SetFit MODEL - 3

Model: MiniLM-L3-v2

N = 16

In [5]:
```
from setfit import SetFitModel, SetFitTrainer
from sentence_transformers.losses import CosineSimilarityLoss
from datasets import Dataset
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, f1_score
import pandas as pd
import os

# === CONFIG ===
DATA_PATH = "multilang_sarcasm_dataset.csv"
MODEL_PATH = "model/setfit_multilang_sarcasm_en"
N_SHOT = 16
MAX_TEST_SAMPLES = 1000

# === LOAD & PREPROCESS ===
df = pd.read_csv(DATA_PATH)

# Filter to English headlines
df = df[df["lang"] == "en"]

# Rename columns to match SETFIT input format
df = df[["article_title", "is_sarcastic"]].rename(columns={"article_ti

# Drop any potential NaNs
df = df.dropna(subset=["text", "label"])

# === TRAIN/TEST SPLIT ===
train_df, test_df = train_test_split(df, test_size=0.2, stratify=df["l

# Few-shot sampling
def sample_few_shot(df, n=64):
    return df.groupby("label").apply(lambda x: x.sample(n=min(n, len(x

fewshot_train_df = sample_few_shot(train_df, N_SHOT)
test_subset_df = test_df.sample(n=min(len(test_df), MAX_TEST_SAMPLES),

# Convert to HuggingFace datasets
train_dataset = Dataset.from_pandas(fewshot_train_df)
test_dataset = Dataset.from_pandas(test_subset_df)

# === LOAD BASE MODEL ===
model = SetFitModel.from_pretrained("sentence-transformers/paraphrase-

# === TRAIN SETUP ===
```

```python
trainer = SetFitTrainer(
    model=model,
    train_dataset=train_dataset,
    eval_dataset=test_dataset,
    loss_class=CosineSimilarityLoss,
    batch_size=16,
    num_iterations=50,
    num_epochs=1,
    column_mapping={"text": "text", "label": "label"},
)

trainer.train()

# Save model
model.save_pretrained(MODEL_PATH)

# Evaluate
y_true = test_dataset["label"]
y_pred = model.predict(test_dataset["text"])
acc = accuracy_score(y_true, y_pred)
f1 = f1_score(y_true, y_pred)

print(f"Accuracy: {acc:.4f} | F1 Score: {f1:.4f}")
```

```
/var/folders/lv/xd91rcv91cq23cjjl0_c93nh0000gn/T/ipykernel_82227/1468048697.py:32: DeprecationWarning: DataFrameGroupBy.apply operated on the grouping columns. This behavior is deprecated, and in a future version of pandas the grouping columns will be excluded from the operation. Either pass `include_groups=False` to exclude the groupings or explicitly select the grouping columns after groupby to silence this warning.
  return df.groupby("label").apply(lambda x: x.sample(n=min(n, len(x)), random_state=42)).reset_index(drop=True)
`SentenceTransformer._target_device` has been deprecated, please use `SentenceTransformer.device` instead.
model_head.pkl not found on HuggingFace Hub, initialising classification head with random weights. You should TRAIN this model on a downstream task to use it for predictions and inference.
Applying column mapping to training dataset
Generating Training Pairs: 100%|██████████| 50/50 [00:00<00:00, 2996.53it/s]
***** Running training *****
  Num examples = 3200
  Num epochs = 1
  Total optimization steps = 200
  Total train batch size = 16
```

[200/200 00:08, Epoch 1/1]

Step	Training Loss

```
Accuracy: 0.6480 | F1 Score: 0.6576
```

SetFit MODEL - 3

Model: MiniLM-L3-v2

N = 32

In [6]:

```
from setfit import SetFitModel, SetFitTrainer
from sentence_transformers.losses import CosineSimilarityLoss
from datasets import Dataset
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, f1_score
import pandas as pd
import os

# === CONFIG ===
DATA_PATH = "multilang_sarcasm_dataset.csv"
MODEL_PATH = "model/setfit_multilang_sarcasm_en_32"
N_SHOT = 32
MAX_TEST_SAMPLES = 1000

# === LOAD & PREPROCESS ===
df = pd.read_csv(DATA_PATH)

# Filter to English headlines
df = df[df["lang"] == "en"]

# Rename columns to match SETFIT input format
df = df[["article_title", "is_sarcastic"]].rename(columns={"article_ti

# Drop any potential NaNs
df = df.dropna(subset=["text", "label"])

# === TRAIN/TEST SPLIT ===
train_df, test_df = train_test_split(df, test_size=0.2, stratify=df["l

# Few-shot sampling
def sample_few_shot(df, n=64):
    return df.groupby("label").apply(lambda x: x.sample(n=min(n, len(x

fewshot_train_df = sample_few_shot(train_df, N_SHOT)
test_subset_df = test_df.sample(n=min(len(test_df), MAX_TEST_SAMPLES),

# Convert to HuggingFace datasets
train_dataset = Dataset.from_pandas(fewshot_train_df)
test_dataset = Dataset.from_pandas(test_subset_df)

# === LOAD BASE MODEL ===
model = SetFitModel.from_pretrained("sentence-transformers/paraphrase-

# === TRAIN SETUP ===
trainer = SetFitTrainer(
    model=model,
    train_dataset=train_dataset,
    eval_dataset=test_dataset,
```

```
    loss_class=CosineSimilarityLoss,
    batch_size=16,
    num_iterations=50,
    num_epochs=1,
    column_mapping={"text": "text", "label": "label"},
)

trainer.train()

# Save model
model.save_pretrained(MODEL_PATH)

# Evaluate
y_true = test_dataset["label"]
y_pred = model.predict(test_dataset["text"])
acc = accuracy_score(y_true, y_pred)
f1 = f1_score(y_true, y_pred)

print(f"Accuracy: {acc:.4f} | F1 Score: {f1:.4f}")
```

```
/var/folders/lv/xd91rcv91cq23cjjl0_c93nh0000gn/T/ipykernel_82227/292298
277.py:32: DeprecationWarning: DataFrameGroupBy.apply operated on the g
rouping columns. This behavior is deprecated, and in a future version o
f pandas the grouping columns will be excluded from the operation. Eith
er pass `include_groups=False` to exclude the groupings or explicitly s
elect the grouping columns after groupby to silence this warning.
  return df.groupby("label").apply(lambda x: x.sample(n=min(n, len(x)),
random_state=42)).reset_index(drop=True)
`SentenceTransformer._target_device` has been deprecated, please use `S
entenceTransformer.device` instead.
model_head.pkl not found on HuggingFace Hub, initialising classificatio
n head with random weights. You should TRAIN this model on a downstream
task to use it for predictions and inference.
Applying column mapping to training dataset
Generating Training Pairs: 100%|██████████| 50/50 [00:00<00:00, 1391.36
it/s]
***** Running training *****
  Num examples = 6400
  Num epochs = 1
  Total optimization steps = 400
  Total train batch size = 16
```

