

By Gayathiri Premkumar - Statistical Programmer II



With biomedical literature expanding exponentially, extracting meaningful events from unstructured text is more vital than ever. Our Python-based pipeline fine-tunes BioBERT to classify domain-specific biomedical events—moving beyond rule-based systems to harness deep contextual understanding. From raw text to event prediction, this framework streamlines tasks like pharmacovigilance and disease progression analysis, accelerating insights in life sciences.

✓ **Bio-BERT** → Domain-specific biomedical language representation model built on top of BERT. (Bidirectional Encoder Representations from Transformers)

✓ **Improve performance ,NLP Tasks** → Relation extraction, Text classification, Sentiment analysis, Machine translation, Question answering.

✓ **Input** → Unstructured textual data from PRO.

✓ **Output** → Drug-related event, Non-event.

✓ **Who will use this model?**

Pharmacovigilance Analysts: quickly identify potential safety signals.

Clinical Researchers: understand the real-world effects of drugs and identify previously unknown side effects.

Healthcare Professionals: *better clinical decision-making.*

```

graph LR
    A[Traditional methods rely on rule-based systems or keyword matching, but machine learning models like BioBERT enhance accuracy by leveraging contextual understanding.] --> B[Using BioBERT in Python typically involves:]
    B --> C[Preprocessing Text Data – Tokenizing & formatting clinical text for model input.]
    C --> D[Fine-tuning BioBERT – Training the model on labeled adverse event datasets.]
    D --> E[Prediction & Evaluation – Classifying text as containing adverse events or not, followed by performance assessment.]
  
```

Traditional methods rely on rule-based systems or keyword matching, but machine learning models like BioBERT enhance accuracy by leveraging contextual understanding.

→

Using BioBERT in Python typically involves:

→

Preprocessing Text Data – Tokenizing & formatting clinical text for model input.

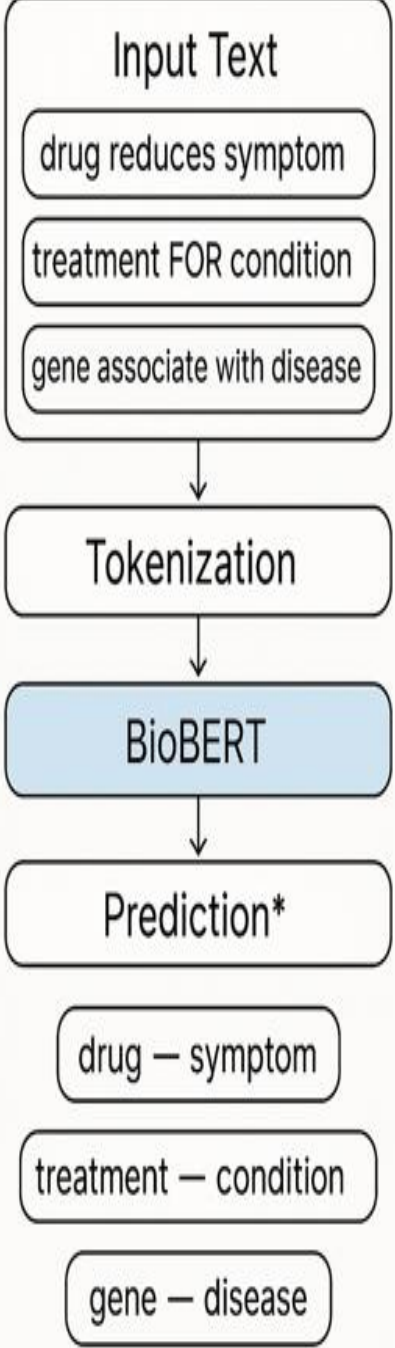
→

Fine-tuning BioBERT – Training the model on labeled adverse event datasets.

→

Prediction & Evaluation – Classifying text as containing adverse events or not, followed by performance assessment.

