Coursework 1: Question classification description

This readme contain 3 sections: Project Structure, Running of code, Description for each function.

1. Project Structure

```
.
├── doucument                   // description of coursework
│   ├── readme.md               // instruction.
│   ├── readme.pdf              // PDF form of readme.
├── data
│   ├── bow_model.pt            // After training bow model, store model in there
│   ├── bilstm_model.pt         // After training bilstm model, store model in there
│   ├── bilstm_ensemble_trained_model0.pt     // Store trained bilstm ensemble model0 in there
│   ├── bilstm_ensemble_trained_model1.pt     // Store trained bilstm ensemble model1 in there
│   ├── bilstm_ensemble_trained_model2.pt     // Store trained bilstm ensemble model2 in there
│   ├── bilstm_ensemble_trained_model3.pt     // Store trained bilstm ensemble model3 in there
│   ├── bilstm_ensemble_trained_model4.pt     // Store trained bilstm ensemble model4 in there
│   ├── bow.config              // bow model configuration
│   ├── bilstm.config           // bilstm model configuration
│   ├── bilstm_ensemble.config  // bilstm_ensemble model configuration
│   ├── train_5500.txt          // dataset(will be splited into train set and dev set)
│   ├── glove.small.txt         // glove pre-train word embedding
│   ├── test.txt                // test set
│   ├── output.txt              // After testing whatever model, store output results in there
├── src                         // source code
│   ├── question_classifier.py  // source code of system
.
```

2. How to use the code

To run this system, you must change the directory to `src` folder. Then set the parameters in `data/bow.config`, the most important of these is `is_Pretrain=False` or `True`, which determine whether using glove pretrain word embedding. Hints: training models costs some times, please be patient

2.1 Training and testing Bow model

We recommend that you set some parameters (in bow.config) like below：

bow randomly initialize word embedding	bow pretrain word embedding
is_Pretrain=False	is_Pretrain=True
learning_rate=0.1	learning_rate=0.1
epoches = 5	epoches = 20

Then to train the `bow` model, you can use the command below:

```
python3 question_classifier.py train -config ../data/bow.config
```

Then you can test the bow model after the above step, you can use the command below:

```
python3 question_classifier.py test -config ../data/bow.config
```

2.2 Training and testing Bilstm model

We recommend that you set some parameters (in bilstm.config) like below：

bow randomly initialize word embedding	bow pretrain word embedding
is_Pretrain=False	is_Pretrain=True
learning_rate=0.1	learning_rate=0.1
epoches = 5	epoches = 5

Then you can train the `bilstm` model, you can use the command below:

```
python3 question_classifier.py train -config ../data/bilstm.config
```

Then you can test the bilstm model after the above step, you can use the command below:

```
python3 question_classifier.py test -config ../data/bilstm.config
```

2.3 Bonus: Training and testing Bilstm Ensemble model

We recommend that you set some parameters (in bilstm_ensemble.config) like below：

bow randomly initialize word embedding	bow pretrain word embedding
is_Pretrain=False	is_Pretrain=True
learning_rate=0.1	learning_rate=0.1
epoches = 3	epoches = 3
ensemble_size=5	ensemble_size=5

To train the `bilstm ensemble` model, you can use the command below:

```
python3 question_classifier.py train -config ../data/bilstm_ensemble.config
```

Then you can test the bilstm ensemble model after the above step, you can use the command below:

```
python3 question_classifier.py test -config ../data/bilstm_ensemble.config
```

3. Description for each function

Here, we only describle some main functions, the detailed information can be seen fome the comments in the source code.

```
def load_dataset(data_path):
    '''
    1. This function load the dataset from
    https://cogcomp.seas.upenn.edu/Data/QA/QC.
    And do some preprocessing: data cleaning, removing stop words,
    and refactoring data structure.
    Then return preprocessed data_set.
    '''

def build_random_vocabulary(word_appear_times):
    '''
    2. This function build the vocabulary for the randomly initialize word
    embeddings method. It also add word: #UKN# in vocabulary.
    You can set word_appear_times value to select words that appearing at
    least kword_appear_times times in the training set.
    Then function return vocabulay.
    '''
```

```
def load_glove(glove_path):
    '''
    3. This function load the glove pre-trained embeddings, and pruning
    pretrained embeddings by removing the words that do not appear in the dataset.
    Then function return the vocabulay for glove pretraining word embeddings.
    '''

def spliteDataset(validation_size):
    '''
    4. This function splite dataset into training and development subset.
    Default validation_size=0.9 which means 9 portions are for training, and the other is for development.
    Then function return the train_set and dev_set
    '''

def word_embedding(is_Pretrain,is_pre_freeze):
    '''
    5. This function realize two kinds of wording embedding by using pytorch:
    First kind is randomly initialize it by  using function below:
    embedding=nn.Embedding(VOCAB_SIZE, WORD_DIM)
    Second kind is using glove pretrained weights. You can set whether using freeze:
    embedding = nn.Embedding.from_pretrained(weights, freeze=is_pre_freeze)
    Both of them return the vector representation of a word.
    '''

def make_bow_vector(sentence, vocabulary):
    '''
    6. This function turns the sentence into a vector form by adding up the vectors for all
    the words and divide by the number of words in the sentence.
    It returns the vector for a sentence.
    '''

def bow_train(bow_model):
    '''
    7. This function trains bow model.
    It returns bow model for testing
    '''

def bow_test():
    '''
    8. This function tests bow model.
    It returns accuracy of test set based on bow model.
    '''

def bilstm_train(bilstm_model):
    '''
    9. This function trains bilstem model.
    It returns bilstm model for testing
    '''

def bilstm_test():
    '''
    1o. This function tests bilstm model.
    It returns accuracy of test set based on bilstm model.
    '''

def bilstm_ensemble_train():
    '''
    1o. This function train bilstm ensemble model.
    It returns bilstm ensemble model for testing
    '''

def bilstm_ensemble_test():
    '''
    1o. This function tests bilstm ensemble model.
    It returns accuracy of test set based on bilstm ensemble model.
    '''

def select_operation():
    '''
```

```
    11. This function checks what command you input in the terminal,
    then determine whether training or testing what model.
    '''
```

/