MIMIC NLP

SUBMITTED BY: AMBIKA GARG

CONGESTIVE HEART FAILURE

Congestive Heart Failure (CHF), identified by ICD-9 code 428.0 in the MIMIC dataset, is a chronic and progressive condition in which the heart is unable to pump blood effectively to meet the body's needs. It often results from underlying conditions such as coronary artery disease, hypertension, or previous heart attacks. Patients with CHF typically experience symptoms like shortness of breath, fatigue, fluid retention, and swelling in the legs. Given its high prevalence and clinical importance, CHF is a critical condition to study in medical NLP research. Analyzing patient notes related to CHF can reveal patterns in diagnosis, treatment, and outcomes, making it an ideal focus for extracting medical entities and applying NLP techniques.

Overview

The objective of this report is to extract medical entities and analyze patient data using advanced Natural Language Processing (NLP) techniques, focusing specifically on clinical notes related to Congestive Heart Failure (CHF) from the MIMIC dataset.

The report includes the following components:

1. Dataset Preparation: The MIMIC clinical database was used, and records associated with Congestive Heart Failure were selected as the primary focus for analysis.
2. Named Entity Recognition (NER) with spaCy: spaCy's general-purpose NER pipeline was applied to extract standard entities using common web-based vocabularies.
3. NER with SciSpacy: Domain-specific models en_core_sci_md and bc5cdr were used to improve medical entity recognition, capturing biomedical and chemical/disease-related terms more accurately.
4. Word2Vec Embedding: Word embeddings were trained separately using the first 1000 and 2000 medical records. Entity extraction was performed using both spaCy and SciSpacy, and embeddings were generated to capture the contextual similarity of terms.
5. T-SNE Visualization: To compare the effectiveness of the NLP pipelines, dimensionality reduction using T-SNE was performed. Visual plots were generated for:

Dataset Preparation

```
SELECT *
FROM NOTEEVENTS n
JOIN DIAGNOSES_ICD d
ON n.SUBJECT_ID =
d.SUBJECT_ID
WHERE d.ICD9_CODE = '4280'
AND n.CATEGORY = 'Discharge
summary'
```

Parent Dataset:

- **MIMIC-III Clinical Database (Medical Information Mart for Intensive Care)**

Tables Used:

- **NOTEEVENTS: Contains free-text clinical notes**
- **DIAGNOSES_ICD: Contains diagnosis codes (ICD-9 format)**

Data Filtering & Join:

- **Joined NOTEEVENTS with DIAGNOSES_ICD on SUBJECT_ID and HADM_ID**
- **Filtered records where ICD9_CODE = 4280 (Congestive Heart Failure)**

Note Type Selection:

- **From ~15 categories in NOTEEVENTS, only Discharge summaries are used.**
- **Why Discharge Summary was selected?**
  - **Most comprehensive note type covering the full hospital stay.**
  - **Includes key sections: diagnosis, procedures, medications, outcomes.**
  - **Rich in medical entities—ideal for NLP tasks like NER.**
  - **More structured and informative than other note types.**
  - **Best suited for analyzing disease progression and treatment, especially for CHF.**

Discharge Summary (Formatted)
Admission Date: [2162-3-3]
Discharge Date: [2162-3-25]
Date of Birth: [2080-1-4]
Sex: M
Service: MEDICINE
Allergies: Patient recorded as having No Known Allergies to Drugs
Attending: [First Name3 (LF) 1828]
Chief Complaint:
Mr. [Known lastname 1829] was seen at [Hospital1 18] after a mechanical fall from a height of 10 feet. CT scan noted unstable fracture of C6-7 & posterior elements.
Major Surgical or Invasive Procedure:
1.Anterior cervical osteotomy, C6-C7, with decompression and excision of ossification of the posterior longitudinal ligament.
2.Anterior cervical deformity correction.
3.Interbody reconstruction.
4.Anterior cervical fusion, C5-C6-C7.
5.Plate instrumentation, C5-C6-C7.
6.Cervical laminectomy C6-C7, T1.
7.Posterior cervical arthrodesis C4-T1.
8.Cervical instrumentation C4-T1.
9.Arthrodesis augmentation with autograft, allograft, and demineralized bone matrix.
History of Present Illness:
Mr. [Known lastname 1829] is an 82-year-old male who had a slip and fall of approximately 10 feet from a balcony. He was ambulatory at the scene.
He presented to the ED here at [Hospital1 18]. CT scan revealed unstable C-spine fracture. He was intubated secondary to agitation.
Patient admitted to trauma surgery service.
Past Medical History:
- Coronary artery disease s/p CABG
- CHF
- HTN
- AICD
- Atrial fibrillation
- Stroke

