# Named Enitty Recognition of Chemical Compounds from Chemical Dataset

19AE3AI04
Sai Sreeja Ramishetti

Design Lab Project
Manjira Sinha

# WHAT IS NER?

Named-entity recognition (NER) (also known as (named) entity identification, entity chunking, and entity extraction) is a subtask of information extraction that seeks to locate and classify named entities mentioned in unstructured text into pre-defined categories such as person names, organizations, locations, medical codes, time expressions, quantities, monetary values, percentages, etc.

**Example**

Jim bought 300 shares of Acme Corp. in 2006.

Names of entities

[Jim]Person bought 300 shares of [Acme Corp.]Organization in [2006]Time.

In this example, a person name consisting of one token, a two-token company name and a temporal expression have been detected and classified.

# WORKFLOW FOR NER

## 1 Data preparation

Obtain a chemical dataset in a suitable format, such as a table or a text file, and preprocess the data as needed. This may include removing extraneous information or formatting the text to make it easier to parse.

## 2 Training data

Create a set of training data that includes examples of chemical compound names, along with their associated entity labels. This can be done manually by annotating the text with entity labels, or automatically using existing labeled datasets.

## 3 Model selection & training

Choose a suitable NER model, such as a rule-based model, a statistical model, or a deep learning model, and train it on the training data.

# FLOWCHART

## 4 Evaluation

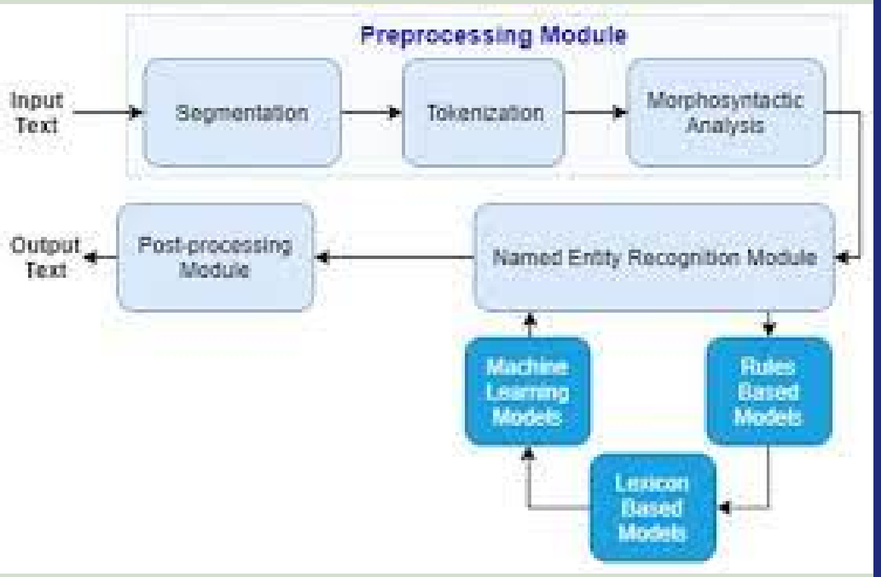
Evaluate the performance of the trained model on a test dataset, using metrics such as precision, recall, and F1 score.

## 5 Deployment

Deploy the trained model to perform NER on new chemical datasets, and use the extracted entity information for downstream tasks such as data analysis or visualization.

# PLATFORMS USED

- GATE supports NER across many languages and domains out of the box, usable via a graphical interface and a Java API.
- SpaCy features fast statistical NER as well as an open-source named-entity visualizer.
- OpenNLP includes rule-based and statistical named-entity recognition.
- ChemDataExtractor, ChemSpot, and ChemicalTagger Tools to be used




# Problem Statement

Create an NER model that can efficiently and accurately extract chemical entity information from a wide range of chemical datasets, with high reproducibility and scalability.

# Aim

The problem statement for named entity recognition (NER) of chemical compounds from a chemical dataset is to automatically identify and extract the names of chemical compounds mentioned in the text. The chemical dataset may contain various types of text,

# Problems Faced

Diversity and complexity of chemical nomenclature.
Chemical names can vary in length, structure, and format
May include chemical symbols, abbreviations, and numerical values.
They can be ambiguous, with different compounds having similar names.

