```
import logging
import set_ops as setops
import setops_sup
from transformers import AutoConfig, AutoModel, AutoTokenizer, PreTrainedModel
import torch
import random
import os
import argparse
import string
import shutil
from collections import defaultdict
import pickle
import time
import pandas as pd

def id_generator(size=6):
    return ''.join(random.choice(string.ascii_uppercase + string.digits) for _ in
range(size))

def sup_train(df_all, model_name, data_tempdir, model_tempdir, n_sample,
train_epoch, bs, local_files_only):
    args = setops_sup.Training_args(model_name=model_name, bs=bs,
train_epoch=train_epoch, output_dir=model_tempdir)
    trainer = setops_sup.SemSOCTrainer(args, local_files_only=local_files_only)
    sample_ls = dict()
    sample_sets = dict()
    cats = df_all.label.value_counts().iloc[:3].index.to_list()
    for cat in cats:
        dfi = df_all[df_all['label']==cat]
        dfi.drop_duplicates()
        sample = dfi.sample(frac=1).iloc[:n_sample]
        sample.to_csv(data_tempdir+f'/{cat}_sample.csv', index=False)
        sample_ls[cat] = set(sample.utterance)
        sample_set = setops_sup.Set(data_path=data_tempdir+f'/{cat}_sample.csv')
        sample_sets[cat] = sample_set
    trainer.SemSOCtrain(list(sample_sets.values()))
    tokenizer = AutoTokenizer.from_pretrained(args.model_name)
    model = AutoModel.from_pretrained(args.output_dir)
    return tokenizer, model

model_name = "princeton-nlp/sup-simcse-bert-base-uncased"
data_dirs = ["data/ag_news/description_3000.csv", "data/FPB/all.csv",
"data/banking77/banking_int3.csv", "data/fb/fb_int3.csv"]
data_tags = ["AGD", "FPB", "Banking77", "FMTOD"]
# data_dirs = ["data/ag_news/title_3000.csv"]
```

```
# data_tags = ["AGT"]

for i in range(len(data_dirs)):
    temp_code = id_generator(4)
    data_tempdir = 'data/temp_' + temp_code
    model_tempdir = 'models/sts_' + data_tags[i]
    if not os.path.exists(data_tempdir):
        os.makedirs(data_tempdir)
    if not os.path.exists(model_tempdir):
        os.makedirs(model_tempdir)

    df_all = pd.read_csv(data_dirs[i])
    tokenizer, model = sup_train(df_all, model_name, data_tempdir, model_tempdir,
n_sample=20, train_epoch=60, bs=48, local_files_only=False)

    shutil.rmtree(data_tempdir)

    print(f"Done with {data_tags[i]}!")

print("Done!")
```