

## Testing Model on PC and Deploying Using Flask:

### 1. Create virtual environment and activate it

It's better to create a virtual environment and install the required libraries in it so that the new libraries don't conflict with the existing ones. Create your project's folder, open cmd and type the following commands for environment creation and activation

```
C:\Users\User\Desktop\New folder (2)>python -m venv C:\Users\User\Desktop\New folder (2)
C:\Users\User\Desktop\New folder (2)>folder\Scripts\activate.bat
```

Figure 1: Environment creation and activation

### 2. Set python packages

Once the environment is activated, we have to install the required libraries as a list of python packages and their versions in a separate file called requirements.txt for example.

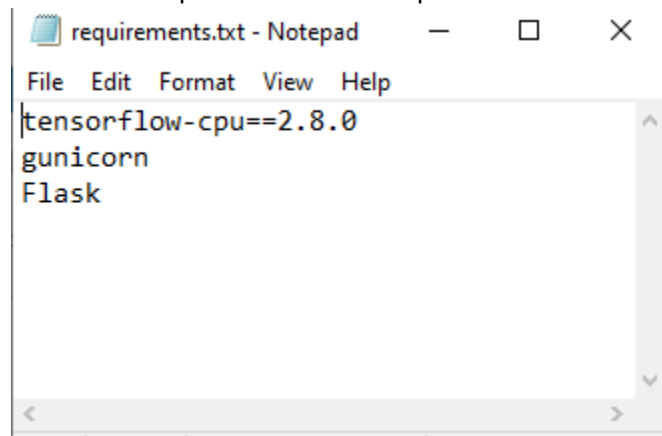











Figure 2: Prerequisite libraries list

Tensorflow library contains many other libraries like numpy, scipy etc.. so they will get installed along the way.

Then we can use the below command to install them.

```
pip install -r requirements.txt
```

### 3. Project Structure

<input type="checkbox"/> Name	Date modified	Type	Size
 <code>__pycache__</code>	4/30/2022 3:52 PM	File folder	
 <code>folder</code>	4/29/2022 8:07 PM	File folder	
 <code>model</code>	4/30/2022 5:28 PM	File folder	
 <code>templates</code>	4/30/2022 4:46 PM	File folder	
 <code>app.py</code>	5/1/2022 1:19 AM	PY File	2 KB
 <code>command.md</code>	5/1/2022 5:53 PM	MD File	1 KB
 <code>en_word_index.txt</code>	4/29/2022 2:41 PM	Text Document	3 KB
 <code>fr_word_index.txt</code>	4/29/2022 2:43 PM	Text Document	6 KB
 <code>requirements.txt</code>	5/3/2022 11:46 AM	Text Document	1 KB

- “folder” folder is my virtual environment
- “model” folder contains “En\_Fr\_Translation.h5” and “model.py” . h5 file is our model trained, and “model.py” is used to import model and execute the prediction.
- “templates” folder contains an “index.html” file used for flask.
- “app.py” file is the web application using flask
- “en\_word\_index.txt” and “fr\_word\_index.txt” are our English and French vocabularies used in the “model.py” script
- “Command.md” includes the commands needed

```

model.py 3
C:\Users\User\Desktop> New Folder (2) > model > model.py > main
1  import os
2  os.environ['TF_CPP_MIN_LOG_LEVEL'] = '3'
3  import tensorflow as tf
4  from tensorflow.keras.models import load_model
5  from tensorflow.keras.preprocessing.sequence import pad_sequences
6  import re
7  import numpy as np
8  import pickle
9  import logging
10
11 class En_Fre_Translation:
12
13     def __init__(self, model_path):
14         logging.info("En_Fre_Translation class initialized")
15         self.model = load_model(model_path)
16         logging.info("Model is loaded!")
17
18     def load_text(self, path):
19         a_file = open(path, "rb")
20         output = pickle.load(a_file)
21         return output
22
23     def translate(self, text, max_seq_length):
24         text = re.sub(r'^\w\s', '', text)
25         en_word_index = self.load_text('en_word_index.txt')
26         fr_word_index = self.load_text('fr_word_index.txt')
27         sentence = [en_word_index[word] for word in text.split()]
28         sentence = pad_sequences([sentence], maxlen=max_seq_length, padding='post')
29         result = self.model.predict(sentence[:1])[0]
30         index_to_words = {id: word for word, id in fr_word_index.items()}
31         index_to_words[0] = '<PAD>'
32         result = ' '.join([index_to_words[prediction] for prediction in np.argmax(result, 1)])
33         result = result.split()
34         output = []
35         for i in result:
36             if i != '<PAD>':
37                 output.append(i)
38         return (' '.join(output))
39
40 def main():
41     model = En_Fre_Translation('..model\En_Fr_Translation.h5')
42     predicted_text = model.translate("she is driving the truck", 21)
43     logging.info(f"This is the translation: \n {predicted_text}")
44
45 if __name__ == "__main__":
46     logging.basicConfig(level=logging.INFO)
47     main()

