In [1]:

```
import flask
import pandas
from flask import Flask, jsonify, request,render_template,Response
from werkzeug.wrappers import Request, Response
import tensorflow as tf
import pickle
```

In [2]:

```
app = Flask(__name__)
```

In [ ]:

```
%%time
path = "pickles"
model = tf.saved_model.load(path)
```

In [ ]:

```
%%time
# loading the dumped file
filename = "model_and_tokenizers/tokenizer_source.pkl"
with open(filename, 'rb') as file:
  tokenizer_source = pickle.load(file)
filename = "model_and_tokenizers/tokenizer_target.pkl"
with open(filename, 'rb') as file:
  tokenizer_target = pickle.load(file)
```

In [ ]:

```
max_len=39
max_len_dec=44
```

In [ ]:

```
def predict_(input_sentence):

  input_sequence=tokenizer_source.texts_to_sequences([input_sentence])
  inputs=pad_sequences(input_sequence, maxlen=max_len, padding='post')
  inputs=tf.convert_to_tensor(inputs)
  result=''
  units=128
  hidden=[tf.zeros((1,units))]
  encoder_output,hidden_state,cell_state=model.encoder(inputs)
  dec_hidden=hidden_state
  dec_input=tf.expand_dims([tokenizer_target.word_index['<start>']],0)
  for t in range(40):
      predictions,dec_hidden,cell_state,attention_weights,context_vector=model.decoder.onestepdecoder((dec_input,
encoder_output,dec_hidden,cell_state))

      predicted_id=tf.argmax(predictions[0]).numpy()
      result+=tokenizer_target.index_word[predicted_id]+' '
      if tokenizer_target.word_index['<end>']==predicted_id:
          return result
      dec_input= tf.expand_dims([predicted_id],0)
  return result
```

reference

 # https://youtu.be/UbCWoMf80PY

In [ ]:

```
@app.route('/')
def index():
    return render_template('index.html')
```

In [ ]:

```
@app.route('/predict', methods=['POST'])
def predict():
    print("request",request.form.values())
    to_predict_list = [x for x in request.form.values()]
    sentence = to_predict_list[0]
    output = predict_(sentence)
    return render_template('index.html', prediction_text='predicted output is  {}'.format(output))
```

In [ ]:

```
# https://stackoverflow.com/questions/31252791/flask-importerror-no-module-named-flask
# this cell should always be in running
if __name__ == '__main__':
    from werkzeug.serving import run_simple
    run_simple('localhost', 9000, app)
```