```
# Author: Shashi Narayan
# Date: September 2016
# Project: Document Summarization
# H2020 Summa Project
####################################

"""
My flags
"""

from __future__ import absolute_import
from __future__ import division
from __future__ import print_function

import numpy as np
import tensorflow as tf

########### ============ Set Global FLAGS ============= #############

### Temporary Directory to avoid conflict with others

# VERY IMPORTANT # : SET this directory as TMP by exporting it

tf.app.flags.DEFINE_string("tmp_directory", "/tmp", "Temporary directory used by rouge code.")

tf.app.flags.DEFINE_string("use_gpu", "/gpu:3", "Specify which gpu to use.")
```

```
### Global setting

#tf.app.flags.DEFINE_string("exp_mode", "train", "Training 'train' or Test 'test' Mode.")

tf.app.flags.DEFINE_string("exp_mode", "train", "train")

tf.app.flags.DEFINE_integer("model_to_load", 100, "Model to load for testing.")

tf.app.flags.DEFINE_boolean("use_fp16", False, "Use fp16 instead of fp32.")

tf.app.flags.DEFINE_string("data_mode",  "cnn", "cnn or dailymail or cnn-dailymail")

### Pretrained wordembeddings features

tf.app.flags.DEFINE_integer("wordembed_size", 200, "Size of wordembedding (<= 200).")

tf.app.flags.DEFINE_boolean("trainable_wordembed", False, "Is wordembedding trainable?")
# UNK and PAD are always trainable and non-trainable respectively.

### Sentence level features

tf.app.flags.DEFINE_integer("max_sent_length", 100, "Maximum sentence length (word per sent.)")

tf.app.flags.DEFINE_integer("sentembed_size", 350, "Size of sentence embedding.")

### Document level features

tf.app.flags.DEFINE_integer("max_doc_length", 110, "Maximum Document length (sent. per document).")

tf.app.flags.DEFINE_integer("max_title_length", 0, "Maximum number of top title to consider.") # 1

tf.app.flags.DEFINE_integer("max_image_length", 0, "Maximum number of top image captions to consider.") # 10

tf.app.flags.DEFINE_integer("target_label_size", 2, "Size of target label (1/0).")
```

```
### Convolution Layer features

tf.app.flags.DEFINE_integer("max_filter_length", 7, "Maximum filter length.")
# Filter of sizes 1 to max_filter_length will be used, each producing
# one vector. 1-7 same as Kim and JP. max_filter_length <=
# max_sent_length

tf.app.flags.DEFINE_string("handle_filter_output", "concat", "sum or concat")
# If concat, make sure that sentembed_size is multiple of max_filter_length.
# Sum is JP's model

### LSTM Features

tf.app.flags.DEFINE_integer("size", 600, "Size of each model layer.")

tf.app.flags.DEFINE_integer("num_layers", 1, "Number of layers in the model.")

tf.app.flags.DEFINE_string("lstm_cell", "lstm", "Type of LSTM Cell: lstm or gru.")

### Encoder Layer features

# Document Encoder: Unidirectional LSTM-RNNs
tf.app.flags.DEFINE_boolean("doc_encoder_reverse", True, "Encoding sentences inorder or revorder.")

### Extractor Layer features

tf.app.flags.DEFINE_boolean("attend_encoder", False, "Attend encoder outputs (JP model).")

tf.app.flags.DEFINE_boolean("authorise_gold_label", True, "Authorise Gold Label for JP's Model.")

### Reinforcement Learning

tf.app.flags.DEFINE_boolean("rouge_reward_fscore", True, "Fscore if true, otherwise recall.") # Not used, always use fscore

tf.app.flags.DEFINE_integer("train_epoch_wce", 20, "Number of training epochs per step.")

tf.app.flags.DEFINE_integer("num_sample_rollout", 10, "Number of Multiple Oracles Used.") # default 10
```

```
### Training features

tf.app.flags.DEFINE_string("train_dir", "C:\\Users\\Jungwoo Lim\\Documents\\GitHub\\Refresh\\address\\to\\training\\directory", "Training directory.")

tf.app.flags.DEFINE_float("learning_rate", 0.001, "Learning rate.")

tf.app.flags.DEFINE_boolean("weighted_loss", True, "Weighted loss to ignore padded parts.")

tf.app.flags.DEFINE_integer("batch_size", 20, "Batch size to use during training.")

tf.app.flags.DEFINE_integer("training_checkpoint", 1, "How many training steps to do per checkpoint.")

###### Input file addresses: No change needed

# Pretrained wordembeddings data

tf.app.flags.DEFINE_string("pretrained_wordembedding","C:\\Users\\Jungwoo Lim\\Documents\\GitHub\\Refresh\\address\\data\\1-billion-word-language-modeling-benchmark-r13output.word2vec.vec","Pretrained wordembedding file trained on the one million benchmark data.")

# Data directory address

tf.app.flags.DEFINE_string("preprocessed_data_directory", "C:\\Users\\Jungwoo Lim\\Documents\\GitHub\\Refresh\\address\\data\\preprocessed-input-directory",
                        "Pretrained news articles for various types of word embeddings.")

tf.app.flags.DEFINE_string("gold_summary_directory",
                        "C:\\Users\\Jungwoo Lim\\Documents\\GitHub\\Refresh\\address\\data\\Baseline-Gold-Models",
                        "Gold summary directory.")

tf.app.flags.DEFINE_string("doc_sentence_directory",
                        "C:\\Users\\Jungwoo Lim\\Documents\\GitHub\\Refresh\\address\\data\\CNN-DM-Filtered-TokenizedSegmented",
                        "Directory where document sentences are kept.")

############ Create FLAGS
FLAGS = tf.app.flags.FLAGS
```