AI-Selectie hulp verwijsbrieven

Ik heb een aparte github repo gemaakt, met 1 enkel script waarin je alle modellen kan aanroepen, die vervolgens een confidence score geeft dat een patient de diagnose:

- **RA** : reumatoide arthritis
- **FMS** : fibromyalgie
- **OA**: osteoarthritis
- **Chronic**: patiënten die langer dan 3 maanden onder behandeling van een reumatoloog blijven (d.w.z., chronisch)

Let op deze scores zijn niet gecalibreerd! (Dus 20% confidence bij RA, kan een hoge kans representeren, terwijl een 20% confidence bij FMS een lage kans kan zijn.) . Ons voorstel is vooral om de volgorde van scores te gebruiken, en indien je een cut-off wil toepassen vooral de originele paper te raadplegen (https://www.nature.com/articles/s41746-025-01495-4)

De slimme AI-triage tool is te downloaden als python programma/software via de volgende link: https://github.com/levrex/implement_triage_agent

In dit document vind je de nodige instructies wat betreft het klaarzetten van de benodigde Python omgeving, dit gebeurt via conda (die ook beschikbaar is voor windows):

H1: Omgeving klaarzetten

Het script is geschreven in Python 3.6.13, en is afhankelijk van bepaalde modules. Om deze omgeving op te zetten, kan je conda gebruiken. Deze is beschikbaar voor zowel windows als linux omgeving:

 # Setting up Conda omgeving

Stap 1. download conda

- https://www.anaconda.com/docs/getting-started/miniconda/main
- https://www.anaconda.com/download/success

Ik zou adviseren om miniconda te installeren i.p.v. de volledige anaconda, dit is namelijk een veel kleinere package,

Stap 2. Zorg ervoor dat conda in je PATH environment staat (Windows only)

- `set PATH=%PATH%;[PATH_TO_CONDA]\anaconda3\condabin`

  Voorbeeld:

- `set PATH=%PATH%;C:\Users\tdmaarseveen\AppData\Local\anaconda3\condabin`

Stap 3: creer een conda omgeving (we noemen de omgeving 'triage_env')

- `conda create -n triage_env python=3.6.13`

Stap 4: activeer je conda omgeving

- `conda activate triage_env`

Stap 5: installeer pip (soort package manager) in je conda omgeving

- `conda install pip`

Stap 6. Download de github repo & plaats op een gewenste locatie in jullie systeem

- Download handmatig : https://github.com/levrex/implement_triage_agent
  - Alternatief: `git clone https://github.com/levrex/implement_triage_agent`

Stap 6. bouw de noodzakelijke python omgeving voor je script binnenin de conda environment

- `pip install -r [PATH_TO_REPO]/requirements.txt`

  Alternatief: Je kan de packages ook handmatig installeren

  - *in requirements.txt staan alle externe modules (inclusief de versies) die de pipeline nodig heeft om het script te draaien. Link: https://github.com/levrex/implement_triage_agent/blob/main/requirements.txt*

Stap 7. Je conda environment is gereed!

- je kan eventueel je conda omgeving weer sluiten in cmd:
  - `conda deactivate triage_env`

  Alternatief: run voorbeeld script: build_conda_env.bat

Disclaimer: in mijn ervaring is het opzetten van de conda omgeving soms nog het lastigste gedeelte. Ik ben beschikbaar voor een teams call indien je hier tegen problemen aanloopt:

T.D.Maarseveen@lumc.nl

H2: Hoe pas je het script (a.k.a. ‘de API’) toe?

Stap 1 : activeer je specifieke conda omgeving (zie hoofdstuk 1) voordat je het script aanroept!

Voorbeeld

- `conda activate triage_env`

Stap 2: run python met prompt (inhoud verwijsbrief van patient)

Voorbeeld input:

- `python run_triage_agent.py --input 'Deze patient heeft reumatoide artritis, acpa positief , ochtend stijfheid sinds kindsaf, nu in remissie'`

Voorbeeld output:

```
{
  "FMS": 0.08652999997138977,
  "RA": 0.20974136888980865,
  "OA": 0.19083213806152344,
  "Chronic": 0.45154139399528503
}
```

Als output van het model krijg je de uitslagen, dit duurt momenteel een paar seconden.

Deze uitslagen zou je kunnen opslaan in een aparte Uitslagen tabel.

H3: Ideeën voor toekomstige uitvoeringen

Optimaliseren van efficientie -> Chunks i.p.v. per patients

Qua efficiëntie is de huidige versie, vrij omslachtig. Het hele model wordt opnieuw ingeladen per patient. En de tfidf-vectorisatie (vertalen van tekst naar vectors) wordt gerunt per patient. In toekomstige versie, is het misschien toch beter om de data in chunks te verwerken i.p.v. per patient?

Mogelijkheid om modellen te vervangen in de toekomst

Maar als eerste pilot lijkt het mij voor nu afdoende. Verder zou het als we eenmaal zoiets hebben opgezet, heel makkelijk zijn om in de toekomst de modellen te vervangen-> als we eenmaal betere iteraties van de modellen beschikbaar hebben, of indien er radicale veranderingen optreden in hoe wij mensen behandelen of triëren.