INSTALLING AND RUNNING THE APP

Install Python3 and pip

1. Update ubuntu packages

```
sudo apt-get update
```

2. Install Python3

```
sudo apt-get install python3.6
```

3. Check if Python3 is installed

```
python3.6 --version
```

4. Install pip for Python3.6

```
sudo apt-get install python3.6-pip
```

5. Install virtualenv

```
sudo pip3.6 install virtualenv
```

6. Create the virtualenv

```
virtualenv --python=python3.6 venv
```

7. Activate virtualenv using

```
. venv/bin/activate
```

Install elastic-search and python dependencies, and running elastic-search server

1. cd into the main repo.
2. Run the following commands

```
# set up keys
curl -sS https://dl.yarnpkg.com/debian/pubkey.g
pg | apt-key add -

# Update files and install default-jre
apt-get update
apt-get install default-jre -y


# Install JDK
apt-get install default-jdk -y

# Install Oracle JDK
add-apt-repository ppa:webupd8team/java
```

```
apt-get update

# Install JDK 9
apt-get install oracle-java8-installer -y
```

```
# Import the Elasticsearch PGP key
wget -qO - https://artifacts.elastic.co/GPG-KEY
-elasticsearch | apt-key add -

# Install transport-https package
apt-get install apt-transport-https -y
echo "deb https://artifacts.elastic.co/packages
/6.x/apt stable main" | tee -a /etc/apt/sources
.list.d/elastic-6.x.list

# Install elastic-search
apt-get update
apt-get install elasticsearch -y

# Running Elasticsearch with systemd
/bin/systemctl daemon-reload
```

```
/bin/systemctl enable elasticsearch.service

# Start elasticsearch
systemctl start elasticsearch.service

# Make a GET request, to check if elasticsearch
 node is running
# curl -X GET "localhost:9200/"

# Install the libraries from pip
pip install -r requirements.txt


```

2. Press `enter` when prompted.
3. You will be asked to accept the license for installing java when prompted. Press enter to accept the license

Loading the different API endpoints of ElasticSearch is working

1. Open the main folder. Change directory into ElasticData

```
cd ElasticData
```

2. Now run

```
cd TheFinalJson
curl localhost:9200/cases/_bulk -H "Content-type:app
lication/json" -X POST --data-binary @CasesFinal.jso
n
```

3. Go back using

```
cd ..
```

4. Go to `acts` and run

```
cd acts
curl localhost:9200/acts/_bulk -H "Content-type:appl
ication/json" -X POST --data-binary @Acts.json
```

5. Come back to the main repo using

```
cd ../..
```

Install Node dependecies

1. Run from the main repository

```
chmod +x run_node.sh
```

and

```
sudo run_node.sh
```

to install node dependencies.

2. cd into the

```
cd node
```

folder. Run

```
node server.js
```

Downloading the trained data and then unzipping into the tmp folder

1. Download the trained model

```
wget 0 https://storage.googleapis.com/bert_models/2018_10_18/uncased_L-12_H-768_A-12.zip
```

2. Unzip the trained model into the main repo using

```
unzip uncased_L-12_H-768_A-12.zip
```

3. Make temp_files

```
mkdir temp_files
```

in the main repo.

Running the bert server

1. Open another terminal and open the main repository.
2. cd into temp_files

```
cd temp_files
```

3. Run

```
bert-serving-start -model_dir ../uncased_L-12_H-768_
A-12 -num_worker=1 -max_seq_len=500
```

Running the nodejs server

1. Open another terminal and open the main repo.
2. Change directory to node

```
cd node
```

to cd into the node directory.

3. Run

```
node server.js
```

for starting the server.

Running the flask server

1. Open another terminal and open the main repo
2. Run

```
python server.py
```