Self Learning Tutorial

AI in Healthcare
MSAI 395T

Summary

In this tutorial I will be investigating clinical notes corresponding to ICD-9 diagnoses related to depression in the MIMIC-III dataset.

This tutorial will cover:

1. Using SQL to create datasets of notes related to depression diagnoses in MIMIC-III.
2. Using the embeddings from Clinical-Bert to visualize important tokens within our datasets.
3. Using a pretrained Clinical-Bert to fine-tune a classifier model for diagnosing depression from clinical notes.

You can find all of the code for this tutorial at github.com/paplant/ai-in-healthcare/tree/main/self-learning-tutorial

Creating the datasets

I started with creating a local postgres database for MIMIC-III following the instructions provided by MIT-LCP
github.com/MIT-LCP/mimic-code/blob/main/mimic-iii/buildmimic/postgres

I wrote a number of SQL queries to construct the dataset. I will only be highlighting the queries themselves and not the results in these slides.

The queries and their outputs full can be found at
github.com/paplant/ai-in-healthcare/blob/main/self-learning-tutorial/sql/

Creating the datasets

First I wrote the following query to extract all of the ICD-9 codes related to depression. This query yields 23 ICD-9 codes.

See *sql/depression_icd9_output.csv* for details.

```
CREATE TEMP TABLE IF NOT EXISTS depression_icd9 AS (
     SELECT *
     FROM d_icd_diagnoses
     WHERE long_title ILIKE '%depressive%'
          OR long_title ILIKE '%depression%'
);
```

Creating the datasets

Next I found all diagnoses events matching one of the depression related ICD-9 codes. This yields 3745 depression related diagnoses events. See *sql/depression_events_output.csv.gz* for results.

```
CREATE TEMP TABLE IF NOT EXISTS depression_events AS (
     SELECT A.*, B.short_title, B.long_title
     FROM (
          SELECT subject_id, hadm_id, seq_num, icd9_code
          FROM diagnoses_icd
          WHERE icd9_code IN (SELECT icd9_code FROM
depression_icd9)
     ) AS A
     LEFT JOIN (
          SELECT icd9_code, short_title, long_title
          FROM depression_icd9
     ) AS B
     ON A.icd9_code = B.icd9_code);
```

Creating the datasets

Next I found all notes with hadm ids matching those in the events results. This query yields 90768 notes results. This is much larger than the number of depression related events because a single hospital admission event will generate many notes.

```
CREATE TEMP TABLE IF NOT EXISTS depression_notes AS (
    SELECT text
    FROM noteevents
    WHERE hadm_id IN (SELECT hadm_id FROM
depression_events)
);
```

Creating the datasets

Next I found all notes with hadm ids NOT matching those in the depression events results. Expectedly this query yielded a much larger 1760576 notes results.

```
CREATE TEMP TABLE IF NOT EXISTS non_depression_notes AS
(
    SELECT text
    FROM noteevents
    WHERE hadm_id NOT IN (SELECT hadm_id FROM
depression_events)
);
```

Creating the datasets

The two generated tables `depression_notes` and `non_depression_notes` are much larger than I need them to be, so I took a random sample of 10000 samples from each to create my final datasets.

```
\copy (SELECT * FROM depression_notes TABLESAMPLE SYSTEM(20) LIMIT 10000) TO
'depression_notes_output.csv' WITH CSV HEADER;

\copy (SELECT * FROM non_depression_notes TABLESAMPLE SYSTEM(10) LIMIT 10000) TO
'non_depression_notes_output.csv' WITH CSV HEADER;
```

See *sql/depression_notes_output.csv.gz* and *sql/non_depression_notes_output.csv.gz* for details

Exploring the embeddings

With the two datasets created I can now examine some of the embeddings.

I will not go into all of the code here, only a summary of the results. The full code can be found at github.com/paplant/ai-in-healthcare/blob/main/self-learning-tutorial/self-learning-tutorial.ipynb

Exploring the embeddings

First I extracted the 100 most frequent words from each dataset (after filtering out stop words and small words) and compared the two outputs.

