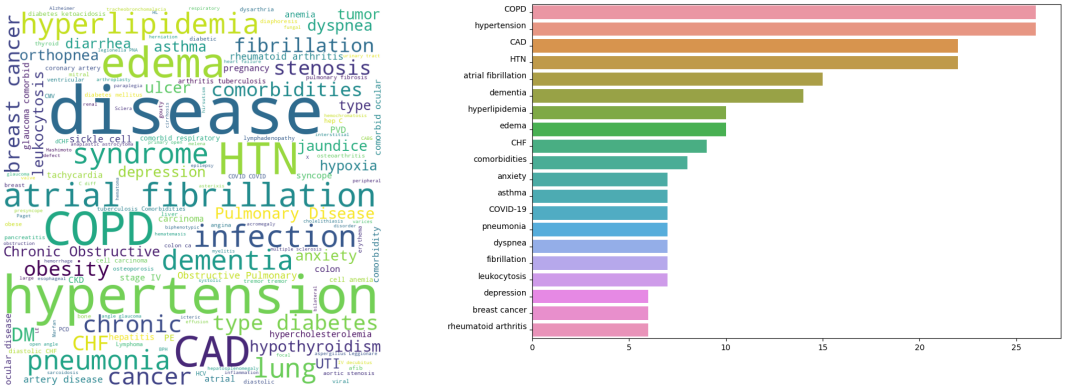


(a) Wordcloud (b) Frequency Table

Figure 1: Frequencies of Words in Admission Notes

2.4 Entity Frequencies

Interestingly, as seen in figure 2b, COPD and hypertension show up incredibly frequently. This, along with disease and CAD, is also reflected in the wordcloud in figure 2a.

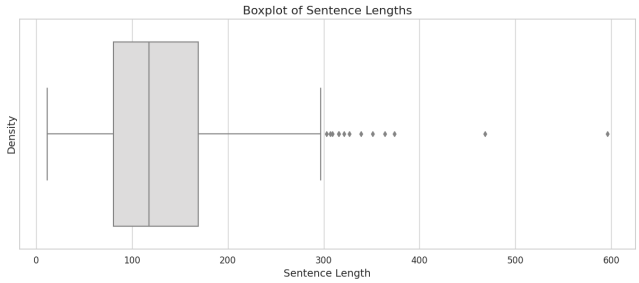


(a) Wordcloud (b) Frequency Table

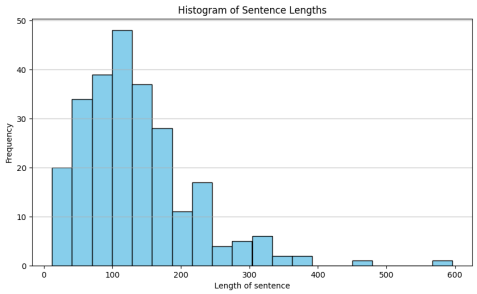
Figure 2: Frequencies of Medical Conditions in Admission Notes

2.5 Sentence Length

The sentence length is important to choose an appropriate length for the input layer of our BERT model. As seen in figure 3b, most sentences fall into a length of below 250 words. In fact, as seen in the boxplot in figure 3b, 75% of all admission notes have a length of below ≈ 180 words.
With this in mind, we can roughly pinpoint an optimal size for the input layer. We are going to experiment with the input lengths of 128, 256 and 512.



(a) Boxplot



(b) Histogram

Figure 3: Visualization Graphs of Admission Note Length

3 Training

3.1 Metrics

For this task (named entity recognition), we want both high precision and high recall. The precision measures the percentage of the model's predicted entities that are correct. High precision indicates a low false positive rate, which is important in medical applications, as a means to avoid false alarms. On the other hand, recall measures the percentage of actual entities that the model correctly identified. High recall is important in medical NER, to ensure that no critical information is missed.
The logical conclusion is using the harmonic mean of both, which is the f1 score. It provides a balance between precision and recall. An ideal model will have both high precision and high recall, leading to a high f1-score - since this is often times the primary metric for NER tasks, this is the metric we will focus on during hyperparameter-tuning.
Using ambiguous parameters with the Stochastic Gradient Descent optimization and an input size of 128 already provided great results, with an f1-score of ≈ 0.85.

3.2 Hyperparameter Tuning

There were a few observations I was able to make during tuning the BERT model.
First of all, changes in the learning rate were significant. Higher learning rates lead to much worse results. For instance, a learning rate of 0.1 lead to an f1 score of ≈ 0.3, while an f1 score of 0.01 lead to an f1 score of ≈ 0.8 with the same hyperparameters. On the other hand, a learning rate that is too small leads to similar behaviour - e.g. a learning rate of 0.0001 lead to an f1-score of ≈ 0.3. The same drop, just in a less extreme manner, can be seen for a learning rate of 0.001, with an f1-score of ≈ 0.5.
Similar behaviour can be observed for changes in the batch size, just in a more moderate way - which makes sense, taking generalization into account. In practice, a higher batch size lead to slightly worse f1-scores. While a batch size of 16 lead to an f1-score of 0.92, a batch size of 32 lead to an f1-score of 0.83.

Interestingly, though, the Adam optimizer performed poorly in comparison to Stochastic Gradient Descent with Momentum. Using the Adam optimizer actually lead to catastrophic forgetting, the f1-score dropped significantly and loss skyrocketed.
The f1-score does no longer significantly change after ≈ 10 epochs, it often even drops afterwards. Regarding the input size, I have found that there is just a slight loss in the f1-score when using a higher input size, i.e. 512 in our case. This makes sense, since the many paddings for the usually shorter admission notes may distort the outcome a bit. In our case, the shortest input size of 128 worked best.
In the end, after tuning the model, we received following parameters:

- **batch size:** 8
- **learning rate:** 0.01
- **epochs:** 10
- **optimizer:** SGD
- **input size:** 128
- **model architecture:** BERT base uncased

These parameters gave us a final f1-score of **0.964**.
The training history of the final model can be observed in figure 4, where we can see a significant increase in the f1-score in the first 5 epochs, followed by a severe drop, which changes again after epoch 7.

