MIMIC-III NLP and Reproduction Code

AI 395T | **AI in Healthcare** | Dr. Ying Ding

Presenter: [redacted]

This slide deck covers 9 **outputs and** (6 required + 3 bonus) and code-behind for each:

Introduction

What disease did I pick?

I picked disease codes related to **Hyperlipidemia** [2724, 2721], which is a common condition characterized by high levels of lipids (fats) in the blood. It is a risk factor for cardiovascular diseases, including heart attacks and strokes, which are leading causes of death worldwide.

What about the text data?

The resultant patients_df dataframe had shape (29, 2) (29 entries) and the TEXT column is comprised of 361,438 characters (or approximately, 54,365 words).

How did you display the results of the t-SNE plots?

In the slides, I provide 3 charts for each value: the plot without entities labeled to observe clusters, the plot with entities labeled, and then a sampled plot with labels (for visual validation of terms).

The Process Step 1: Read the Data in

I based my read in off of Terence Lim's endorsed post on Ed Discussion.

[5]:
```
notes_path = 'data/NOTEEVENTS.csv'
diagnoses_path = 'data/DIAGNOSES_ICD.csv'
summary_path = 'data/summary.csv'
```

[6]:
```
# Read in code adapted from Terence Lim's endorsed post in Ed Discussion
notes_df =  pd.read_csv(notes_path, low_memory=False)
diagnoses_df = pd.read_csv(diagnoses_path, low_memory=False)
```

[66]:
```
dis_sum_df = notes_df.loc[notes_df['CATEGORY'] == 'Discharge summary', ['SUBJECT_ID', 'HADM_ID', 'TEXT']]
dis_sum_df['subj_hadm'] = list(zip(dis_sum_df['SUBJECT_ID'].astype(int), dis_sum_df['HADM_ID'].astype(int)))

# I decided to focus on hyperlipidemia related diagnoses
disease_list = ['2724', '2721']
disease_df = diagnoses_df[diagnoses_df['icd9_code'].isin(disease_list)].copy()
disease_df['subj_hadm'] = list(zip(disease_df['subject_id'].astype(int),
                                   disease_df['hadm_id'].astype(int)))

patients_df = dis_sum_df[['TEXT', 'subj_hadm']]\
    .join(disease_df.set_index('subj_hadm')['icd9_code'], on='subj_hadm', how='left')\
    .dropna()\
    .drop(columns=['subj_hadm'])

print('Verify that diseases extracted equals those in input list:',
      sorted(disease_list), sorted(np.unique(patients_df['icd9_code'])), sep='\n')

patients_df.to_csv(summary_path)
patients_df.shape
```

```
Verify that diseases extracted equals those in input list:
['2721', '2724']
['2721', '2724']
```

[66]:
```
(29, 2)
```

The Process Step 2: Entity Extract

Next, I performed entity extract using this generic template (piping changed, but process remains the same)

```
def extract_entities(text):
    doc = nlp(text)
    entities = [(ent.text, ent.start_char, ent.end_char, ent.label_) for ent in doc.ents]
    return entities

selected_disease_codes = ['2724', '2721']
selected_disease_patients_df = patients_df[patients_df['icd9_code'].isin(selected_disease_codes)]

# Process each discharge summary text and extract entities
for index, row in selected_disease_patients_df.iterrows():
    text = row['TEXT']
    entities = extract_entities(text)
    # Visualize entities using displaCy
    doc = nlp(text)
    displacy.render(doc, style='ent', jupyter=True)
```

Where nlp() piping was used for both spacy, scispacy, and (later) MedSpacy

Entity Extract: Spacy vs. SciSpacy

What is this showing?

Here we can see the limited NER capability of spacy in the healthcare domain and the improvement a domain specific training set can have.

