UDACITY

‹ Return to Classroom

Generate TV Scripts

REVIEW

CODE REVIEW

HISTORY

▸ problem_unittests.py

▾ helper.py

```python
import os
import pickle
import torch


SPECIAL_WORDS = {'PADDING': '<PAD>'}


def load_data(path):
    """
    Load Dataset from File
    """
    input_file = os.path.join(path)
    with open(input_file, "r") as f:
        data = f.read()

    return data


def preprocess_and_save_data(dataset_path, token_lookup, create_lookup_tables):
    """
    Preprocess Text Data
    """
    text = load_data(dataset_path)

    # Ignore notice, since we don't use it for analysing the data
    text = text[81:]

    token_dict = token_lookup()
    for key, token in token_dict.items():
        text = text.replace(key, ' {} '.format(token))

    text = text.lower()
    text = text.split()

    vocab_to_int, int_to_vocab = create_lookup_tables(text + list(SPECIAL_WORDS.values
    int_text = [vocab_to_int[word] for word in text]
    pickle.dump((int_text, vocab_to_int, int_to_vocab, token_dict), open('preprocess.


def load_preprocess():
    """
    Load the Preprocessed Training data and return them in batches of <batch_size> or
    """
    return pickle.load(open('preprocess.p', mode='rb'))


def save_model(filename, decoder):
    save_filename = os.path.splitext(os.path.basename(filename))[0] + '.pt'
    torch.save(decoder, save_filename)


def load_model(filename):
    save_filename = os.path.splitext(os.path.basename(filename))[0] + '.pt'
    return torch.load(save_filename)

```

RETURN TO PATH