Social History:
Patient recently discharged from [Hospital1] for severe depression.
Family reports patient was very sad and attempted to kill himself by wrapping a telephone cord around his neck.
Lives with his elderly wife, worked as a chemist in [Country 532].
Family History:
Non-contributory
Physical Exam:
Pre-surgical exam not obtained (patient intubated and sedated).
Post-surgical exam (TSICU per surgery team):
- Breathing without assistance, NAD
- Vitals: T 97.5, HR 61, BP 145/67, RR 22, SaO2 98
- A-fib, rate controlled
- Abdomen: Soft, non-tender
- Anterior/Posterior cervical incisions
- Edema in all four extremities, no facial edema
- Able to grossly move all four extremities, neurointact to light touch
- Distal pulses weakly intact

Mp with primary care physician as needed

edicine Consult:
VS: T 98.9, BP 142/70, HR 61, RR 20, SpO2 96% RA
I/O: BM yesterday 220/770
General: Awake, calm, cooperative
Neck: c-collar removed
CV: Irregular, normal S1, S2, no murmurs
Lungs: Clear to auscultation anteriorly
Abdomen: Obese, Soft, NTND
Extremities: Trace bilateral lower extremity edema, 1+ upper extremity edema
Neuro: Cognition impaired
Pertinent Results:
Admission Labs:
WBC 8.4
RBC 4.43*
Hgb 11.9*
Hct 38.6*
MCHC 30.9*
INR 1.9*
Glucose 121*
CK 183*
Phosphorus 2.0*
Lactate 2.3*
(others omitted for brevity)
Radiology:
CT C-Spine:
Fracture of C6
Anterior widening at C6-C7 disc
Prevertebral hematoma
Degenerative disease
Ossification of ligaments
CT Abdomen/Pelvis:
No acute injuries
Pleural pseudotumors
Pancreatic lesion (possible pseudocyst)
Bilateral renal cysts
Head CT:
No acute intracranial hemorrhage
Cerebellar encephalomalacia (old infarct)
Nasal bone fractures
Other Imaging (selected):
CT Pelvis, Chest X-Ray, CT Head follow-up
Upper extremity ultrasound: DVT in right arm veins
Swallow study: Aspiration of both thin and puree liquids
Brief Hospital Course:
Mr. [Known lastname 1829] fell from ~10 feet.
CT scans showed C6-C7 fracture. Underwent:
[2162-3-4] Anterior cervical decompression/fusion
[2162-3-5] Posterior cervical arthrodesis

Spacy

- **Use en_core_web_sm model to extract named entities from the first 1000 medical notes.**
- **en_core_web_sm: It is lightweight and fast, making it suitable for quick tasks or resource-constrained environments, but it may miss domain-specific entities in clinical or biomedical texts.**

```
import spacy

nlp_spacy = spacy.load('en_core_web_sm')

doc = []

for text in df_4280_discharge_summary_sample:
  doc.append(nlp_spacy(text))

congestive_heart_failure_entities = []

for i in range(len(doc)):
  for ent in doc[i].ents:
    entities = (ent.text, ent.start_char, ent.end_char, ent.label_)
    congestive_heart_failure_entities.append(entities)

congestive_heart_failure_entities = pd.DataFrame(congestive_heart_failure_entities, columns=['text', 'start_char', 'end_char', 'label'])
```

Spacy

List top 10 Entity text and Labels from Spacy

```
# Top 10 Entity text
congestive_heart_failure_entities[congestive_heart_failure_entities['label']!='CARDINAL'].value_counts('text').head(10)
```

text	count
Tablet	6071
daily	2254
First	1961
Tablet PO DAILY	1640
MD	1461
Tablet PO	1300
un	1250
Tablet(s	1076
CT	1055
PO	1041

dtype: int64

```
# Top 10 Entity Label
congestive_heart_failure_entities[congestive_heart_failure_entities['label']!='CARDINAL'].value_counts('label').head(10)
```

label	count
ORG	61542
DATE	40710
PERSON	18668
GPE	9189
TIME	5143
PERCENT	4963
PRODUCT	3109
ORDINAL	2806
QUANTITY	2316
NORP	1704

Spacy

Named Entity Recognition for the First Medical Note

As expected, the results were not sufficient for medical applications, as the model failed to accurately capture clinical and domain-specific terms.

SciSpacy

- ScispaCy is a SpaCy extension tailored for biomedical and scientific text.
- Offers a variety of pretrained models for tasks like Named Entity Recognition (NER) and tokenization.

Models used in the report:

1. **en_core_sci_md:**
   a. General-purpose biomedical model.
   b. Trained on a broad range of scientific literature (PubMed abstracts).