Spacy

Major Surgical or Invasive Procedure:

[** 2184-8-5 DATE **]: Cardiac ORG catheterization, no intervention

History of Present Illness WORK_OF_ART :

72 CARDINAL yo F PRODUCT with PMHx of 2vessel CAD s/p RCA atherectomy in '[** 67 CARDINAL **],

HTN, morbid obesity, Hyperlipidemia PERSON who presents with dyspnea x

3days CARDINAL , worse past day with a dry cough. Symptoms started

abruptly on Sunday night TIME with SOB ORG while walking to bathroom. SOB ORG

remaind persistent over the following days DATE , with worsening DOE ORG .

She initially presented to [** First ORDINAL Name8 (NamePattern2) **] [**Last Name (Namel

was found to

be hypotensive and was transferred to [**Hospital1 18 CARDINAL **] ED with suggested

diagnosis of PNA ORG , incidentally found to have elevated troponin

of 1.73 CARDINAL . She was started on heparain at OSH. Received azithro

and Ceftriaxone GPE at OSH. Was on neo at 100 mcg QUANTITY . Got RIJ ORG in ou

ED. Crackles at bases, Febrile to 100.1 CARDINAL . Gave levofloxacin. Put

on levophed in ED GPE . O2 CARDINAL sat high 90's on 4L. CXR ORG here appears t

have bilateral infiltrates. ECG ORG here afib rate [**Street Address(2) 5679**]

elevations V4 CARDINAL -V6. Patient denies chest pain.

SciSpacy

Major Surgical ENTITY or Invasive Procedure ENTITY :

[**2184-8-5**]: Cardiac catheterization ENTITY , no intervention ENTITY

History ENTITY of Present Illness:

72 yo F ENTITY with PMHx ENTITY of 2vessel CAD ENTITY s/p RCA ENTITY atherectomy ENTITY in '[**67**],

HTN ENTITY , morbid obesity ENTITY , Hyperlipidemia ENTITY who presents with dyspnea ENTITY x

3days , worse past day ENTITY with a dry cough ENTITY . Symptoms ENTITY started

abruptly on Sunday ENTITY night ENTITY with SOB ENTITY while walking ENTITY to bathroom ENTITY . SOB ENTITY

remaind ENTITY persistent ENTITY over the following days ENTITY , with worsening ENTITY DOE ENTITY .

She initially presented to [**First Name8 (NamePattern2) **] [**Last Name (NamePattern1) 5678**] hospital and was found to

be hypotensive ENTITY and was transferred to [**Hospital1 18**] ED ENTITY with suggested

diagnosis ENTITY of PNA ENTITY , incidentally ENTITY found to have elevated ENTITY troponin of 1.73 ENTITY . She was

started on heparain ENTITY at OSH ENTITY . Received azithro ENTITY and Ceftriaxone ENTITY at OSH ENTITY . Was on neo

ENTITY at 100 mcg ENTITY . Got RIJ ENTITY in our

ED ENTITY . Crackles ENTITY at bases ENTITY , Febrile ENTITY to 100.1. Gave levofloxacin ENTITY . Put

on levophed ENTITY in ED ENTITY . O2 ENTITY sat high 90's on 4L ENTITY . CXR ENTITY here appears to

have bilateral infiltrates ENTITY . ECG ENTITY here afib rate ENTITY [**Street Address(2 ENTITY) 5679**]

elevations ENTITY V4-V6. Patient ENTITY denies chest pain ENTITY .