Below are the tokens only appearing within the depression related dataset. It is interesting to note that no obvious words related to depression appear within the top 100 words.

depression only tokens: ['access', 'admitting', 'also', 'amount', 'apical', 'appearance', 'appears', 'appreciable', 'becomes', 'bilobectomy', 'bronch', 'cancer', 'change', 'chest', 'clip', 'coiled', 'communicated', 'compared', 'confluent', 'contraindications', 'contrast', 'copd', 'correct', 'course', 'currently', 'decrease', 'desaturations', 'diagnosed', 'dobbhoff', 'effusion', 'endotracheal', 'evaluate', 'evaluation', 'extensive', 'female', 'final', 'findings', 'glass', 'ground', 'hemorrhage', 'identifier', 'images', 'improved', 'includes', 'increasing', 'initial', 'interval', 'intubation', 'jbre', 'line', 'lower', 'lung', 'mdct', 'moderate', 'multifocal', 'nasogastric', 'newly', 'number', 'nurse', 'opacity', 'optiray', 'overall', 'parenchymal', 'physician', 'pleural', 'pneumothorax', 'portable', 'position', 'previous', 'prior', 'projects', 'proximal', 'pulmonary', 'radiograph', 'reason', 'recently', 'relatively', 'report', 'requirements', 'requiring', 'shows', 'slight', 'subclavian', 'substantially', 'time', 'tube', 'tubes', 'unchanged', 'venous', 'well', 'woman', 'year']

Exploring the embeddings

The non depression related tokens and the common intersection are below.

It is interesting how few tokens in the top 100 are shared between the two datasets. This might imply that our selection criteria is important in determining the content of the notes.

not depression only tokens: ['accompanied', 'admission', 'anesthesiology', 'apparently', 'aspirated', 'asystole', 'asystolic', 'beats', 'bleed', 'blood', 'called', 'cardiac', 'cardioverted', 'care', 'ceftazidime', 'chief', 'code', 'complete', 'complex', 'concerning', 'covers', 'date', 'dates', 'depressions', 'dictation', 'discharge', 'drain', 'erythromycin', 'evident', 'extremity', 'facial', 'found', 'fourth', 'gentleman', 'given', 'gram', 'hernia', 'history', 'hospital', 'hydralazine', 'intensive', 'intubated', 'joules', 'labetalol', 'large', 'management', 'medications', 'metoprolol', 'minute', 'narrow', 'neurology', 'neurontin', 'note', 'noted', 'obtained', 'outside', 'past', 'point', 'pontine', 'present', 'pressure', 'pressures', 'progressed', 'prolonged', 'protonix', 'pulse', 'quite', 'rapidly', 'rates', 'regained', 'return', 'returned', 'rhythm', 'segment', 'several', 'sided', 'sinus', 'suctioned', 'supraventricular', 'systolic', 'tachycardia', 'team', 'times', 'total', 'transferred', 'troponin', 'unit', 'unstable', 'ventricle', 'ventriculostomy', 'weakness', 'wife']

common tokens: ['left', 'medical', 'patient', 'placed', 'please', 'post', 'right', 'status']