**Solution:**

NER models need to be trained on large and diverse chemical datasets and should incorporate domain-specific knowledge and rules.

# SAMPLE INPUT

3-Isobutyl-5-methyl-1-(oxetan-2-ylmethyl)-6-[(2-oxoimidazolidin-1-yl)methyl]thieno[2,3-d]pyrimidine-2,4(1H,3H)-dione (racemate)
813 mg (1.84 mmol) of the compound from Example 243A were dissolved in 40 ml of dioxane, and 461 mg (2.76 mmol) of CDI were added. The mixture was stirred at RT for 16 h. The reaction solution was then concentrated on a rotary evaporator. The residue was dissolved in 15 ml of DMSO and this solution was purified by means of preparative HPLC (Method 14). Combination of the product fractions and freeze-drying gave 383 mg (42% of theory) of the title compound

# PRE TRAINED MODELS

Example 194 3 ChemicalDrugs -
ChemicalDrugs Iso ChemicalDrugs but ChemicalDrugs yl ChemicalDrugs -5 ChemicalDrugs -
ChemicalDrugs met ChemicalDrugs hyl ChemicalDrugs -1-
ChemicalDrugs ( ChemicalDrugs o ChemicalDrugs xe ChemicalDrugs tan ChemicalDrugs -
ChemicalDrugs 2 ChemicalDrugs -
ChemicalDrugs y ChemicalDrugs lme ChemicalDrugs thy ChemicalDrugs l)-6-
ChemicalDrugs [ ChemicalDrugs ( 2 ChemicalDrugs -
ChemicalDrugs o ChemicalDrugs xo ChemicalDrugs imi ChemicalDrugs da ChemicalDrugs zo Chemic
1-
ChemicalDrugs y ChemicalDrugs l)met ChemicalDrugs hyl ChemicalDrugs ] ChemicalDrugs thi Chem
d ] ChemicalDrugs p ChemicalDrugs yri ChemicalDrugs mid ChemicalDrugs ine-
2,4 ChemicalDrugs (1H,3H)- dio ChemicalDrugs ne ChemicalDrugs (racemate) 813 mg (1.84
mmol) of the compound from Example 243A were dissolved in 40 ml of
dio ChemicalDrugs xan ChemicalDrugs e ChemicalDrugs , and 461 mg (2.76 mmol) of CDI


```
[
  {
    "entity_group": "ChemicalDrugs",
    "score": 0.9114004373550415,
    "word": "3",
    "start": 12,
    "end": 13
  },
  {
    "entity_group": "ChemicalDrugs",
    "score": 0.8697298765182495,
    "word": "-",
    "start": 13,
    "end": 14
  },
  {
    "entity_group": "ChemicalDrugs",
    "score": 0.9986028075218201,
    "word": "iso",
    "start": 14,
    "end": 17
  },
  {
    "entity_group": "ChemicalDrugs",
    "score": 0.9990529417991638,
    "word": "##but",
    "start": 17,
    "end": 20
  },
  {
    "entity_group": "ChemicalDrugs",
    "score": 0.9991874098777771,
    "word": "##yl",
    "start": 20,
    "end": 22
  },
```


