Stance detection using different attention mechanisms.

Hardik Sahi (20743327) **Karan Vijay Singh (20745105)**

Note: The base code structure has been taken from Stanford 224n Winter 2018 course default project. (https://github.com/abisee/cs224n-win18-squad)

We have chosen to do Default project (Fake News Stance Detection) as a project for MSCI 641 (Text Analytics).
In this project, we have tried to explore the applicability of following types of attention mechanisms in NLP:

1. Dot product attention
2. Bidaf Attention
3. Dot product + self attention

The presented guide describes how to reproduce the results.

Creating Environment:
Python: 2.7
Requirements to run the code:

- colorama==0.3.9
- nltk==3.2.5
- numpy==1.14.0
- six==1.11.0
- tensorflow==1.4.1
- tensorflow-tensorboard==0.4.0
- tqdm==4.19.5

All of these requirements are specified in **requirements.txt.**

Run the script: **create_env_install_dependencies.sh**

This script does the following:

1. Creates environment named **‘stance’** with python 2.7
2. Installs all the above mentioned requirements in the environment.
3. Divides the train data into train and dev data. Preprocesses it and stores it on disk.

Training the model:

1. Open Terminal. Activate **‘stance’** environment.
2. Open code folder and run the following command:

```
python main.py --experiment_name=’folder_name_to_save_checkpoint’ --mode=train
--attention_type=’dot_product / bidaf / self_attention’ --reduction_type=’mean / max’
```

This code uses the chosen attention module and trains the model, saves checkpoint in the folder specified in --experiment_name argument.

Note: **dot_product / bidaf / self_attention modules are implemented in modules.py**

Monitoring training using tensorboard:

1. Go to experiments folder.
2. Run **tensorboard --logdir=. --port=5678**
3. Go to http://localhost:5678 to access tensorboard.

Testing the model:

1. Go to code folder and run the following command:

```
python main.py --mode=official_competition_eval
--ckpt_load_dir=’path_to_checkpoint_saved_during_training’
--experiment_name=’experiment_name_during training’ --attention_type=’dot_product /
bidaf / self_attention’ --reduction_type=’mean / max’ --result_output_path=’output_folder’
```

This code will read the competition dataset, preprocess it and then pass it through the trained model.
The predicted stance is saved in **stance.csv** file in folder **result_output_path**

For BERT code: [IN BERT folder]
Install bert-tensorflow OR git clone https://github.com/google-research/bert.git