

# Hands-on: Clinical NLP with Python

## Clinical NLP Pipeline Example

```
# Install: pip install scispacy import spacy import scispacy from scispacy.linker import EntityLinker # Load clinical model nlp = spacy.load("en_core_sci_md") nlp.add_pipe("scispacy_linker", config={"resolve_abbreviations": True}) # Process clinical text text = "Patient prescribed metformin 500mg for T2DM. HbA1c was 7.2%" doc = nlp(text) # Extract entities for ent in doc.ents: print(f" {ent.text} > {ent.label_}") if ent._.kb_ents: cui = ent._.kb_ents[0][0] # UMLS CUI print(f" UMLS: {cui}")
```

```
# BioBERT for NER using Transformers from transformers import AutoTokenizer, AutoModelForTokenClassification import torch # Load BioBERT model tokenizer = AutoTokenizer.from_pretrained("dmis-lab/biobert-v1.1") model = AutoModelForTokenClassification.from_pretrained( "dmis-lab/biobert-v1.1" ) # Tokenize and predict inputs = tokenizer(text, return_tensors="pt") outputs = model(**inputs) predictions = torch.argmax(outputs.logits, dim=2)
```

## NLP Pipeline Flow



- pip install scispacy
- Biomedical NER models
- UMLS entity linking



- Medical Concept Annotation
- Unsupervised learning
- Active learning interface

- Abbreviation detection
- Negation with NegEx

- Context detection
- SNOMED/UMLS linking



### BioBERT/ClinicalBERT

- Pretrained on PubMed/MIMIC
- Fine-tuning for NER
- Relation extraction
- Question answering
- Hugging Face integration



### Evaluation Metrics

- Precision: correct/predicted
- Recall: correct/actual
- F1 score: harmonic mean
- Entity vs token level
- Cross-validation