[400/400 00:17, Epoch 1/1]

Step	Training Loss

```
Accuracy: 0.6900 | F1 Score: 0.6645
```

SetFit MODEL - 3

Model: MiniLM-L3-v2

N = 64

In [1]:

```
from setfit import SetFitModel, SetFitTrainer
from sentence_transformers.losses import CosineSimilarityLoss
from datasets import Dataset
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, f1_score
import pandas as pd
import os

# === CONFIG ===
DATA_PATH = "multilang_sarcasm_dataset.csv"
MODEL_PATH = "model/setfit_multilang_sarcasm_en_64"
N_SHOT = 64
MAX_TEST_SAMPLES = 1000

# === LOAD & PREPROCESS ===
df = pd.read_csv(DATA_PATH)

# Filter to English headlines
df = df[df["lang"] == "en"]

# Rename columns to match SETFIT input format
df = df[["article_title", "is_sarcastic"]].rename(columns={"article_ti

# Drop any potential NaNs
df = df.dropna(subset=["text", "label"])

# === TRAIN/TEST SPLIT ===
train_df, test_df = train_test_split(df, test_size=0.2, stratify=df["l

# Few-shot sampling
def sample_few_shot(df, n=64):
    return df.groupby("label").apply(lambda x: x.sample(n=min(n, len(x

fewshot_train_df = sample_few_shot(train_df, N_SHOT)
test_subset_df = test_df.sample(n=min(len(test_df), MAX_TEST_SAMPLES),

# Convert to HuggingFace datasets
train_dataset = Dataset.from_pandas(fewshot_train_df)
test_dataset = Dataset.from_pandas(test_subset_df)

# === LOAD BASE MODEL ===
model = SetFitModel.from_pretrained("sentence-transformers/paraphrase-

# === TRAIN SETUP ===
trainer = SetFitTrainer(
    model=model,
    train_dataset=train_dataset,
    eval_dataset=test_dataset,
    loss_class=CosineSimilarityLoss,
    batch_size=16,
    num_iterations=50,
```

```
    num_epochs=1,
    column_mapping={"text": "text", "label": "label"},
)

trainer.train()

# Save model
model.save_pretrained(MODEL_PATH)

# Evaluate
y_true = test_dataset["label"]
y_pred = model.predict(test_dataset["text"])
acc = accuracy_score(y_true, y_pred)
f1 = f1_score(y_true, y_pred)

print(f"Accuracy: {acc:.4f} | F1 Score: {f1:.4f}")
```

```
/Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/site-
packages/tqdm/auto.py:21: TqdmWarning: IProgress not found. Please upda
te jupyter and ipywidgets. See https://ipywidgets.readthedocs.io/en/sta
ble/user_install.html
  from .autonotebook import tqdm as notebook_tqdm
/var/folders/lv/xd91rcv91cq23cjjl0_c93nh0000gn/T/ipykernel_82829/450116
200.py:32: DeprecationWarning: DataFrameGroupBy.apply operated on the g
rouping columns. This behavior is deprecated, and in a future version o
f pandas the grouping columns will be excluded from the operation. Eith
er pass `include_groups=False` to exclude the groupings or explicitly s
elect the grouping columns after groupby to silence this warning.
  return df.groupby("label").apply(lambda x: x.sample(n=min(n, len(x)),
random_state=42)).reset_index(drop=True)
`SentenceTransformer._target_device` has been deprecated, please use `S
entenceTransformer.device` instead.
model_head.pkl not found on HuggingFace Hub, initialising classificatio
n head with random weights. You should TRAIN this model on a downstream
task to use it for predictions and inference.
Applying column mapping to training dataset
Generating Training Pairs: 100%|██████████| 50/50 [00:00<00:00, 1095.85
it/s]
***** Running training *****
  Num examples = 12800
  Num epochs = 1
  Total optimization steps = 800
  Total train batch size = 16
```

[800/800 00:35, Epoch 1/1]

Step	Training Loss
500	0.166300

```
Accuracy: 0.7150 | F1 Score: 0.7016
```