2. **en_ner_bc5cdr_md:**
   a. Specialized NER model trained specifically to identify:
      i. Diseases

Model: en_core_sci_md

Scispacy with en_core_sci_md

```
[ ] import en_core_sci_md
    scispacy_base_nlp = en_core_sci_md.load()

    doc_core_sci_md = []

    for text in df_4280_discharge_summary_sample:
      doc_core_sci_md.append(scispacy_base_nlp(text))
```

```
/usr/local/lib/python3.11/dist-packages/spacy/language.py:2195: FutureWarning: Possible set union at position 6328
  deserializers["tokenizer"] = lambda p: self.tokenizer.from_disk(  # type: ignore[union-attr]
```

```
[ ] scispacy_entities_base_model_entities = []

    for i in range(len(doc)):
      for ent in doc[i].ents:
        entities = (ent.text, ent.start_char, ent.end_char, ent.label_)
        scispacy_entities_base_model_entities.append(entities)

    scispacy_entities_base_model_entities = pd.DataFrame(scispacy_entities_base_model_entities, columns=['text', 'start_char', 'end_char', 'label'])
```

Model: en_ner_bc5cdr_md

```
import en_ner_bc5cdr_md
scispacy_en_ner_bc5cdr_md_nlp = en_ner_bc5cdr_md.load()

en_ner_bc5cdr_md_doc = []
for text in df_4280_discharge_summary_sample:
  en_ner_bc5cdr_md_doc.append(scispacy_en_ner_bc5cdr_md_nlp(text))
```

```
/usr/local/lib/python3.11/dist-packages/spacy/util.py:922: UserWarning: [W095] Model 'en_ner_bc5cdr_md' (0.5.4) was trained with spaCy v3.7.4 and may
  warnings.warn(warn_msg)
/usr/local/lib/python3.11/dist-packages/spacy/language.py:2233: FutureWarning: Possible set union at position 6328
  deserializers["tokenizer"] = lambda p: self.tokenizer.from_disk(  # type: ignore[union-attr]
```

```
[ ] en_ner_bc5cdr_md_text = []
    for doc in en_ner_bc5cdr_md_doc:
      en_ner_bc5cdr_md_text.append(doc.text)
```

```
[ ] scispacy_entities_bc5cdr_model_entities = []

    for i in range(len(en_ner_bc5cdr_md_doc)):
      for ent in en_ner_bc5cdr_md_doc[i].ents:
        entities = (ent.text, ent.start_char, ent.end_char, ent.label_)
        scispacy_entities_bc5cdr_model_entities.append(entities)

    scispacy_entities_bc5cdr_model_entities = pd.DataFrame(scispacy_entities_bc5cdr_model_entities, columns=['text', 'start_char', 'end_char', 'label'])
```

SciSpacy

List Top 10 Entities and Text from both the models

Model: en_core_sci_md:

```
[ ] # Top 10 Entity text
scispacy_entities_base_model_entities[scispacy_entities_base_model_entities['label']!='CARDINAL'].value_counts('text').head(10)
```

text	count
Tablet	11367
patient	8624
day	7209
PO	6901
Sig	4165
daily	2998
Daily	2778
admission	2426
Patient	2264
days	2146

dtype: int64

```
[ ] # Top 10 Entity label
scispacy_entities_base_model_entities[scispacy_entities_base_model_entities['text']!='CARDINAL'].value_counts('label').head(10)
```

label	count
ENTITY	567918

dtype: int64

Model: en_ner_bc5cdr_md

```
# Top 10 Entity text
scispacy_entities_bc5cdr_model_entities[scispacy_entities_bc5cdr_model_entities['label']!='CARDINAL'].value_counts('text').head(10)
```

text	count
Tablet	11367
patient	8624
day	7209
PO	6901
Sig	4165
pain	3040
daily	2998
Daily	2778
admission	2426
Patient	2264