The Process Step 3: word2vec/t-SNE plots

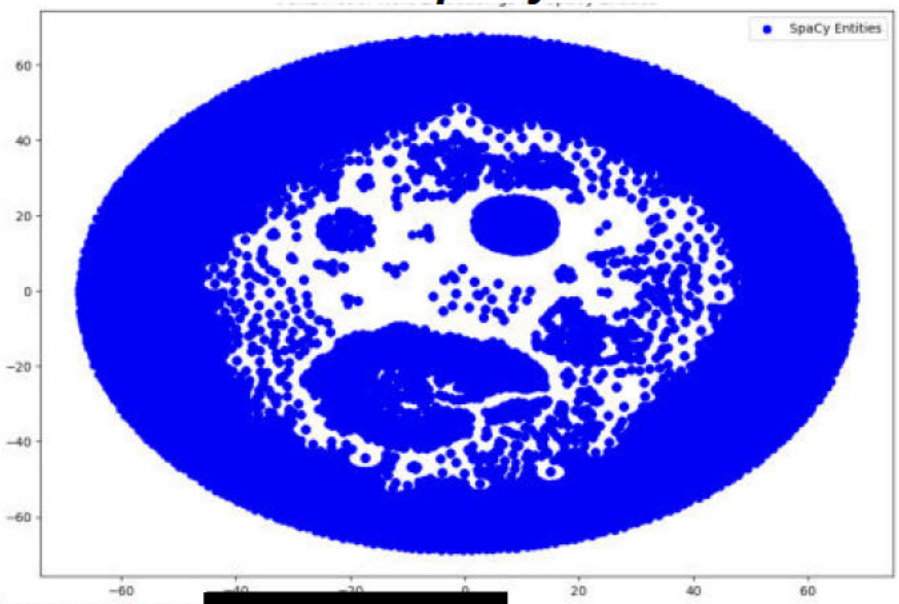
I made 3 versions of the plot (word2vec process remained the same) :

1. ***Without Labels (for visual simplicity given size)***
2. *With Labels (for completeness)*
3. *Sampled with Labels (for visual validation)*

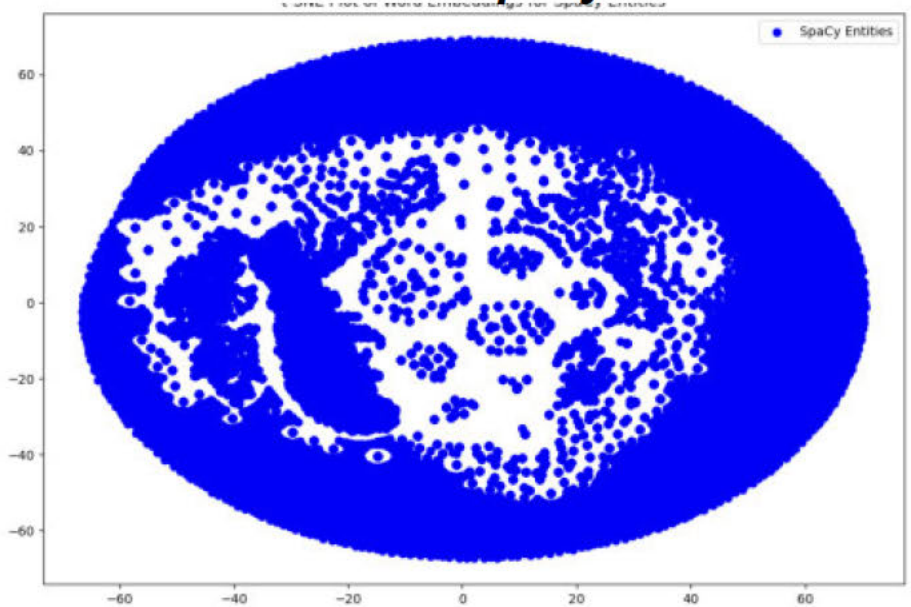
Plot without labels

```
word2vec_spacy = Word2Vec(sentences=spacy_sentences, vector_size=100, window=5, min_count=1, workers=4)
word_embeddings_spacy = [word2vec_spacy.wv[word] for sentence in spacy_sentences for word in sentence]

# t-SNE
tsne_model = TSNE(perplexity=11, early_exaggeration=12, n_components=2, init='pca', n_iter=1000, random_state=23)
word_embeddings_tsne_spacy = tsne.fit_transform(word_embeddings_spacy)

# Plot t-SNE
plt.figure(figsize=(12, 8))
plt.scatter(word_embeddings_tsne_spacy[:, 0], word_embeddings_tsne_spacy[:, 1], color='blue', label='SpaCy Entities')
plt.title('t-SNE Plot of Word Embeddings for SpaCy Entities')
plt.legend()
plt.show()
```