```

Figure 3: screenshot of model.py script used to load model and give a prediction for a specified example

To test our saved model, we created a model.py script that contains all needed statements to make a prediction for a specified example “she is driving the truck” (Figure 3).

Run the model.py using the below command (Figure 4)

```

C:\Users\User\Desktop\New Folder (2)>folder\Scripts\activate.bat

(folder) C:\Users\User\Desktop\New Folder (2)>python model\model.py
INFO:root:En_Fre_Translation class initialized
INFO:root:Model is loaded!
INFO:root:This is the translation:
elle conduit le nouveau camion noir

```

Figure 4: screenshot of activating environment and running model.py

#### 4. Creating Web application using flask

Here we create the routes for the endpoints of our application. We used html files to create an input form so the user can write English sentence and press the submit button to translate it to French using route ("/"). The sentence is translated using translate function that imports the model.py file to make the prediction.

```

app.py
C:\Users\User\Desktop\New Folder (2) > app.py > ...
1  from flask import Flask, request, render_template
2  import logging
3  import os
4
5  from model.model import En_Fre_Translation
6
7  # define model path
8  model_path = './model/En_Fr_Translation.h5'
9  #model_path = 'model\En_Fr_Translation.h5'
10
11 app = Flask(__name__)
12
13 # create instance
14 model = En_Fre_Translation(model_path)
15 logging.basicConfig(level=logging.INFO)
16
17 def translate(sentence):
18     logging.info("translation request received!")
19     prediction = model.translate(sentence,21)
20     logging.info("translation from model",prediction)
21
22     return prediction
23
24 @app.route("/")
25 def my_form():
26     return render_template('index.html')
27
28 @app.route("/", methods=['GET','POST'])
29 def input_text():
30     if request.method == 'POST':
31         text=request.form['en_sent']
32         logging.info("input sentence is =", text)
33         translated = translate(text)
34         logging.info("Send translation request!")
35
36         return translate(text)
37
38 def main():
39     """Run the Flask app."""
40     app.run(host="0.0.0.0", port=8000, debug=True)
41
42
43 if __name__ == "__main__":
44     main()
45

```

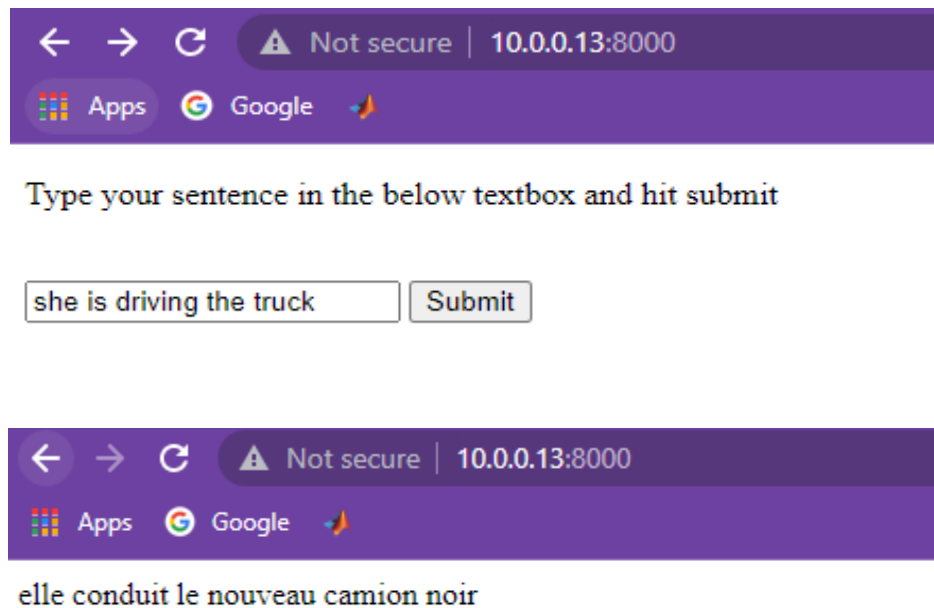
Figure 5: "app.py" script to create flask application