dtype: int64

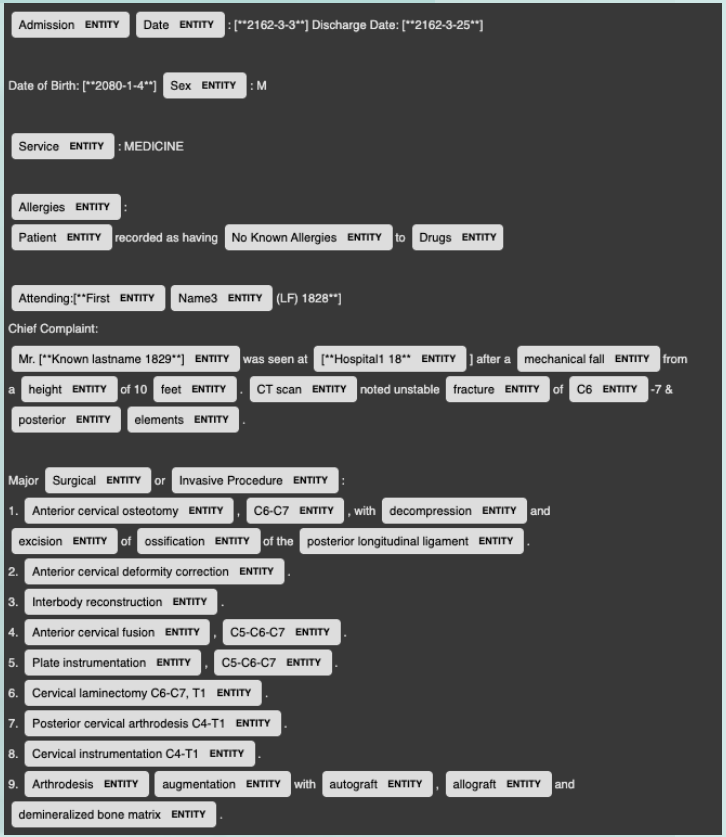
```
[ ] # Top 10 Entity label
scispacy_entities_bc5cdr_model_entities[scispacy_entities_bc5cdr_model_entities['text']!='CARDINAL'].value_counts('label').head(10)
```

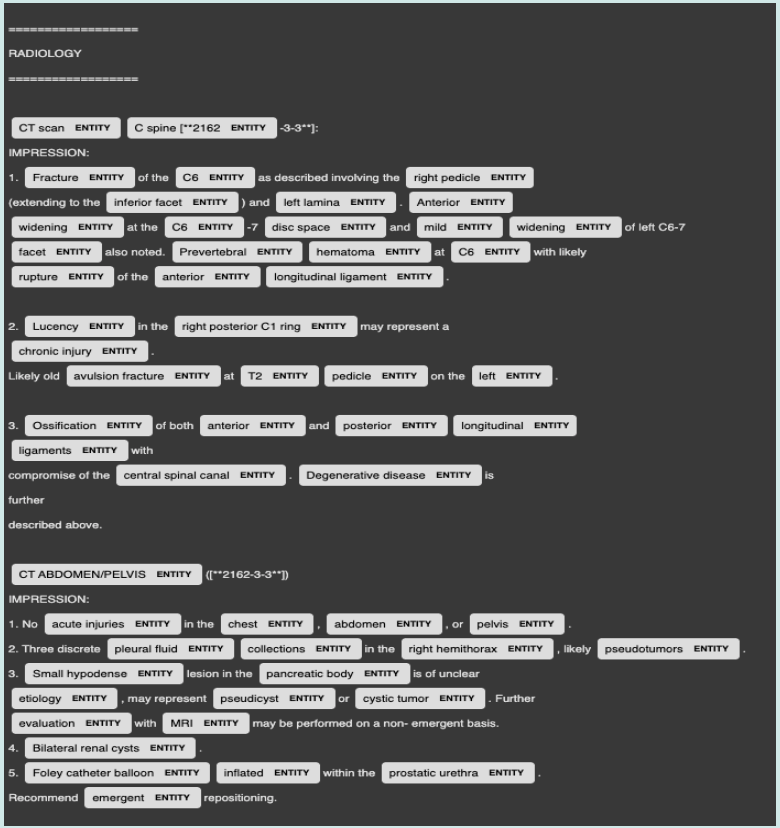
label	count
ENTITY	567918
DISEASE	72155
CHEMICAL	56153

dtype: int64

SciSpacy

Name Entity Recognition using SciSpacy model.





Comparison Between Spacy and SciSpacy

Criteria	Spacy	SciSpacy
Domain Adaptation	Poor	Strong (trained on biomedical corpora)
Entity Coverage	Limited to general entities (e.g., PERSON, ORG, DATE)	Specialized (e.g., diseases, chemicals, anatomy)
Model Examples	`en_core_web_sm` returns irrelevant entities on clinical text	`en_ner_bc5cdr_md` identifies *Diseases* and *Chemicals* well
Result from Analysis	No disease/chemical recognized	Disease and Entity recognized.

Word2Vec

Word2Vec turns words into **dense vector representations** (e.g., 300-dimensional vectors) such that **words with similar meanings are close together** in vector space.

Word2Vec uses a shallow neural network and learns embeddings in an **unsupervised** way. It has two main architectures:

1. **CBOW (Continuous Bag of Words)**:
   Predicts the **current word** based on its **context** (surrounding words).
   Example: Given "The cat __ on the mat", it predicts "sat".