SpaCy



SciSpacy



The Process Step 3: word2vec/t-SNE plots

T-SNE plot code with and without sampling

```
def tsne_plot(model, words):
    "Creates a t-SNE model and plots it"
    labels = []
    tokens = []

    for word in words:
        if word in model.wv:
            tokens.append(model.wv[word])
            labels.append(word)
        else:
            print(f"Skipping '{word}' as it is not present in the model's vocabulary.")

    tsne_model = TSNE(perplexity=11, early_exaggeration=12, n_components=2, init='pca', n_iter=1000, random_state=23)
    new_values = tsne_model.fit_transform(np.array(tokens))  # Convert tokens to a NumPy array

    x = []
    y = []
    for value in new_values:
        x.append(value[0])
        y.append(value[1])

    plt.figure(figsize=(16, 16))
    for i in range(len(x)):
        plt.scatter(x[i], y[i])
        plt.annotate(labels[i],
                     xy=(x[i], y[i]),
                     xytext=(5, 2),
                     textcoords='offset points',
                     ha='right',
                     va='bottom')
    plt.show()
```

That's kind of messy, so let's sample it real quick:

```
def tsne_plot(model, words, sample_size=50):
    "Creates a t-SNE model and plots a sample of it"
    labels = []
    tokens = []

    # Select a random sample of words
    selected_words = np.random.choice(words, size=sample_size, replace=False)

    for word in selected_words:
        if word in model.wv:
            tokens.append(model.wv[word])
            labels.append(word)
        else:
            print(f"Skipping '{word}' as it is not present in the model's vocabulary.")

    tsne_model = TSNE(perplexity=11, early_exaggeration=12, n_components=2, init='pca', n_iter=1000, random_state=23)
    new_values = tsne_model.fit_transform(np.array(tokens))  # Convert tokens to a NumPy array

    x = []
    y = []
    for value in new_values:
        x.append(value[0])
        y.append(value[1])

    plt.figure(figsize=(16, 16))
    for i in range(len(x)):
        plt.scatter(x[i], y[i])
        plt.annotate(labels[i],
                     xy=(x[i], y[i]),
                     xytext=(5, 2),
                     textcoords='offset points',
                     ha='right',
                     va='bottom')
    plt.show()
```

