```
"""Sentiment Analysis API for individual stories"""
from app import app
from sent_analysis import subjectize, polarize
from models import Story
from flask import jsonify, session
from helpers import *



@app.route('/story/<id>/polarity', methods=['POST'])
def show_pol_calls(id):
    try:
        # check to see if id represents a sqlalchemy object that needs converted to dict to be fed to SA
functions
        # write logic to save score to db if story saved
        db_story = Story.query.get(id)
        story = db_story_to_dict(db_story)
    except:
        results = session['results']
        story = [story for story in results if story['id'] == id][0]
    score = polarize(story)
    if not score:
        story['pol'] = "No Data"
    else:
        score = score['article_res']['result']
        story['pol'] = str(score)
    return jsonify({'response': story['pol']})


@app.route('/story/<id>/subjectivity', methods=['POST'])
def show_sub_calls(id):
    try:
        # check to see if id represents a sqlalchemy object that needs converted to dict to be fed to SA
functions
        db_story = Story.query.get(id)
        story = db_story_to_dict(db_story)
    except:
        results = session['results']
        story = [story for story in results if story['id'] == id][0]
    score = subjectize(story)
    if not score:
        story['sub'] = "No Data"
    else:
        score = score['measure']
        story['sub']= str(score)
    return jsonify({'response': story['sub']})
```