Guide on Running a Local Copy of Automated Short Answer Scoring

These instructions will get a copy of the project running on a local machine. The project was written using Python 3.7.

Getting started

1. Copy or clone project.
   The project can be found on https://github.com/vemichelleve/asas.git.

   ```
   git clone https://github.com/vemichelleve/asas.git
   ```

2. Download **GloVe**.
   GloVe is used to represent words in vectors. Download from https://www.kaggle.com/thanakomsn/glove6b300dtxt and save inside asas directory.
3. Make asas as working directory.

   ```
   cd asas/
   ```

Requirements

Installing pip and venv

Make sure that the machine as **Python 3** installed. The code needs to be run inside a virtual environment to isolate package installation.

Note: Anaconda can also be used as an alternative.

1. Install package manager, **pip**, to install the packages.

   ```
   curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py

   python3 get-pip.py
   ```

2. Install **virtual environment** using pip.

   ```
   pip install virtualenv
   ```

3. Create a virtual environment.

   ```
   virtualenv --system-site-packages -p python3 ./venv
   ```

4. Activate virtual environment.

   ```
   source venv/bin/activate
   ```

Installing backend dependencies

1. Install **Django**.

   ```
   pip install django
   ```

2. Install **Django REST framework**.

   ```
   pip install djangorestframework
   ```

3. Install **django-cors-headers**.

   ```
   pip install django-cors-headers
   ```

Installing machine learning libraries

The libraries need to be installed in an isolated **virtual environment**, so make sure that it is active.

1. Install **TensorFlow**.

   ```
   pip install tensorflow==1.13.1
   ```

   This exact version of TensorFlow is required due to obsolete functions in the later version.

   For GPU usage, install TensorFlow for GPU. Version 1.13.1 requires CUDA 10.1. Read https://www.tensorflow.org/install/gpu for further information.

   ```
   pip install tensorflow-gpu==1.13.1 # For CUDA 10.1
   ```

   For CUDA 9.0, version 1.12.0 is required. This version is not available in pip, so download the file from https://pypi.org/project/tensorflow-gpu/1.12.0/#files and install with pip.

   ```
   pip install <file_path> # For CUDA 9.0
   ```

2. Install **Keras** 2.1.0.

   ```
   pip install keras=2.1.0
   ```

   This exact version is required due to obsolete functions in the newer versions that is used in this project.

3. Install **pandas**.

   ```
   pip install pandas
   pip install scikit-learn
   ```

4. Install **scikit-learn**.

5. Install **nltk**.

```
pip install nltk
```

6. Download nltk stopwords.

```
python3
>> import nltk
>> nltk.downlaod(‘stopwords’)
>> quit()
```

Installing frontend dependencies

1. Download and install **npm** and **Node.js** from https://www.npmjs.com/get-npm.
2. Change directory to `frontend/`.

```
cd frontend/
```

3. Install package dependencies using **npm**.

```
npm install
```

4. Return to parent directory.

```
cd ..
```

Running a local copy

Back-end

1. Make sure virtual environment is activated before running the back-end server.

```
source venv/bin/activate
```

2. Run Django server.

```
python3 manage.py runserver
```

The server will be at http://127.0.0.1:8000

However, if remote server is needed, the server can be changed to the machine’s IP address, for example, http://155.69.151.177:8000, by running the code below.

```
python3 manage.py runserver 0:8000
```

The server address can be found on the command line.

3. Do not close the terminal to keep the server running.

Frond-end

1. Open a new terminal.
2. Change server address (**only if remote server is used**).

Change to server IP address in `frontend/src/App.js` line 67.

```
c.setCookie('url', 'http://<IP_address>:8000', '')
```

3. Start react project.

```
cd frontend/

npm start
```

The website will be available at http://localhost:3000.