《自然语言处理》实验报告

陈子谦 10235501454

1. 实验目标

- 实现基于最大正向匹配(最大长度为7)的中文分词系统，实现对自然语言文本的自动切分。
- 给出前10个中文高频项的分词结果：读取`cn_stopwords.txt`停用词字典，依据占据全文分词总量（去除标点及停用词）的概率降序依次输出

2. 实验结果（前10个高频词项统计结果）

NLP > output.txt

```
1   民族 => 38 (0.0447)
2   中华文明 => 28 (0.0329)
3   文化 => 27 (0.0318)
4   统一 => 23 (0.0271)
5   统一性 => 20 (0.0235)
6   发展 => 17 (0.0200)
7   一体 => 17 (0.0200)
8   中华民族 => 17 (0.0200)
9   形成 => 15 (0.0176)
10  多元 => 13 (0.0153)
11
```

3. 代码实现

3.1 词典加载模块

本模块首先打开词典文件，然后逐行读取文件内容，将每个词条存入一个动态分配的数组中。采用自动扩容机制，能够根据实际词典大小动态调整数组空间，避免内存浪费同时支持大词典的加载。过程中对打开文件、读取内容、内存分配等各个环节都加入了错误检查，确保在任何异常情况下都能及时提示并安全退出。最终，所有后续用于匹配的词条均以字符串形式存储在内存中。

3.2 语料读取模块

语料读取模块负责从指定文本文件中按行读取数据。每读取一行文本，就按顺序将其加入到一个动态数组中，保证文本中所有句子都能被完整保存，供后续分词算法调用。模块内部同样使用自动扩容技术，使得即使面对较大规模的文本文件，也能灵活处理而不会因预设数组空间不足导致数据丢失。与此同时，该模块配备了详细的错误处理机制，确保在文件读取过程中，一旦出现问题能够及时释放资源并给出清晰错误提示，从而提高程序的健壮性。

3.3 分词处理模块

匹配策略：
从文本起始位置（变量`start`）出发，逐步尝试不同长度的子串（变量`match_len`），其最大长度由常量`MAX_CHINESE_BYTES`定义（例如最多支持7个汉字）。每次提取的子串（例如变量`temp`）首先进入候选检查范围。例如：

```
int start = 0;
while (start < text_length) {
    for (int match_len = MAX_CHINESE_BYTES; match_len > 0; match_len--) {
        char temp[MAX_CHINESE_BYTES+1] = {0};
        strncpy(temp, text + start, match_len);
        // 检查 temp 是否在词典中，并非停用词
    }
    // 若没有匹配项则降级为单字符切分
}
```

这里，通过逐渐降低匹配长度，确保即使最长匹配失败，也能降级至单字符切分，防止漏分。

标点处理：
由于对于单个标点的导入、匹配未出现无法正确识别“、《”这样的符号粘连情况，模块内部预先维护了一个静态结构体数组，列出所有中英文常见标点（例如 "，"、"。"、"！" 等）。在处理过程中，首先调用 `check_punctuation` 函数判断当前位置是否为标点；如果是，直接将 `start` 移过该标点，以避免因标点干扰产生错误匹配。

词典与停用词匹配：
当候选词被提取后，程序会进一步判断该词是否存在于词典中，并检查是否为停用词。只有当 `is_stopword` 返回 0 时，该词项才被视为合法，并调用 `add_token_count` 进行词频统计。

3.4 词频统计模块

实时统计：
当分词处理模块确定一个词项有效后，便调用 `add_token_count` 函数记录该词项。该函数首先扫描一个全局统计数组（类型 `wordCount`），检查该词项是否已存在；若存在则将对应计数增加，否则将该词项添加到数组中。为保证内存安全，统计模块采用了动态数组和自动扩容策略。

排序与输出：
文本处理完毕后，系统将调用快速排序（使用 `qsort` 函数）对全局词频数组进行降序排列。

3.5 结果输出模块

在完成所有文本处理和词频统计后，系统将排序后的高频词项和各自出现的概率以格式化的文本形式写入输出文件中。

3.6 创新点

1. **动态内存管理与自动扩容：**
   系统中所有模块均采用动态数组以及自动扩容策略，无论是词典、语料还是词频统计，都能够根据实际数据量自动调整内存空间，从而保证程序在各种负载情况下都能稳定运行。这种设计极大提高了系统的灵活性和鲁棒性。
2. **精确的UTF-8编码及标点处理：**
   考虑到中文文本的特殊性，系统专为UTF-8编码设计了字符长度判断方法，确保对每个字符都能正确识别。同时，通过静态数组详细列出中英文标点，实现了对标点的精准过滤，避免了因标点干扰而导致的匹配误差，从而保障了分词的准确性。