# 近似连接实验报告

**2011011258　计 13　　谭志鹏**

## 一、　实验目的

基于 q-gram 实现两个文本的进行连接，分别实现编辑距离和 Jaccard 距离的两个函数。

## 二、　实验算法

主要使用了前缀匹配加倒排索引等方法。

具体的，对于文本的存储，首先分别读入两个文件内容到 file1_str, file2_str, 然后扫描建立其 q-gram 的向量，file1_grams, file2_grams。

```
vector<string> file1_str;
vector< vector<string> > file1_grams;
vector<string> file2_str;
vector< vector<string> > file2_grams;

map< string, vector<int> > file2_idx;
```

对两个 grams 向量的每一行依照全局的出现次数（df）进行升序排序（使用快排算法），即出现次数多的 gram 被拍到后面。然后分别对于编辑距离和 Jaccard 距离需要满足的前缀 prefix 长度，对 file2 的前缀部分建立倒排列表，并且去除了前缀部分的重复 gram。函数如下：

```
void SimJoiner::
createEdPrefixId(vector< vector<string> >& file_grams,
          vector<string>& file_str, map< string, vector<int> >& file_idx)
{
    int pre_gramnum = edthre * gramlen + 1;     //前缀长度
    for(int i = 0; i < file_grams.size(); i ++)
    {
        int gramnum = file_grams[i].size();
        int num = min(pre_gramnum, gramnum);

        vector<string>& vgram = file_grams[i];
        for(int j = 0; j < num ; j ++)
        {
            string gram = vgram[j];
            vector<int>& tvec = file_idx[gram];
            if(tvec.empty() || tvec[tvec.size() - 1] != i) //第二个条件排除重复部分
                tvec.push_back(i);
        }
    }
}
```

最后查找时，扫描 file1 中每一个串，取其相应 ED 或 Jaccard 前缀长度在 file2 中倒排表 file2_idx 中查询，非空的即是 candidate，然后调用相应编辑距离和 Jaccard 距离函数进行验证筛选。

对于编辑距离，尝试过进行第二轮的 count filter 进行筛选，不过发现在测试系统上会超时，可能由于进行 count filter 太费时，剪枝效果不好造成的。

## 三、 总结

通过本次试验对于数据库近似连接算法有了实际的尝试，发现仅仅使用前缀过滤的方法速度还是比较慢。尝试过 ppt 上介绍的其他包括 count filter 等方法，效果不理想，因此这个问题还有待进一步的算法优化。需要改进的地方还很多。