The flask app runs on the host at port 8000. To test the app, go to cmd and write:

```
(folder) C:\Users\User\Desktop\New Folder (2)>python app.py
* Serving Flask app 'app' (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
INFO:werkzeug: * Running on all addresses (0.0.0.0)
  WARNING: This is a development server. Do not use it in a production deployment.
* Running on http://127.0.0.1:8000
* Running on http://10.0.0.13:8000 (Press CTRL+C to quit)
INFO:werkzeug: * Restarting with stat
```

Figure 6: Run application on local host

We used this url “http://10.0.0.13:8000” to run application on the browser. Once the page is opened, you can type your sentence, press submit button to view the translation.



← → ↻ ⚠ Not secure | 10.0.0.13:8000

Apps Google

Type your sentence in the below textbox and hit submit

she is driving the truck Submit

elle conduit le nouveau camion noir

Figure 7: Sentence translation

## Deploy using Heroku Method 1

To deploy our model using Heroku it's simpler and faster. We just need to add a library to requirements file which is gunicorn, and create “Procfile” file that will initiate flask app.

1. We need Create account/Login on Heroku, create project, download Heroku Git (Github had an error).
2. We need to Create Procfile to initiate Flask app command

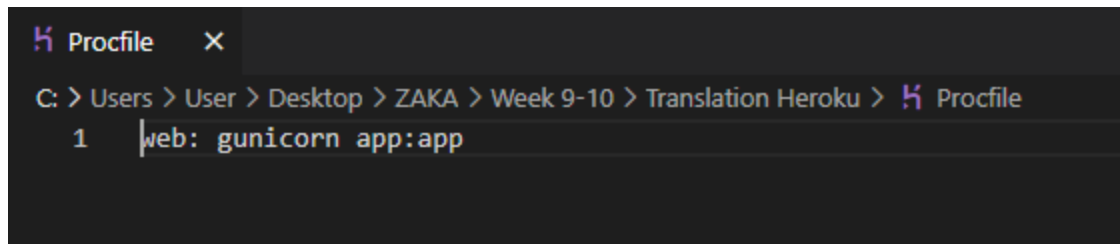


Figure 8:Procfile

3. We need to initialize a git folder
4. add to it all the required files (Procfile app.py, en\_word\_index.txt, fr\_word\_index.txt, model/ , requirements.txt, templates/)
5. Commit files
6. Push them

model	5/1/2022 3:36 AM	File folder	
templates	5/1/2022 3:36 AM	File folder	
app.py	5/1/2022 1:19 AM	PY File	2 KB
command.md	5/2/2022 8:56 PM	MD File	1 KB
en_word_index.txt	4/29/2022 2:41 PM	Text Document	3 KB
fr_word_index.txt	4/29/2022 2:43 PM	Text Document	6 KB
Procfile	5/1/2022 2:01 AM	File	1 KB
requirements.txt	5/2/2022 8:43 PM	Text Document	1 KB

Figure 9: Project structure

```
C:\Users\User\Desktop\ZAKA\Week 9-10\Translation Heroku>Heroku login
» Warning: heroku update available from 7.53.0 to 7.60.2.
heroku: Press any key to open up the browser to login on q to exit:
Opening browser to https://cli-auth.heroku.com/auth/cli/browser/2ae3eaaaf-14ae-4a29-8d20-5f64518fb729?requestor=5FMyNTY.g2gDbQAAAAwxNzguMTM1LjMuOTRuBgCYr0yA
gAFIAAFRgA.7bvGzFYZc1DyYkptuGUS_10Q_-41FkXB6lBRajkkUD0
Logging in... done
Logged in as kamar.shehimi@gmail.com

C:\Users\User\Desktop\ZAKA\Week 9-10\Translation Heroku>git config --global user.email kamar.shehimi@gmail.com

C:\Users\User\Desktop\ZAKA\Week 9-10\Translation Heroku>git init
Reinitialized existing Git repository in C:/Users/User/Desktop/ZAKA/Week 9-10/Translation Heroku/.git/

C:\Users\User\Desktop\ZAKA\Week 9-10\Translation Heroku>heroku git:remote -a eng-fre-translation
» Warning: heroku update available from 7.53.0 to 7.60.2.
set git remote heroku to https://git.heroku.com/eng-fre-translation.git

C:\Users\User\Desktop\ZAKA\Week 9-10\Translation Heroku>git add Procfile app.py en_word_index.txt fr_word_index.txt model/ requirements.txt templates/

C:\Users\User\Desktop\ZAKA\Week 9-10\Translation Heroku>git commit -m "Initial commit"
[master f473a60] Initial commit
1 file changed, 1 insertion(+), 4 deletions(-)

C:\Users\User\Desktop\ZAKA\Week 9-10\Translation Heroku>git push heroku master
Enumerating objects: 27, done.
Counting objects: 100% (27/27), done.
Delta compression using up to 4 threads
Compressing objects: 100% (21/21), done.
Writing objects: 44% (12/27), 7.96 MiB | 92.00 KiB/s
```

