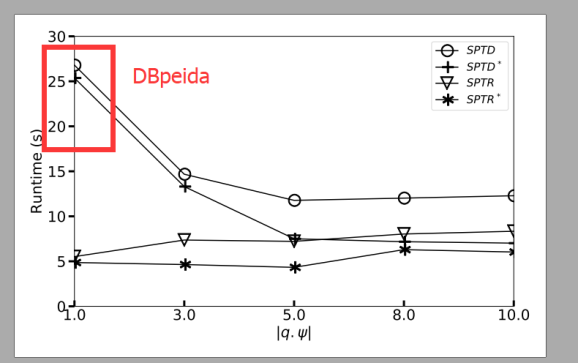
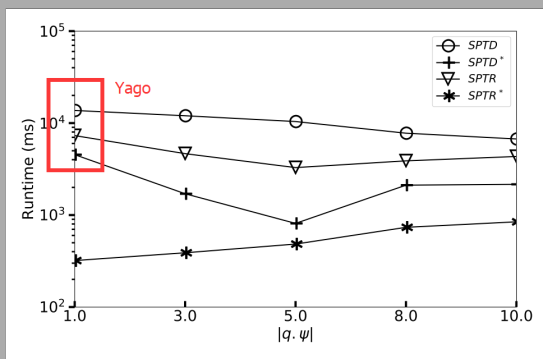
1. 对于Yago和DBpedia，在num\_word=1是时，单个时间查询平均时间都很大，通过分析记录的数据，发现当nw=1时，部分查询样本计算GetSemanticTree的次数比其他查词数计算GetSemantic多许多，同时在GetSemanticTree前，会根据可达性计算最小时间差，对于Yago，TF-label的不稳定性，会造成部分样本查询时间长。综上，nw=1时间长，是因为部分样本计算最小时间差花时多，GetSemantic次数多造成。之所以会出现这种情况，我觉得是因为，nw=1时，当已经找到了k个tree，但由于第k个阈值较大，而该词的最小时间差\*最短路径又较小，没法排除对应pid，造成GetSemanticTree次数多；而对于查词数到一定数时，当已经找到k个tree时，即使第k个阈值较大，可能word1的最小时间差\*最短路径较小，但是word2最小时间差\*最短路径较大，而我们计算的是多个word的和，所以更有可能出现计算出的score总和大于阈值，而剪掉对应的pid。



1. 对于Yago和DBpedia的时间范围查找，改变查询的时间范围，Yago没有什么变化，而DBpedia有明显变化，分析记录的数据，发现无论是Yago还是DBpedia，查询时间范围小时，部分样本GetSemantic次数会较多一些，而Yago没什么变化是因为yago一般一次 GetSemantic很快（yago图连通性比较弱），而DBpedia一般一次 GetSemantic很慢（DBpedia图连通性较强）。然后为什么查词范围小时，GetSemantic会变多，我猜测是因为，查词范围小时，pid附近的满足条件的节点少，这意味着可能要遍历更深以找到符和条件的tree，而遍历深意味着阈值更大，也就GetSemantic次数更多些。

