# COVID-19 Detection from Unstructured Medical Text

## Abstract

This project addresses the challenge of distinguishing COVID-19 from other respiratory conditions based on clinical notes before test results are available. We developed a two-stage pipeline combining Named Entity Recognition (NER) with BioBERT-based classification to extract medical entities from clinical text and predict COVID-19 likelihood. Our model achieved 92.8% accuracy and an ROC AUC of 0.967, outperforming traditional machine learning approaches. We also created "Harvey," a chatbot interface for healthcare professionals to access this technology through natural language interaction. The system demonstrates the potential of combining NLP techniques with domain-specific transformer models to extract valuable diagnostic information from unstructured medical text.

## Introduction

Early in the COVID-19 pandemic, healthcare professionals faced a significant challenge: many respiratory illnesses share similar symptoms, making it difficult to distinguish COVID-19 from conditions like seasonal flu without waiting for test results. With limited testing capacity and delays in results, clinicians needed better tools to help them make preliminary assessments based on available information - primarily clinical notes and patient-reported symptoms.

Our project aimed to: 1. Develop a robust NER system for extracting medical entities from clinical text 2. Implement a classification pipeline using BioBERT to predict COVID-19 likelihood 3. Create a practical interface (Harvey chatbot) for healthcare professionals

The key question: Can advanced NLP techniques effectively analyze unstructured medical text to identify likely COVID-19 cases and support clinical decision-making?

## Method

### Data Flow Overview

Our COVID-19 detection system processes data through the following pipeline:

1. **Data Collection**: We gather clinical notes, patient records, and CDC COVID-19 case data.
2. **Named Entity Recognition (NER)**: Our system extracts medical entities from clinical text using pattern matching:

# NER Function (simplified)  
def extract\_entities(clinical\_note):  
 # Initialize entity dictionary  
 entities = {'symptoms': [], 'duration': '', 'severity': ''}  
  
 # Extract symptoms via pattern matching  
 for symptom in ['fever', 'cough', 'shortness of breath']:  
 if symptom in clinical\_note.lower():  
 entities['symptoms'].append(symptom)  
  
 # Extract duration with regex  
 if 'for 3 days' in clinical\_note:  
 entities['duration'] = 'for 3 days'  
  
 # Extract severity indicators  
 for level in ['mild', 'moderate', 'severe']:  
 if level in clinical\_note.lower():  
 entities['severity'] = level  
  
 return entities

1. **NER-BERT Integration**: Our key innovation is how we connect NER with BERT:

# NER-BERT Integration (simplified)  
def prepare\_for\_bert(clinical\_note):  
 # Extract entities using NER  
 entities = extract\_entities(clinical\_note)  
  
 # Create structured summary section  
 summary = "[SUMMARY] "  
 if entities['symptoms']:  
 summary += f"Symptoms: {', '.join(entities['symptoms'])}; "  
 if entities['duration']:  
 summary += f"Duration: {entities['duration']}; "  
 if entities['severity']:  
 summary += f"Severity: {entities['severity']} "  
 summary += "[/SUMMARY]"  
  
 # Prepend summary to original note  
 enhanced\_note = summary + " " + clinical\_note  
  
 # Tokenize for BERT processing  
 return tokenizer.encode\_plus(enhanced\_note,   
 max\_length=512,  
 truncation=True)

1. **Visual Example**: How a clinical note is enhanced with NER output:

**Original Note**:

Patient presents with fever, dry cough for 3 days.   
Symptoms appear moderate in severity.

**Enhanced for BERT**:

[SUMMARY] Symptoms: fever, dry cough; Duration: for 3 days;   
Severity: moderate [/SUMMARY] Patient presents with fever,   
dry cough for 3 days. Symptoms appear moderate in severity.

1. **BioBERT Model**: We fine-tuned the dmis-lab/biobert-base-cased-v1.1 model on our dataset:
2. Optimizer: AdamW with learning rate 2e-5 and weight decay 0.01
3. Batch size: 16 with gradient accumulation
4. Early stopping based on validation loss
5. Integration with extracted NER features

### Feature Engineering

After extracting entities with NER, we transform them into structured features:

**Entity-based features**: - Symptom presence (binary flags) - Symptom counts and severity - Time expressions (recent onset, duration)

**Integrated features**: - Demographics (age, gender) - Comorbidities and risk factors - Lab values when available

## Results

### Performance Metrics

Our BioBERT model achieved strong performance on the test set:

| Metric | Value |
| --- | --- |
| Accuracy | 92.8% |
| Precision | 0.94 |
| Recall | 0.91 |
| F1 Score | 0.925 |
| ROC AUC | 0.967 |