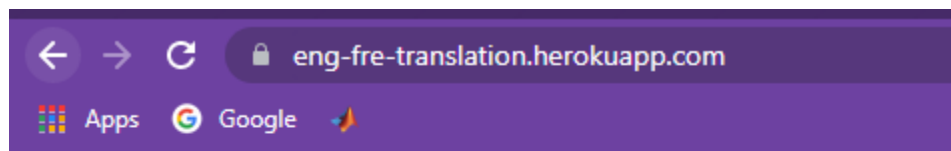
Figure 10: Commands to deploy using Heroku

```

d-2.8.0 tensorflow-tf-gcs filesystem-0.23.0 tensorflow-1.11.0 tf-estimator-nightly-2.8.0.dev2021122109 typing-extensions-4.1.1
remote: -----> Discovering process types
remote: Procfile declares types -> web
remote: -----> Compressing...
remote: Done: 321.5M
remote: -----> Launching...
remote: ! Warning: Your slug size (321 MB) exceeds our soft limit (300 MB) which may affect boot time.
remote: Released v3
remote: https://eng-fre-translation.herokuapp.com/ deployed to Heroku
remote: Verifying deploy... done.
To https://git.heroku.com/eng-fre-translation.git
* [new branch] master -> master
C:\Users\User\Desktop\ZAKA\Week 9-10\Translation Heroku>

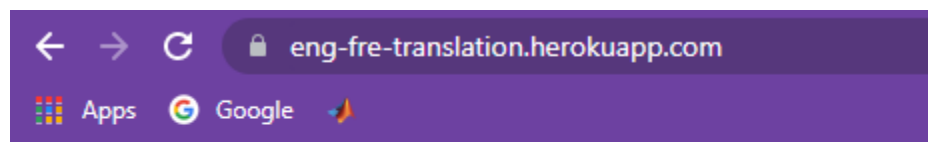
```

Figure 11: Follow this link to open app



Type your sentence in the below textbox and hit submit

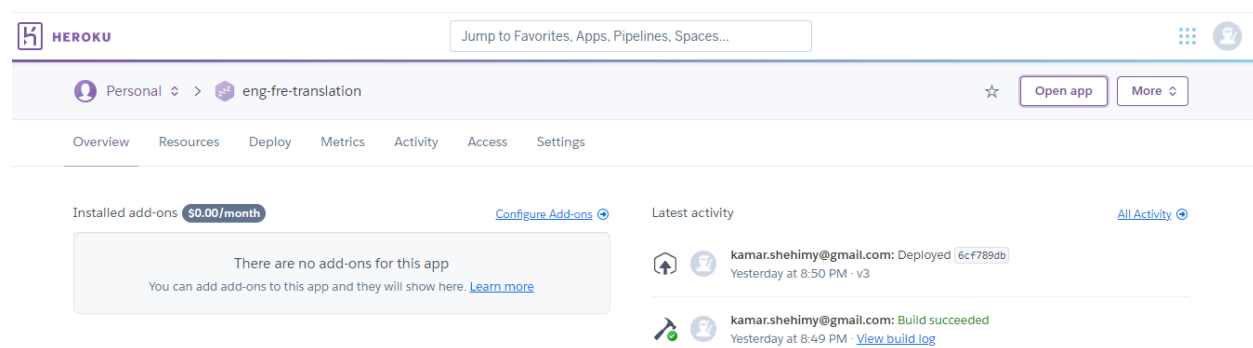
she likes apples



elle aime les pommes

Figure 82: Deploying model using Heroku

We can also open our Heroku account and click on open app



Type (<https://eng-fre-translation.herokuapp.com/>) to try the app on Heroku