2. **Skip-gram**:
   Predicts **context words** given the **current word**.
   Example: Given "sat", it tries to predict "The", "cat", "on", etc.

Word2Vec

- Use en_ner_bc5cdr_md to build the corpus from the medical notes related to diabetes.
- Preprocess the notes to remove stop_words, punctuations and keep only Alphabetic tokens
- Train 2 Word2Vec model
  - W2v_model_1000 (from first 1000 notes)
  - W2v_model_2000 (from first 2000 notes)

```
[ ] import en_ner_bc5cdr_md
    import re

    scispacy_en_ner_bc5cdr_md_nlp = en_ner_bc5cdr_md.load()

    # Tokenize and prepare data for scispacy bc5cdr
    # Clean the dataset for text
    def clean_medical_note(text):
      # Step 1: Remove PHI placeholders like [** ... **]
      text = re.sub(r'\[\*\*.*?\*\*\]', ' ', text)

      # Step 2: Process with spaCy
      doc = scispacy_en_ner_bc5cdr_md_nlp(text.lower())

      tokens = [
            token.text.lower() for token in doc
            if (token.is_alpha or token.is_digit)  # keep only alphabetic tokens
            and token not in scispacy_en_ner_bc5cdr_md_nlp.Defaults.stop_words  # remove stopwords
            and not token.is_punct                    # remove punctutations
        ]

      doc = scispacy_en_ner_bc5cdr_md_nlp(" ".join(tokens))
      # Flatten the list of entities
      return [ent.text for ent in doc.ents]

    tokenized_notes = [clean_medical_note(note) for note in df_4280_discharge_summary_sample_2000]
```

```
from gensim.models import Word2Vec

# Configure the Word2Vec model
vector_size = 200
model_bc5cdr_2000 = Word2Vec(sentences=tokenized_notes, vector_size=vector_size, window=10, min_count=10, workers=4)
model_bc5cdr_2000.wv.key_to_index.keys()
```

Show hidden output

```
[ ] # Train the model
    model_bc5cdr_2000.train(tokenized_notes, total_examples=model_bc5cdr_2000.corpus_count, epochs=10)
```

```
WARNING:gensim.models.word2vec:Effective 'alpha' higher than previous training cycles
(1377459, 2041260)
```

```
[ ] model_bc5cdr_1000 = Word2Vec(sentences=tokenized_notes[:1000], vector_size=vector_size, window=10, min_count=30, workers=4)
    model_bc5cdr_1000.wv.key_to_index.keys()
```

Show hidden output

```
[ ] # Train the model
    model_bc5cdr_1000.train(tokenized_notes[:1000], total_examples=model_bc5cdr_1000.corpus_count, epochs=10)
```

Word2Vec Comparison

- Find the most similar words to “Pain” and “Diarrhea”
- Word2vec_model_2000 outperforms word2vec_model_1000
- W2vec_model_1000 identifies the complications and treatments related to diabetes

Word2vec_model_2000 Model

```
model_bc5cdr_2000.wv.most_similar("pain")
```

```
[('dilaudid', 0.7675082683563232),
 ('oxycodone', 0.7077162861824036),
 ('contin', 0.6500534415245056),
 ('morphine', 0.6415982246398926),
 ('left shoulder pain', 0.6307831406593323),
 ('tylenol', 0.6191858053207397),
 ('chronic pain', 0.588632345199585),
 ('chronic back pain', 0.581362247467041),
 ('hydromorphone', 0.5738732814788818),
 ('tramadol', 0.5682240724563599)]
```

Word2vec_model_1000 Model

```
model_bc5cdr_1000.wv.most_similar("pain")
```

```
[('dilaudid', 0.7781369090080261),
 ('oxycodone', 0.7328147292137146),
 ('nausea', 0.6958606839179993),
 ('tylenol', 0.6638454794883728),
 ('zofran', 0.6585942506790161),
 ('constipation', 0.6004403829574585),
 ('lidocaine', 0.5629581809043884),
 ('swelling', 0.560228705406189),
 ('sl', 0.5559645891189575),
 ('morphine', 0.5477560758590698)]
```

T-SNE

NE Highlight Plot

is plot highlights the key word
Red and other important words
yellow.

r Analysis I've created the
lowing plots:

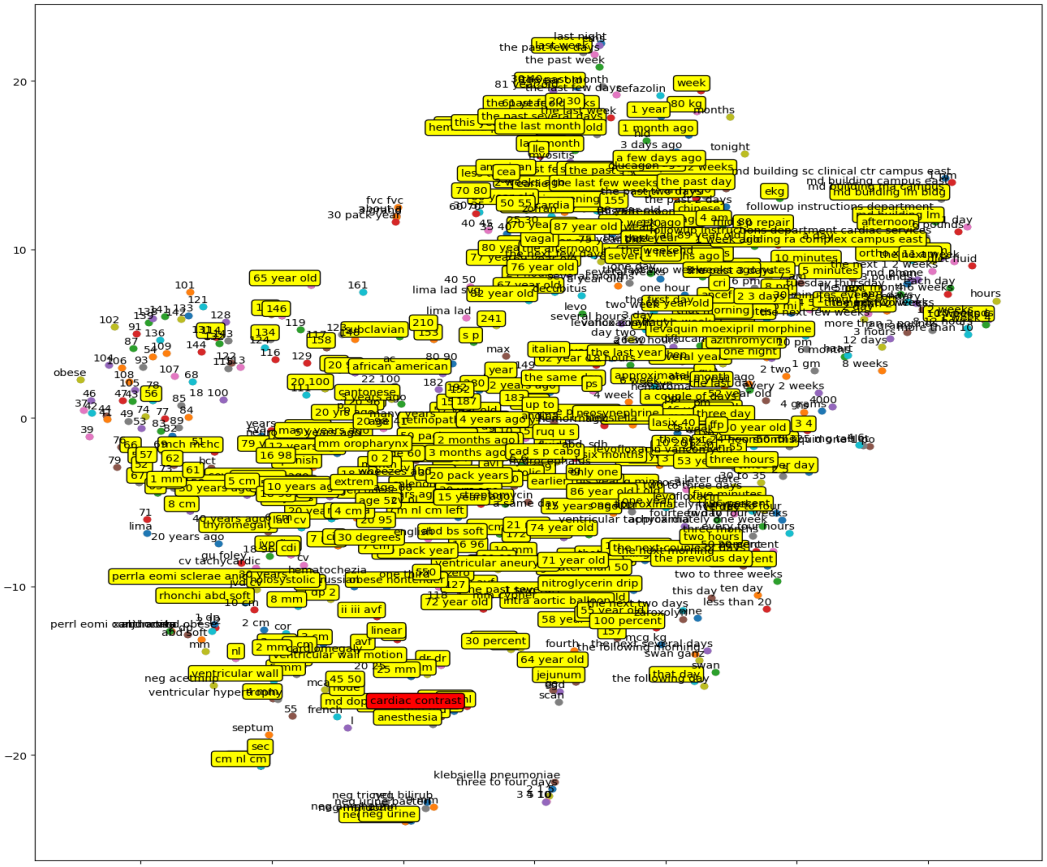
1. Using Spacy, Model: en_core_web_sm, First 2000 records
2. Using Scispacy, Model: en_ner_bc5cdr_md_text, First 1000 records
3. Using Scispacy, Model: en_ner_bc5cdr_md_text, First 2000 records

```
def tsne_plot_highlight(model, words, key_word, highlighted_words, preTrained=False):
  labels = []
  tokens = []

  for word in words:
    if preTrained:
      tokens.append(model[word])
    else:
      tokens.append(model.wv[word])
    labels.append(word)

  tsne_model = TSNE(perplexity=min(30, len(words) - 1), early_exaggeration=12, n_components=2, init='pca', n_iter=1000, random_state=23)
  tokens = np.array(tokens)
  new_values = tsne_model.fit_transform(tokens)

  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)):
    if labels[i] in highlighted_words:
      plt.scatter(x[i],y[i], c='black')
      plt.annotate(labels[i],
                   xy = (x[i], y[i]),
                   xytext = (5, 2),
                   textcoords = 'offset points',
                   ha = 'right',
                   va = 'bottom',
                   bbox = dict(boxstyle = 'round,pad=0.3', facecolor = 'yellow', edgecolor = 'black', lw=1))
    elif labels[i] == key_word:
      plt.scatter(x[i],y[i], c='black')
      plt.annotate(labels[i],
                   xy = (x[i], y[i]),
                   xytext = (5, 2),
                   textcoords = 'offset points',
                   ha = 'right',
                   va = 'bottom',
                   bbox = dict(boxstyle = 'round,pad=0.3', facecolor = 'red', edgecolor = 'black', lw=1))
    else:
      plt.scatter(x[i],y[i], c='gray')
      plt.annotate(labels[i],
                   xy=(x[i], y[i]),
                   xytext=(5, 2),
                   textcoords='offset points',
                   ha='right',
                   va='bottom')
    plt.scatter(x[i],y[i])
```