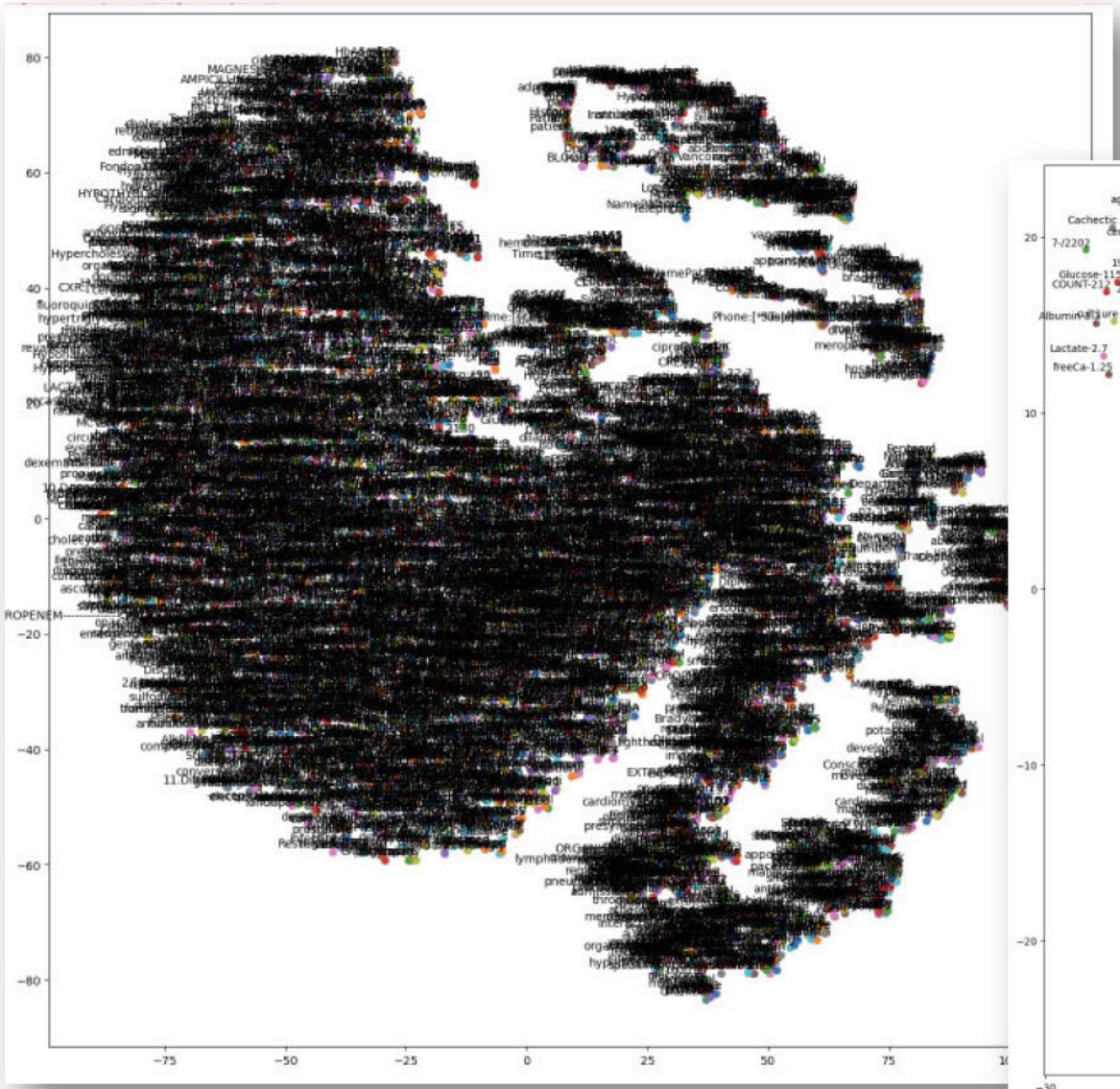
```
vocabs = word2vec_model.wv.index_to_key
sample_size = 100  # Set the sample size
tsne_plot(word2vec_model, vocabs, sample_size)
```