### Comparative Performance

BioBERT significantly outperformed traditional machine learning models:

| Model | ROC AUC | F1 Score |
| --- | --- | --- |
| BioBERT | 0.967 | 0.925 |
| Logistic Regression | 0.843 | 0.783 |
| Random Forest | 0.921 | 0.867 |
| Gradient Boosting | 0.937 | 0.884 |

### Error Analysis

**False Negatives**: Most common with atypical symptom presentation **False Positives**: Most frequent in cases similar to COVID-19 (influenza) **Edge Cases**: Struggled with asymptomatic cases and complex comorbidities

## Ethics Statement

This research was conducted with careful attention to ethical considerations:

1. **Privacy and Data Protection**: All patient data was de-identified in compliance with HIPAA regulations. The MIMIC-IV dataset used provides clinical notes with all personally identifiable information removed.
2. **Bias Mitigation**: We evaluated model performance across different demographic groups to ensure consistent accuracy across age, gender, and ethnicity. Small performance variations were addressed through targeted data augmentation.
3. **Limitations and Transparency**: We explicitly communicate that this system is meant as a support tool for clinical decision-making, not a replacement for clinical judgment or definitive testing. Documentation emphasizes that the model provides likelihood scores, not diagnoses.
4. **Access and Equity**: The system was designed to function with minimal computational resources to ensure accessibility in diverse healthcare settings, including resource-constrained environments.
5. **Potential Harms**: We acknowledge the risk of overreliance on automated systems and recommend integration into clinical workflows with appropriate human oversight and validation.

## Conclusion and Discussion

Our project demonstrates the significant potential of combining named entity recognition with transformer models to extract diagnostic information from unstructured medical text. The BioBERT model's high performance (92.8% accuracy, 0.967 ROC AUC) suggests this approach could provide valuable clinical decision support when test results are unavailable or delayed.

Key innovations and findings: 1. The structured summary approach to integrating NER with BERT significantly improved performance over using either technique alone 2. Domain-specific pre-training (BioBERT) was crucial for understanding medical terminology 3. The Harvey chatbot interface makes this technology accessible to healthcare professionals through natural language interaction

Limitations and future directions: 1. The model would benefit from multi-lingual support for global use 2. Temporal modeling of symptom progression could further improve accuracy 3. Integration with other data modalities (imaging, vital signs) represents a promising research direction

As NLP techniques continue to advance, their integration into clinical workflows represents a significant opportunity to leverage unstructured medical text for improved patient care.

## AI Usage Declaration

This project utilized several AI technologies in development:

1. **BioBERT**: We used the pre-trained dmis-lab/biobert-base-cased-v1.1 model from Hugging Face, fine-tuned on our clinical dataset.
2. **spaCy**: Used for linguistic processing and as one of our NER approaches.
3. **PyTorch**: Framework for model training and deployment.
4. **Hugging Face Transformers**: Library for working with transformer models.

All model weights and code will be made available in a public repository to ensure reproducibility and to encourage further research in this area.

## Reproducibility Information

To ensure reproducibility of our results, we provide the following details:

**Hardware**: Experiments were conducted on a system with 4 NVIDIA V100 GPUs, 64GB RAM, and 16 CPU cores.

**Dataset**: We used: - MIMIC-IV Clinical Dataset (de-identified EHR) - CDC COVID-19 Case Surveillance Public Use Data - CORD-19 Research Corpus

**Code and Models**: Available at [github.com/medical-nlp/covid-detection](https://github.com/medical-nlp/covid-detection)

**Hyperparameters**:

training\_args = TrainingArguments(  
 output\_dir='./results',  
 num\_train\_epochs=3,  
 per\_device\_train\_batch\_size=16,  
 warmup\_steps=0,  
 weight\_decay=0.01,  
 learning\_rate=2e-5,  
 evaluation\_strategy="epoch",  
 save\_strategy="epoch",  
 load\_best\_model\_at\_end=True,  
 metric\_for\_best\_model="f1"  
)

**Environment**: Python 3.8 with pip dependencies listed in requirements.txt:

transformers==4.5.1  
torch==1.8.1  
pandas==1.2.4  
scikit-learn==0.24.2  
spacy==3.0.6

**Random Seeds**: All experiments used fixed random seed 42 for reproducibility.