山东大学 计算机科学与技术 学院

信息检索与数据挖掘课程实验报告

学号：201600130050	姓名： 张涵	班级： 人工智能班

实验题目：vsm 处理文件

实验内容：

预处理文本数据集，并且得到每个文本的 VSM 表示

实验过程中遇到和解决的问题：

（记录实验过程中遇到的问题，以及解决过程和实验结果。可以适当配以关键代码辅助说明，但不要大段贴代码。）

实验中，先使用 CreateDictionary（）方法生成特征词典（特征词典及之后的所有生成文件全部在项目的 res 文件夹内），特征词典的生成方法是提取特征词并将其插入特征词典中：

```
while (!fin.eof()) {
    if (ch == '\n') { ch = fin.get(); continue; }
    strcpy_s(str1, getterm(ch));
    for (int i = 0; i<dictionarysize; i++)          //提取特征词后,扫描特征词典,若词典中存在此特征词,则置标志位为1
        if (!strcmp(TD.TermArray[i].t, str1)) { flag1 = 1; break; }
    ch = fin.get();
    strcpy_s(str2, gettype(ch));
    for (int i = 0; i<50; i++)                      //提取特征词词性以后,扫描不纳入特征词典的词性数组,若存在此词性,则置标志位为1
        if (!strcmp(str2, type[i])) { flag2 = 1; break; }
    if (!flag1 && !flag2) {                  //若两个标志位均为0,则把特征词id,特征词,词性插入特征词典
        TD.TermArray[dictionarysize].term_ID = id;
        strcpy_s(TD.TermArray[dictionarysize].t, str1);
        strcpy_s(TD.TermArray[dictionarysize].type, str2);
        dictionarysize++;
        id++;
    }
```

其次是生成倒排索引表：

```
//索引项
struct Index {
    int term_ID;                //特征词标号
    int TF[DOCUMENTMAXSIZE];    //词频数组
    int ni;                     //词i在段落中出现的段落数目
};
//类，倒排索引表
class IndexTable {
public:
    struct Index IndexArray[DICTIONARYMAXSIZE]; //索引项数组
    int documentsize;                           //文档的个数
public:
    IndexTable() { documentsize = 0; }                  //构造函数
    void CreateIndexTable();                           //倒排索引表生成
}IT;
```

倒排索引表中需要计算词频 TF：

```
for (i = 0; i<TD.dictionarysize; i++) {                    //计算词频TF
    fin.open("../res/58343.txt", ios::in, 0);
    if (!fin) {
        cout << "打不开文件--文档集.txt!\n";
        exit(0);
    }
    IT.IndexArray[i].term_ID = TD.TermArray[i].term_ID;
```

而在扫描过程中，还要判断下一个读到的字符是什么，如果是文本结束符 EOF，则结束扫描，如果是回车符，则扫描下一行。

之后还要生成向量空间模型（CreatVSM），这也是最重要的一步，不仅要计算向量元素平方和和向量模，还要注意取 idfi=log（N/ni），最后生成 vsm.arff 文件：

```
for (int i = 0; i<TD.dictionarysize; i++) {            //计算并打印向量空间模型
    for (int j = 0; j<IT.documentsize; j++) {
        VSM.Vector[i][j] = TFIDF[i][j] / MOD[j];       //计算新的向量元素
        sprintf_s(s, "%15.10f", VSM.Vector[i][j]);
        fout << s;
        //fout<<"   "<<VSM.Vector[i][j]<<"  ";
    }
    fout << endl;
}
```

结论分析与体会：

本次实验最大的体会就是，程序写的太烂了，很多地方可以优化，但是我又不知道该怎么去优化。

使用 c++来写数据处理真的比 Python 要难的多，可我又不会用 Python，目前还在学习之中，仍做不到使用 Python 来实现程序功能，所以只好使用 c++来写第一个实验，希望在第二个实验能够使用 Python 来做。