pucpr-Clinicalnerpt-Chemical

# PRE TRAINED MODELS

Example 194 3 CHEM - CHEM I CHEM so CHEM but CHEM yl CHEM - CHEM 5- CHEM me CHEM th CHEM yl CHEM - CHEM 1 CHEM - ( CHEM ox CHEM e CHEM tan CHEM -2-ylme CHEM th CHEM yl CHEM )- CHEM 6 CHEM - [ CHEM ( CHEM 2 CHEM - CHEM ox CHEM o CHEM im CHEM idazol CHEM idin-1- yl) CHEM me CHEM th CHEM yl CHEM ] CHEM th CHEM i CHEM eno CHEM [ CHEM 2, CHEM 3- CHEM d CHEM ] CHEM py CHEM rim CHEM id CHEM ine CHEM - CHEM 2, CHEM 4(1 CHEM H, CHEM 3 CHEM H)- CHEM di CHEM one CHEM ( ra CHEM ce CHEM mate CHEM ) 813 mg (1.84 mmol) of the compound from Example 243A were dissolved in 40 ml of di CHEM ox CHEM ane CHEM , and 461 mg (2.76 mmol) of CDI were added. The mixture was stirred at RT for 16 h. The reaction solution was then concentrated on a rotary evaporator. The residue was dissolved in 15 ml of D CHEM MS CHEM O CHEM and this solu tion CHEM was puri PROC fied PROC by means of prepara PROC tive PROC H PROC PL PROC C PROC (Method 14). Combination of the product fractions and fre PROC ez PROC e-drying gave 383 mg (42% of theory) of the title compound


```
[
  {
    "entity_group": "CHEM",
    "score": 0.992956280708313,
    "word": "3",
    "start": 12,
    "end": 13
  },
  {
    "entity_group": "CHEM",
    "score": 0.9947956204414368,
    "word": "-",
    "start": 13,
    "end": 14
  },
  {
    "entity_group": "CHEM",
    "score": 0.9976467490196228,
    "word": "I",
    "start": 14,
    "end": 15
  },
  {
    "entity_group": "CHEM",
    "score": 0.9978088736534119,
    "word": "so",
    "start": 15,
    "end": 17
  },
  {
    "entity_group": "CHEM",
    "score": 0.9947524070739746,
    "word": "but",
    "start": 17,
    "end": 20
  },
  {
    "entity_group": "CHEM",
    "score": 0.9965983033180237,
    "word": "yl",
    "start": 20,
```


lcampillos/roberta-es-clinical-trials-ner

# PRE TRAINED MODELS

Example 194 3-Isobutyl-5-methyl-1-(oxetan-2-ylmethyl)-6-[(2-oxoimidazolidin-1-yl)methyl]thieno[2,3-d]pyrimidine-2,4(1H,3H)-dione CHEMICAL (racemate) 813 mg (1.84 mmol) of the compound from Example 243A were dissolved in 40 ml of dioxane CHEMICAL, and 461 mg (2.76 mmol) of CD CHEMICAL I were added. The mixture was stirred at RT for 16 h. The reaction solution was then concentrated on a rotary evaporator. The residue was dissolved in 15 ml of DMSO CHEMICAL and this solution was purified by means of preparative HPLC (Method 14). Combination of the product fractions and freeze-drying gave 383 mg (42% of theory) of the title compound


```
[
  {
    "entity_group": "CHEMICAL",
    "score": 0.9999988675117493,
    "word": "3 - Isobutyl - 5 - methyl - 1 - ( oxetan - 2 - ylmethyl ) - 6 - [ (
    "start": 12,
    "end": 128
  },
  {
    "entity_group": "CHEMICAL",
    "score": 0.9975202679634094,
    "word": "dioxane",
    "start": 220,
    "end": 227
  },
  {
    "entity_group": "CHEMICAL",
    "score": 0.9848846197128296,
    "word": "CD",
    "start": 255,
    "end": 257
  },
  {
    "entity_group": "CHEMICAL",
    "score": 0.9830291867256165,
    "word": "DMSO",
    "start": 417,
    "end": 421
  }
]
```