These Diagram illustrating how biomedical text is processed to identify relationships between key medical terms. It follows this sequence:



- **Input text** — examples like “drug reduces symptom.”
- **Tokenization** — the text is broken down into tokens.
- **Bio BERT** — a pre-trained biomedical language model processes the tokens.
- **Prediction** — it outputs relationships such as drug–symptom, treatment–condition, or gene–disease.

```
import torch
import pandas as pd
from transformers import BertTokenizer, BertForSequenceClassification, Trainer, TrainingArguments
from sklearn.model_selection import train_test_split
from torch.utils.data import Dataset
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import accuracy_score, precision_recall_fscore_support
```

```
# Split data
train_texts, val_texts, train_labels, val_labels = train_test_split(
    data['description'], data['label'], test_size=0.2, random_state=42)
train_dataset = TextDataset(train_texts.tolist(), train_labels.tolist())
val_dataset = TextDataset(val_texts.tolist(), val_labels.tolist())

# Load Pretrained BioBERT Model
num_labels = len(label_encoder.classes_)
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
model = BertForSequenceClassification.from_pretrained(
    (model_name, num_labels=num_labels)).to(device)

# Training Arguments
training_args = TrainingArguments(
    output_dir='./results',
    # evaluation_strategy='epoch', # Removed this argument
    per_device_train_batch_size=8,
    per_device_eval_batch_size=8,
    num_train_epochs=3,
    save_strategy='epoch',
    logging_dir='./logs',
    report_to="none"
)

# Compute metrics
def compute_metrics(eval_pred):
    logits, labels = eval_pred
    preds = torch.argmax(torch.tensor(logits), dim=-1)
    acc = accuracy_score(labels, preds)
    precision, recall, f1, _ = precision_recall_fscore_support(
        labels, preds, average='weighted')
    return {"accuracy": acc, "precision": precision, "recall": recall, "f1": f1}

# Trainer
trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=train_dataset,
    eval_dataset=val_dataset,
    compute_metrics=compute_metrics
)

# Train model
trainer.train()

# Evaluate model
metrics = trainer.evaluate()
print("Evaluation Metrics:", metrics)

# Prediction Function
def predict_classification(text):
    inputs = tokenizer(str(text), return_tensors="pt", truncation=True, padding=True, max_length=128)
    inputs = {key: val.to(device) for key, val in inputs.items()} # Move inputs to same device as model
    outputs = model(**inputs)
    probs = torch.nn.functional.softmax(outputs.logits, dim=-1)
    prediction = torch.argmax(probs, dim=-1).item()
    return label_encoder.inverse_transform([prediction])[0]
```

```
1] df=pd.read_csv("event_classification_sample.csv", encoding='latin-1')

df['predicted_label'] = df['description'].apply(predict_classification)
df
```

index	description	Drug	classification	predicted_label
0	while taking Drug- A the patient was diagnosed with acute myocardial infarction.	Drug A	AE	Non Event
1	No significant abnormalities were found in the blood test results.	Drug B	Non Event	AE
2	The clinical trial showed promising results for the new drug.	Drug A	Non Event	Non Event
3	The patient experienced severe chest pain and shortness of breath.	Drug B	AE	AE
4	Trying to sleep on with Drug A is so weird. There are so many jitters	Drug A	Non Event	AE
5	I take my Drug B like 2 hours before I get up for the day and like those last couple hours of sleep are always nightmares	Drug B	AE	AE
6	The patient was admitted to the hospital for observation.	Drug A	AE	Non Event

[illegible]

- BioBERT acts as a powerful feature extractor for biomedical text, and a subsequent classification layer uses these features to categorize events, leveraging BioBERT's deep understanding of the biomedical domain.
- Event classification using BioBERT involves training a biomedical language model to identify and categorize relationships or events—like adverse drug reactions—from unstructured clinical text.
- *This approach is particularly useful for pharmacovigilance, AE analysis, automating structured data extraction, Disease Progression and Treatment Monitoring , Clinical Trial Recruitment & Automated Coding from clinical narratives.*

Contact the author at:
 EMAIL: gayathiri.premkumar@sensanbio.com
 ORGANIZATION: SENSAN BIOSCIENCES, India.
 WEBSITE: WWW.SENSANBIO.COM