T-SNE

NE Highlight Plot 1
ing Spacy,
odel: en_core_web_sm
es, First 2000 records

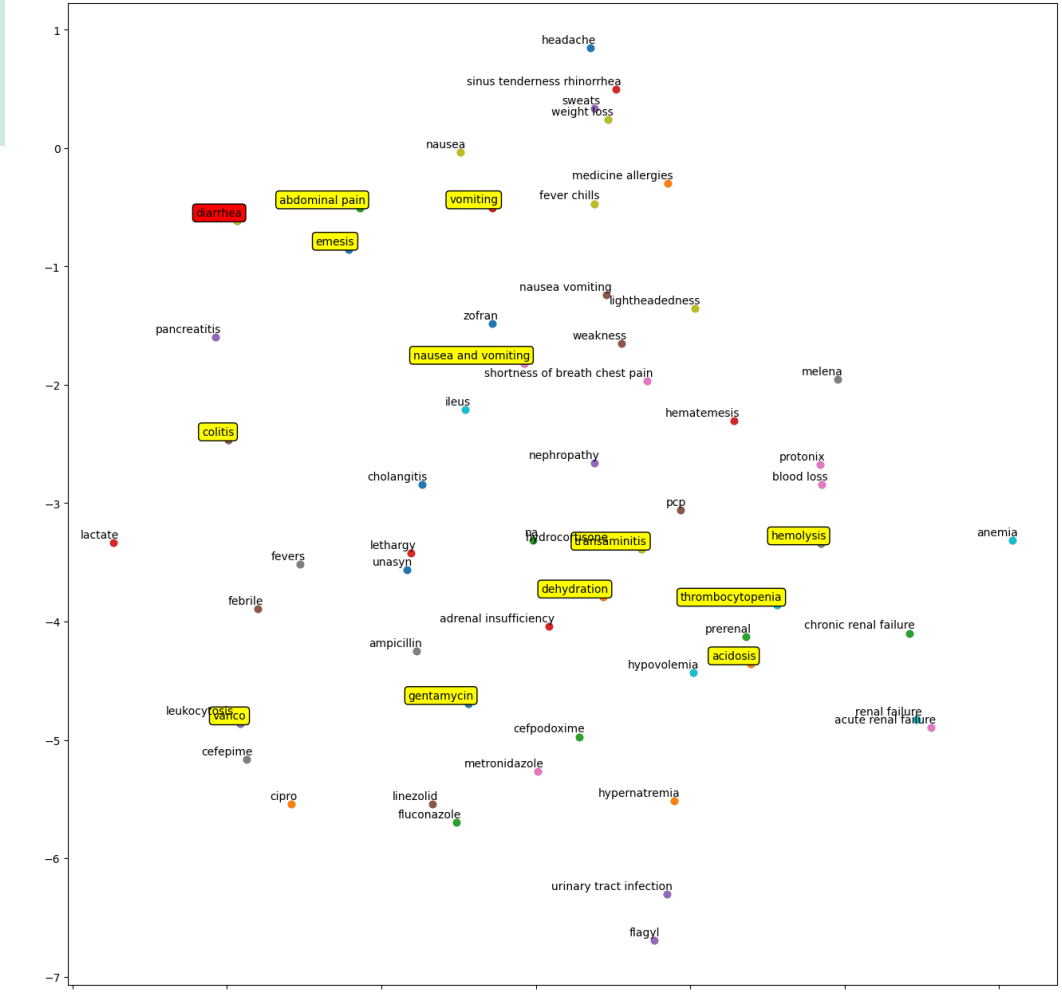
- **This scatter plot contains the word that have at least 0.4 similarity to “cardiac_arrest”**
- **Highlights the word with atleast 0.70 similarity to “cardiac_arrest” in yellow**
- **As expected, there are few highlighted words but they are not meaningfully close.**



T-SNE

-SNE Highlight Plot 2
sing SciSpacy,
lodel: bc5cdr
ses, First 1000 records

- **The plot uses the stronger model than spacy and weaker than bc5cdr(2000)**
- **This scatter plot contains the word that have at least 0.4 similarity to “diarrhea”**
- **Highlights the word with atleast 0.70 similarity to “diarrhea” in yellow**
- **As expected, there are few highlighted words and they are somewhat meaningfully close.**