The Process Step 3: word2vec/t-SNE plots

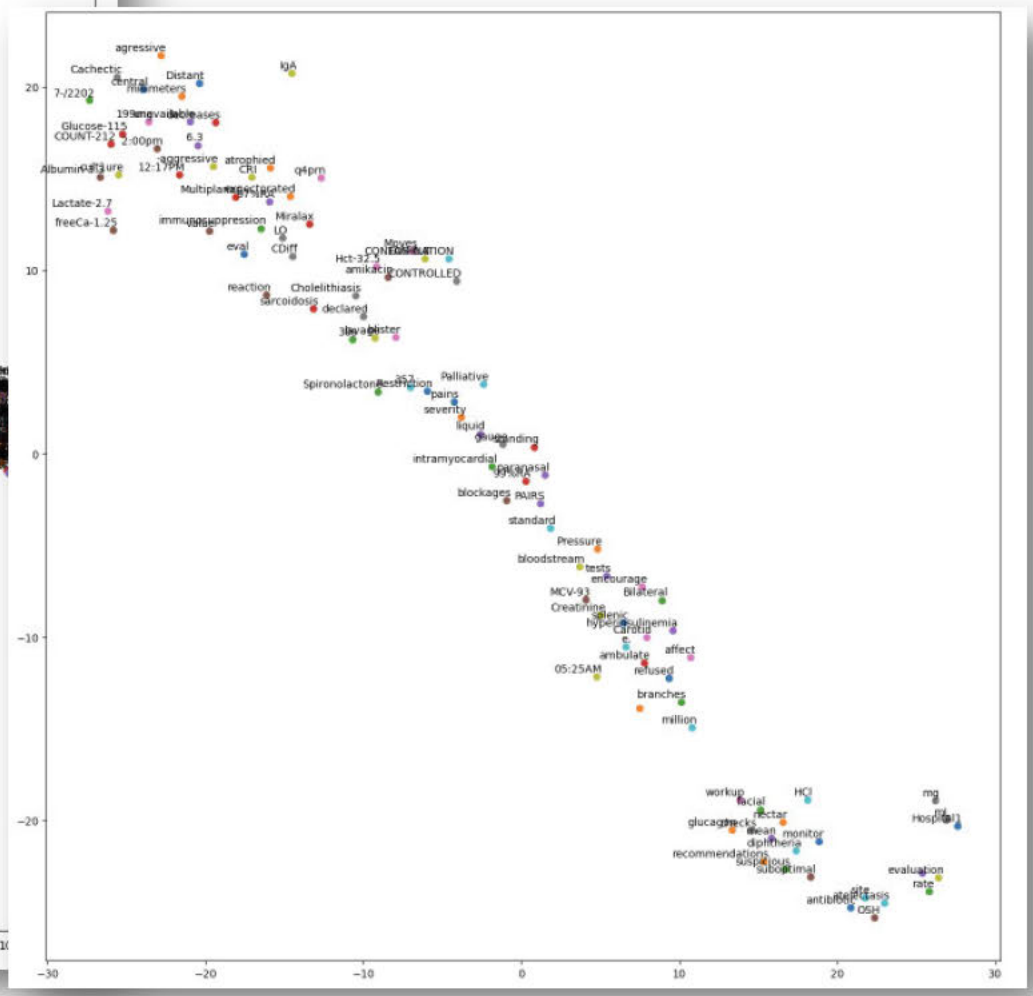
I made 3 versions of the plot (word2vec process remained the same) :

1. *Without Labels (for visual simplicity given size)*
2. ***With Labels (for completeness)***
3. ***Sampled with Labels (for visual validation)***

Full Plot



Sample Plot (n=100)



The Process Step 4: MedSpacy

Now, let's look at entity extract with MedSpacy

```
import medspacy
from medspacy.ner import TargetRule

selected_disease_codes = ['2724', '2721']
selected_disease_patients_df = patients_df[patients_df['icd9_code'].isin(selected_disease_codes)]

nlp = medspacy.load(enable=['sentencizer', 'medspacy_target_matcher'])
target_rules = [
    TargetRule('hyperlipidemia', 'DISEASE'),
    TargetRule('lipid', 'SUBSTANCE'),
    TargetRule('hypertension', 'DISEASE'),
    TargetRule('obesity','CONDITION'),
    TargetRule('cardiac','ENTITY')
]
nlp.get_pipe('medspacy_target_matcher').add(target_rules)

# Entity extract
medspacy_sentences = []
entities = []
for text in selected_disease_patients_df['TEXT']:
    doc = nlp(text)
    medspacy_sentences.append([token.text for token in doc if not token.is_stop])
    entities.extend([ent.text for ent in doc.ents if ent.label_ == 'DISEASE'])
```

Major Surgical or Invasive Procedure:

[**2184-8-5**]: Cardiac ENTITY catheterization, no intervention

History of Present Illness:

72 yo F with PMHx of 2vessel CAD s/p RCA atherectomy in '[**67**],

HTN, morbid obesity CONDITION , Hyperlipidemia DISEASE who presents with dyspnea x

3days , worse past day with a dry cough. Symptoms started

abruptly on Sunday night with SOB while walking to bathroom. SOB

remaind persistent over the following days, with worsening DOE.