alvaroalon2/biobert_chemical_ner

# CODE

+ Code + Text

```
[ ] x_train = []
    y_train = []
    x_dev = []
    y_dev = []
```

```
[ ] import os
```

```
[ ] for i in range(0,1500):
        ch = str(i)
        l = len(ch)
        while(l<4):
            ch = "0"+ch
            l=l+1
        xloc="train/"+ch+".txt"
        yloc="train/"+ch+".ann"
        isExisting = os.path.exists(xloc)
        if(isExisting==False):
            continue
        with open(xloc) as f1:
            lines1 = f1.readlines()
        with open(yloc) as f2:
            lines2 = f2.readlines()
        f1.close()
        f2.close()
        strr=""
        for j in lines1:
            strr=strr+j
        x_train.append(strr)
        y_train.append(lines2)
```

```
for i in range(0,1500):
    ch = str(i)
    l = len(ch)
    while(l<4):
        ch = "0"+ch
        l=l+1
    xloc="dev/"+ch+".txt"
    yloc="dev/"+ch+".ann"
    isExisting = os.path.exists(xloc)
    if(isExisting==False):
        continue
    with open(xloc) as f1:
        lines1 = f1.readlines()
    with open(yloc) as f2:
        lines2 = f2.readlines()
    f1.close()
    f2.close()
    strr=""
    for j in lines1:
        strr=strr+j
    x_dev.append(strr)
    y_dev.append(lines2)
```

```
[ ] print(len(x_train))
    print(len(x_dev))
```

```
900
600
```

```
[ ] y_train[0][3].split()
```

```
y_train[0][3].split()
```

```
['T3', 'YIELD_PERCENT', '563', '566', '42%']
```

```
[ ] lt = []
    for j in range(0,len(y_train[0])):
        lstt=y_train[0][j].split()
        lt.append((lstt[2],lstt[3],lstt[1]))
    lt
```

```
[('417', '421', 'OTHER_COMPOUND'),
 ('305', '309', 'TIME'),
 ('585', '599', 'REACTION_PRODUCT'),
 ('563', '566', 'YIELD_PERCENT'),
 ('255', '258', 'STARTING_MATERIAL'),
 ('298', '300', 'TEMPERATURE'),
 ('12', '139', 'REACTION_PRODUCT'),
 ('220', '227', 'SOLVENT'),
 ('8', '11', 'EXAMPLE_LABEL'),
 ('166', '192', 'STARTING_MATERIAL'),
 ('555', '561', 'YIELD_OTHER')]
```

```
[ ] training_data=[]
    dev_data = []

    for i in range(0,len(x_train)):
        mpp={}
        mpp['text']=x_train[i]
        mpp['entities']=[]
        for j in range(0,len(y_train[i])):
            lstt=y_train[i][j].split()
```

```
[ ]        for j in range(0,len(y_train[i])):
               lstt=y_train[i][j].split()
               mpp['entities'].append((int(lstt[2]),int(lstt[3]),lstt[1]))
           training_data.append(mpp)

       for i in range(0,len(x_dev)):
           mpp={}
           mpp['text']=x_dev[i]
           mpp['entities']=[]
           for j in range(0,len(y_dev[i])):
               lstt=y_dev[i][j].split()
               mpp['entities'].append((int(lstt[2]),int(lstt[3]),lstt[1]))
           dev_data.append(mpp)
```

```
print(len(training_data))
print(len(dev_data))
```

```
900
600
```

```
[ ] training_data[0]
```

```
{'text': 'Example 194\n3-Isobutyl-5-methyl-1-(oxetan-2-ylmethyl)-6-[(2-oxoimidazolidin-1-yl)methyl]thieno[2,3-d]pyrimidine-2,4(1H,3H)-dione (racemate)\n813 mg (1.84 mmol) of the compound from Example 243A were dissolved in 40 ml of dioxane, and 461 mg (2.76 mmol) of CDI were added. The mixture was stirred at RT for 16 h. The reaction solution was then concentrated on a rotary evaporator. The residue was dissolved in 15 ml of DMSO and this solution was purified by means of preparative HPLC (Method 14). Combination of the product fractions and freeze-drying gave 383 mg (42% of theory) of the title compound',
 'entities': [(417, 421, 'OTHER_COMPOUND'),
  (305, 309, 'TIME'),
 (12, 139, 'REACTION_PRODUCT'),
 (220, 227, 'SOLVENT'),
 (8, 11, 'EXAMPLE_LABEL'),
 (166, 192, 'STARTING_MATERIAL'),
 (555, 561, 'YIELD_OTHER')]}
```

```
[ ] from spacy.tokens import DocBin
    from tqdm import tqdm
    import spacy

    nlp = spacy.blank("en") # load a new spacy model
    doc_bin = DocBin()
```

```
from spacy.util import filter_spans

for training_example  in tqdm(training_data):
    text = training_example['text']
    labels = training_example['entities']
    doc = nlp.make_doc(text)
    ents = []
    for start, end, label in labels:
        span = doc.char_span(start, end, label=label, alignment_mode="contract")
        if span is None:
            print("Skipping entity")
        else:
            ents.append(span)
```

```python
from spacy.util import filter_spans

for training_example  in tqdm(training_data):
    text = training_example['text']
    labels = training_example['entities']
    doc = nlp.make_doc(text)
    ents = []
    for start, end, label in labels:
        span = doc.char_span(start, end, label=label, alignment_mode="contract")
        if span is None:
            print("Skipping entity")
        else:
            ents.append(span)
    filtered_ents = filter_spans(ents)
    doc.ents = filtered_ents
    doc_bin.add(doc)

doc_bin.to_disk("train.spacy")
```

```
Skipping entity
Skipping entity
Skipping entity
Skipping entity
Skipping entity
Skipping entity
Skipping entity
Skipping entity
Skipping entity
Skipping entity
Skipping entity
Skipping entity
```

```python
from spacy.tokens import DocBin
doc_bin2 = DocBin()
from spacy.util import filter_spans

for training_example  in tqdm(dev_data):
    text = training_example['text']
    labels = training_example['entities']
    doc = nlp.make_doc(text)
    ents = []
    for start, end, label in labels:
        span = doc.char_span(start, end, label=label, alignment_mode="contract")
        if span is None:
            print("Skipping entity")
        else:
            ents.append(span)
    filtered_ents = filter_spans(ents)
    doc.ents = filtered_ents
    doc_bin2.add(doc)
```

```
 50%|
Skipping entity
Skipping entity
```

```
[ ] len(doc_bin2)
```

```
600
```

```
[ ] len(doc_bin)
```

```
900
```

```
[ ] doc_bin2.to_disk("dev.spacy")
```

```
!python -m spacy init fill-config base_config.cfg config.cfg
```

```
✓ Auto-filled config with all values
✓ Saved config
config.cfg
You can now add your data and train your pipeline:
python -m spacy train config.cfg --paths.train ./train.spacy --paths.dev ./dev.spacy
```

```
[ ] !python -m spacy train config.cfg --output ./ --paths.train ./train.spacy --paths.dev ./dev.spacy
```

```
ℹ Saving to output directory: .
ℹ Using CPU
ℹ To switch to GPU 0, use the option: --gpu-id 0

========================== Initializing pipeline ==========================
[2023-05-01 14:05:30,902] [INFO] Set up nlp object from config
[2023-05-01 14:05:30,907] [INFO] Pipeline: ['tok2vec', 'ner']
[2023-05-01 14:05:30,909] [INFO] Created vocabulary
[2023-05-01 14:05:31,731] [INFO] Added vectors: en_core_web_lg
[2023-05-01 14:05:32,636] [INFO] Finished initializing nlp object
```

```
[2023-05-01 14:05:31,731] [INFO] Added vectors: en_core_web_lg
[2023-05-01 14:05:32,636] [INFO] Finished initializing nlp object
[2023-05-01 14:05:37,733] [INFO] Initialized pipeline components: ['tok2vec', 'ner']
✓ Initialized pipeline
```

```
[ ] nlp_ner = spacy.load("model-best")
```

```
doc = nlp_ner('Example 194\n3-Isobutyl-5-methyl-1-(oxetan-2-ylmethyl)-6-[(2-oxoimidazolidin-1-yl)methyl]thieno[2,3-d]pyrimidine-2,4(1H,3H)-dione (racemate)\n813 mg (1.84 mmol) of the

spacy.displacy.render(doc, style="ent", jupyter=True)
```

