"""

Candidate architectures for each task's.

"""

from \_\_future\_\_ import annotations

from typing import Dict

from .features import get\_features

from .graph import Graph

from .lib.model import ClemNet

from .params import ModelTypeEnum

import tensorflow as tf

class MagicRecsClemNet(Graph):

def get\_logits(self, features: Dict[str, tf.Tensor], training: bool) -> tf.Tensor:

with tf.name\_scope("logits"):

inputs = get\_features(features=features, training=training, params=self.params.model.features)

with tf.name\_scope("OONC\_logits"):

model = ClemNet(params=self.params.model.architecture)

oonc\_logit = model(inputs=inputs, training=training)

with tf.name\_scope("EngagementGivenOONC\_logits"):

model = ClemNet(params=self.params.model.architecture)

eng\_logits = model(inputs=inputs, training=training)

return tf.concat([oonc\_logit, eng\_logits], axis=1)

ALL\_MODELS = {ModelTypeEnum.clemnet: MagicRecsClemNet}