She initially presented to [**First Name8 (NamePattern2) **] [**Last Name (NamePattern1) 5678**] hospital and was found to

be hypotensive and was transferred to [**Hospital1 18**] ED with suggested

diagnosis of PNA, incidentally found to have elevated troponin

of 1.73. She was started on heparain at OSH. Received azithro

and Ceftriaxone at OSH. Was on neo at 100 mcg. Got RIJ in our

ED. Crackles at bases, Febrile to 100.1. Gave levofloxacin. Put

on levophed in ED. O2 sat high 90's on 4L. CXR here appears to

have bilateral infiltrates. ECG here afib rate [**Street Address(2) 5679**]

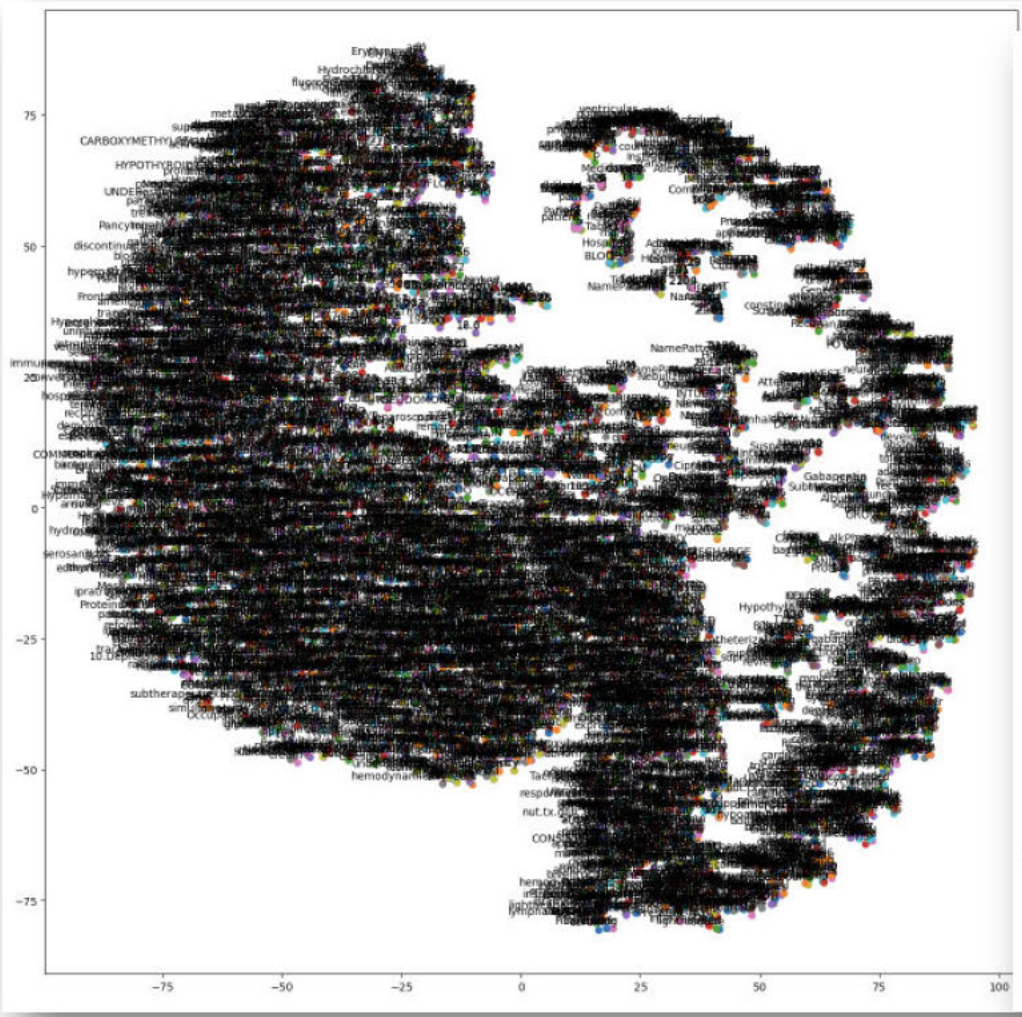
elevations V4-V6. Patient denies chest pain.

Here, we see a possible use case for tuning custom NER engines with MedSpacy's TargetRules

MedSpacy's word2vec/t-SNE plots

This used the same t-SNE plot code as for spacy/scispacy

Full Plot (n=7000)



Sample Plot (n=100)