# RESULTS

```
============================= Training pipeline =============================
ℹ Pipeline: ['tok2vec', 'ner']
ℹ Initial learn rate: 0.001
E    #       LOSS TOK2VEC  LOSS NER  ENTS_F  ENTS_P  ENTS_R  SCORE
---  ------  ------------  --------  ------  ------  ------  ------
  0       0          0.00     85.71    0.00    0.00    0.00    0.00
  0     200       4002.58  10521.94   48.75   55.53   43.45    0.49
  0     400        562.18   5976.94   70.64   71.25   70.04    0.71
  0     600        565.79   4512.95   71.82   71.69   71.96    0.72
  0     800        512.20   3690.81   80.89   80.53   81.25    0.81
  1    1000        543.58   3333.80   82.84   83.55   82.14    0.83
  1    1200        540.30   3088.63   81.30   80.24   82.39    0.81
  1    1400        687.25   3068.26   82.62   83.97   81.32    0.83
  1    1600        783.00   2614.99   83.31   83.12   83.50    0.83
  2    1800        753.02   2497.87   86.62   87.55   85.70    0.87
  2    2000        931.51   2821.82   87.46   87.24   87.69    0.87
  3    2200       1083.40   3761.94   87.84   88.75   86.94    0.88
  3    2400       1660.86   3731.91   88.59   88.37   88.81    0.89
  4    2600       2996.37   4945.30   88.25   89.49   87.04    0.88
✓ Saved pipeline to output directory
model-last
```