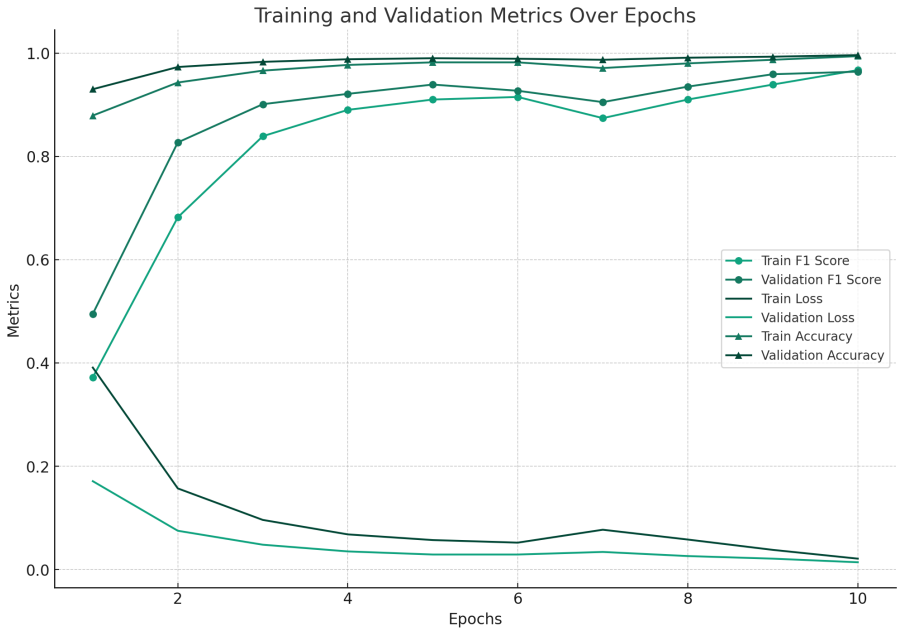


Figure 4: Training with optimal Parameters

3.3 Transfer-Learning

Unfortunately, the transfer-learning technique seems to have been a failure - it usually performed slightly than the newly trained BERT base model. After looking at the training data of BioBERT, it was clear why the transfer learning model is so prone to creating false positives: The definition used for labelling the *disease* data is slightly different to how we define a *medical condition*. For instance, the transfer learning model, after already trained on admission notes, still classifies *diarrhea* as a disease/medical condition. In our use-case, though, we defined *diarrhea* to not be a medical condition, but rather a symptom. All these differences sum up to give us a slightly less high f1-score, which is incredibly unfortunate.

4 Insights and Timetable

The project was wonderful for exploring the possibilities of Named Entity Recognition, and showed me how to prepare reusable architecture for such tasks. However, I did run into several problems, especially with annotating data - in the future, I am going to specifically define what to include and exclude in an entity. During the course of annotating my data, I had to redo labelling a few times, simply due to the fact that I changed my mind repeatedly on how I would define a medical condition. After a few attempts, I gave up and looked up several definitions, looked into examples and thesauri, until I finally was able to pinpoint exactly what a medical condition is, and accurately separate it from symptoms. Trivially, annotating data was much simpler afterwards.

4.1 Timetable

- **Requirement-Engineering**
  Time Planned: 5h
  Spend: 4h
  Notes: Thanks to a colleague, finding appropriate tools for data annotation was easy.
- **Capturing and Annotating Data**
  Time Planned: 25h
  Time Spent: 35h
  Notes: Data Annotation was way harder than I anticipated, vastly due to the very complicated medical lingo and unclearly defined differences between symptoms and medical conditions.
- **Describing Data**
  Time Planned: 5h
  Time Spent: 3.5h
  Notes: Insights were incredibly useful for finding a good value for maximum tokens.
- **Implementing BERT**
  Time Planned: 15h
  Time Spent: 22h
  Notes: Unexpected difficulties utilizing BioBERT for transfer-learning, due to the fact that the entities had a different label.
- **Tuning BERT**
  Time Planned: 10h
  Time Spent: 19h (active)
  Notes: Setting up the environment for hyperparameter-tuning was not as hard as expected, but the tuning itself was way more computationally intense than anticipated. Furthermore, there initially was weird behaviour with the error metrics (explosive error rate and rapid forgetting) - fixing the bug was rather expensive.

References

[Alonso Casero(2021)] Álvaro Alonso Casero. 2021. Named entity recognition and normalization in biomedical literature: a practical case in SARS-CoV-2 literature. (July 2021). `https://oa.upm.es/67933/` Unpublished.

[Eyre et al.(2021)] Hannah Eyre, Alec B. Chapman, Kelly S. Peterson, Jianlin Shi, Patrick R. Alba, Makoto M. Jones, Tamara L. Box, Scott L. DuVall, and Olga V. Patterson. 2021. Launching into clinical space with medspaCy: a new clinical text processing toolkit in Python. *CoRR* abs/2106.07799 (2021). arXiv:2106.07799 `https://arxiv.org/abs/2106.07799`

[Nakayama et al.(2018)] Hiroki Nakayama, Takahiro Kubo, Junya Kamura, Yasufumi Taniguchi, and Xu Liang. 2018. doccano: Text Annotation Tool for Human. `https://github.com/doccano/doccano` Software available from https://github.com/doccano/doccano.