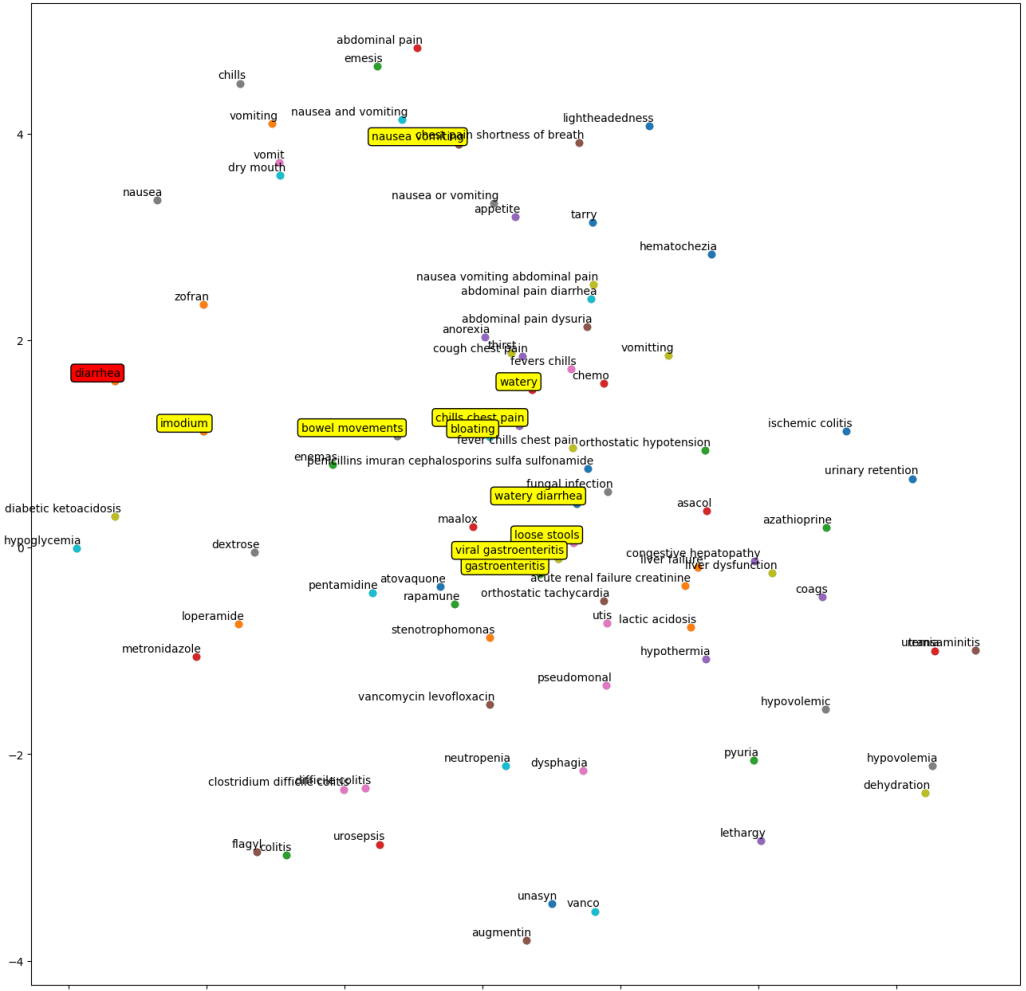


T-SNE

NE Highlight Plot 3
ing SciSpacy,
odel: bc5cdr
es, First 2000 records

e plot uses the strong
del bc5cdr(2000)

- **This scatter plot contains the word that have at least 0.4 similarity to “diarrhea”**
- **Highlights the word with atleast 0.70 similarity to “diarrhea” in yellow**
- **As expected, there are few highlighted words and they are meaningfully closer and clustered together than SciSpacy(1000).**



Clinical NLP Pipeline SciSpacy + MedSpacy

Pipeline Steps

1. **Named Entity Recognition (NER) with scispaCy**
   - Uses en_ner_bc5cdr_md model
   - Detects diseases and chemicals from biomedical text
   - Example entities: *pneumonia*, *Lasix*, *CAD*

2. **medSpaCy Pipeline Setup**
   - Loads clinical components like ConTextComponent and Sectionizer
   - Disables medSpaCy's built-in NER to avoid conflicts with scispaCy

3. **Transfer Entities to medSpaCy Doc**
   - Maps scispaCy-detected entities into a medSpaCy Doc
   - Preserves start/end character positions and entity labels

4. **Contextual Analysis (medSpaCy)**
   Each entity is annotated with:
   - is_negated: e.g., *no chest pain* → negated
   - is_historical: e.g., *history of pneumonia*
   - is_family: e.g., *family history of CAD*
   - section_category: e.g., *past_medical_history*, *medications*

5. **Output: Entities and context information stored in a pandas DataFrame**

Top Affirmed, Patient-specific Mentions:

text	count
pain	4117
CHF	2303
edema	2277
chest pain	2180
pneumonia	1892
CAD	1763
Lasix	1755
shortness of breath	1644
Aspirin	1581
p.o	1562

dtype: int64

Most Common Sections:

section	count
hospital_course	60037
medications	51764
history_of_present_illness	41086
past_medical_history	21657
labs_and_studies	20933
observation_and_plan	20225
physical_exam	12061
patient_instructions	10678
diagnoses	3722
allergy	3487

dtype: int64