Exploring the embeddings

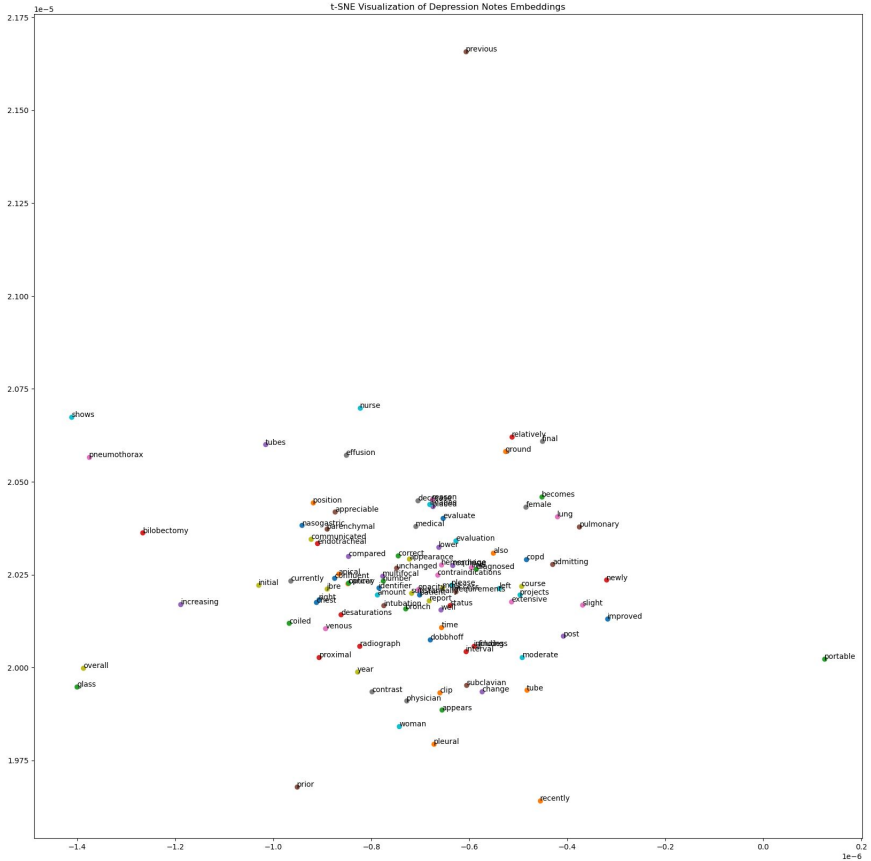
Next I visualized the word embeddings for the top 100 tokens in both datasets using TSNE.

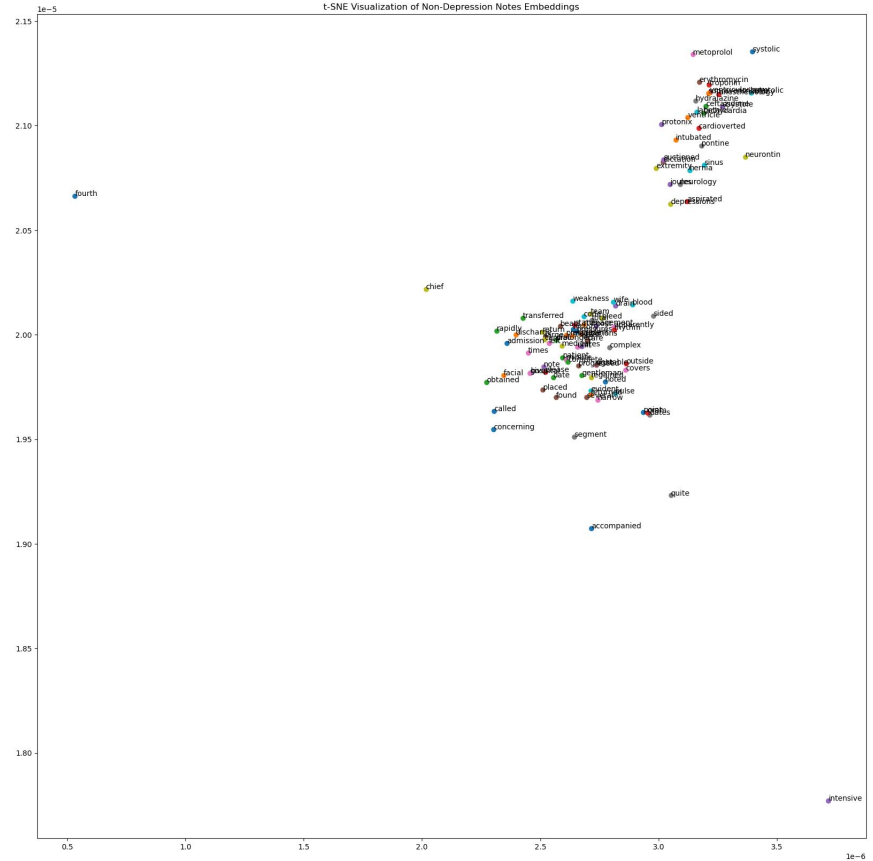
For the NLP tasks I used a BERT based model: emilyalsentzer/Bio_ClinicalBERT for tokenization as shown did in the lectures.