Example 194 EXAMPLE_LABEL

3-Isobutyl-5-methyl-1-(oxetan-2-ylmethyl)-6-[(2-oxoimidazolidin-1-yl)methyl]thieno[2,3-d]pyrimidine-2,4(1H,3H)-dione (racemate) REACTION_PRODUCT

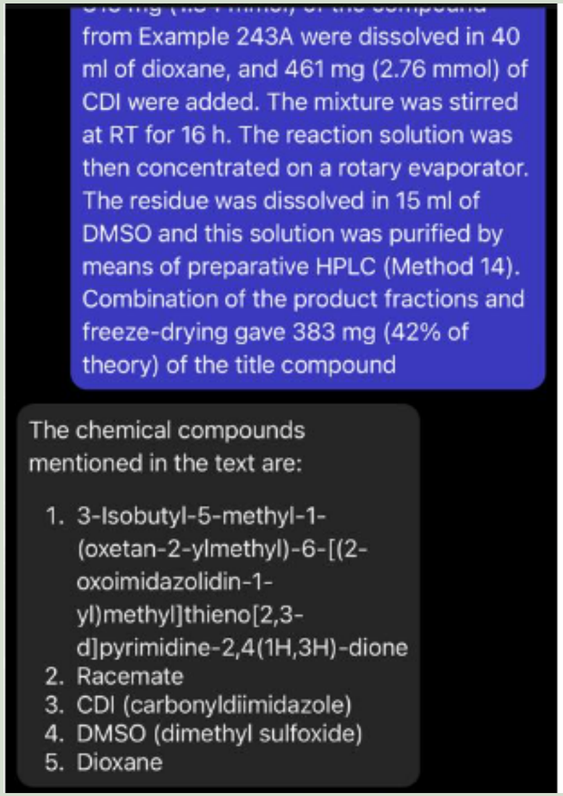
813 mg (1.84 mmol) of the compound from Example 243A STARTING_MATERIAL were dissolved in 40 ml of dioxane SOLVENT , and 461 mg (2.76 mmol) of CDI STARTING_MATERIAL were added. The mixture was stirred at RT TEMPERATURE for 16 TIME h. The reaction solution was then concentrated on a rotary evaporator. The residue was dissolved in 15 ml of DMSO SOLVENT and this solution was purified by means of preparative HPLC (Method 14). Combination of the product fractions and freeze-drying gave 383 mg YIELD_OTHER ( 42% YIELD_PERCENT of theory) of the title compound REACTION_PRODUCT

# Conclusion

The problem statement for named entity recognition (NER) of chemical compounds from a chemical dataset is to automatically identify and extract the names of chemical compounds mentioned in the text. The chemical dataset may contain various types of text, NER models need to be trained on large and diverse chemical datasets and should incorporate domain-specific knowledge and rules.

# FUTURE WORK

using Chat GPT  for named entity recognition of chemical compounds from chemical dataset




As the volume and complexity of chemical datasets continue to grow, the need for efficient and accurate NER models will only increase. Future research in this area will likely focus on developing more robust and generalizable models, incorporating more domain-specific knowledge and rules, and exploring new techniques such as deep learning and neural networks. Overall, NER of chemical compounds from chemical datasets holds great promise for advancing research and innovation in the chemical domain.

# Thank you