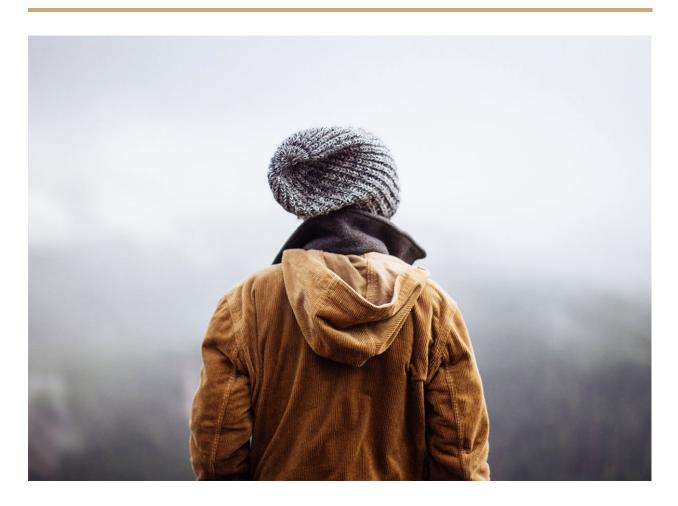
何骥实习总结

化学反应逆合成

3N-MCTS



简介

原论文(3N-mcts)

3N (expand network, in scope filter, rollout network) mcts (蒙特卡洛树搜索)

已完成项目

Expand network 已完成,可用。

In scope network 已完成 ,可用。

Rollout network 已完成,不可用(精确度不高,可能是数据问题)。
MCTS 已完成, 可用。

实际上可以不用rollout network也能实现逆合成。

借鉴最新版alphago,即alphago zero 的思路,一个网络既作为 expand network 也作为 rollout network。直接利用贪心算法来 rollout,一走到底。我的模型设计就是基于此 (丢掉rollout network, 只保留expand network 和 in scope filter)

不足之处:

没有像论文中把expand network 和 in scope filter整合成一个pipeline, 即 expand network 输出作为 in scope filter 的输入。

模型接口及使用(详见dev.ipynb):

1. mcts_node.py 包含以下两个类:

class State(object):

状态类代表了逆反应合成的状态,它的一个属性叫mols, mols 里面存储的是其逆合成的分子,原始状态的mols里面只有产物一个分子。随着树搜索的进行,mols里的分子会愈来愈多。我们最终希望mols里的分子都在build block mols里面。

class MCTSNode(object):

这个类是MCTS的节点类,是MCTS的基础设施。它有几个重要的参数, 分别是

state 状态类, class State

parent_local_trans_id 母节点局部变换ID, int

local_global_trans_maps 当前节点局部变换到全局变换映射, list

local_trans_mol_maps 当前节点局部变换到分子的映射, list

parent 母节点, class MCTSNode

注:

母节点局部变换ID: 母节点变到当前节点所对应变换的局部ID

局部ID通过 local_global_trans_maps 可以变换到全局ID

局部ID通过 local_trans_mol_maps 可以对应到相应参与反应的分子

2. mcts_policy.py 主要包含

class MCTS_Policy_old(object):

其参数主要有:

expand_network

in_scope_filter

lbl1 expand rule label encoder

lbl2 expand network relabel encoder

exp_indexes 用到的扩展网络的特征列

num_readouts 树搜索深度(rollout 深度)

这部分是整个MCTS的核心,而这部分的核心则是

tree_search(self, build_block_mols, parallel_readouts=**None**),

其中 parallel_readouts指的是同时并行处理的节点数。这个函数是alphago zero中mcts核心部分。

expand_network.py & In scope filter.py

这部分根据原论文编写。详细信息见原论文。

其中expand_network要和MCTS整合在一起的话需要一个接口, 这个借口 就是run_many这个函数

Retrosynthetic.py

这部分是在逆合成时需要直接调用的部分。

它的具体使用见 dev.ipynb

最后关于改进的建议:

- 1. 不建议去纠结rollout network,毕竟最新版的alphago都抛掉了快速走子网络,转而使用expand network,逆其道而行之不好。
- 2. 把 expand network 和 inscope filter 整合成一个pipeline。
- 3. 好好定义building block mols, 这方面我真的有点方。
- 4. **这点比较重要** single_step_synthesis 函数(在reaction.py里)每次只返回一个mol, 这是不对的,应该返回多个mol (A--->B+C), 况且我的 MCTS算法可以同时处理多个mols,不用担心MCTS处理不了。

最后祝大家开发快乐。