For TSNE I experimented with an alternative implementation by RAPIDS.ai in their cuml package. This implementation of TSNE is GPU accelerated and will execute much faster than the CPU only implementation in scikit-learn.

See https://docs.rapids.ai/api/cuml/stable/api/#tsne for details.

Exploring the embeddings





Exploring the embeddings

Looking at the two TSNE plots of the top 100 tokens, I can see the tokens from the depression related dataset appear to be less tightly coupled than those of the non-depression related samples. This might imply that the notes related to depression are selecting for a different set of tokens, while the notes for non depression are clustered around the most common tokens in the notes dataset overall.

Training a classifier

Next I attempted to train a model to predict a depression diagnoses from the contents of a note.

I use the two datasets along with the pretrained Bio_ClinicalBERT to fine tune a classifier for predicting depression based on the provided clinical notes.

I tagged the samples from the two datasets (0 not depression, 1 depression), merged them, and then split the data into a 16000 training samples, 2000 test samples, and 2000 validation samples (80/10/10).

I used the AutoModelForSequenceClassification from hugging face to fine tune the pretrained model and trained for three epochs.

Training a classifier

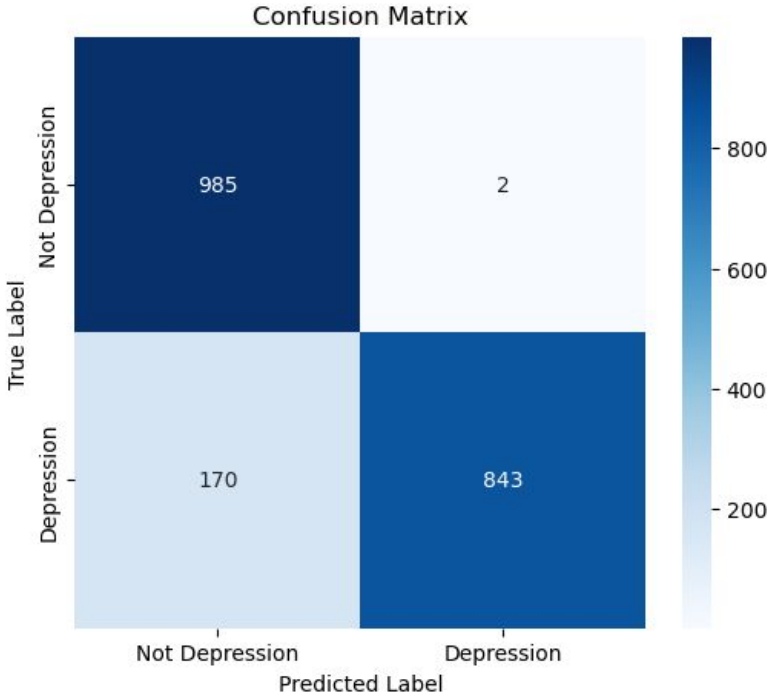
After training I evaluated the model on the validation set. It achieved a strong F1-Score of 0.92 and perfect precision on the label Depression.

	Precision	Recall	F1-Score	Support
Not Depression	0.85	1.00	0.92	987
Depression	1.00	0.83	0.91	1013
Accuracy			0.91	2000
Macro Avg	0.93	0.92	0.91	2000
Weighted Avg	0.93	0.91	0.91	2000

Training a classifier

The confusion matrix on the right was produced from evaluating the model on the validation set. It show that the model performs well overall. Almost all of its mistakes are on false negatives.

Given tokens related to depression did not appear in the top 100 tokens of our dataset, it must be that the model has learned other less obvious patterns in the data and may generalize.



Recap

In this tutorial:

- I showed how to use SQL for creating a dataset based on the criterion of depression with related notes, in Postgres.
- Then I explored the word embeddings of the notes and visualized them using a GPU accelerated TSNE by RAPIDS.ai.
- Finally I demonstrated how to fine tune a pretrained BERT model on our dataset to classify depression diagnoses and evaluated the results to determine that our model had learned to classify and might generalize to novel samples.