**2021년 2학기**

**프로그래밍과 문제해결**

**Assignment #2**

**담당교수: 윤은영**

**학번: 20217016**

**학과: 무은재학부**

**이름: 최대현**

**POVIS ID: daehyeonchoi**

**명예서약(Honor Code)**

**“나는 이 프로그래밍 과제를 다른 사람의 부적절한 도움 없이 완수하였습니다.”**

**Sturcture Chart**

이번 프로그램의 structure chart는 다음과 같다.

테이블이(가) 표시된 사진

자동 생성된 설명

**알고리즘**

* **Psuedocode**

1. Train, test. Exit 중의 한 가지 커맨드를 입력한다.
2. **2-1, train 커맨드를 입력할 경우)**

Train 하기 위한 파일 이름을 입력받아 읽기모드로 파일을 연다.

Train할 파일은 label \t text 구조로 제작되어 있고, 여기서 label과 text를 분리하여 각 단어의 빈도수를 체크하고 정렬하여 stats.txt 파일에 저장한다.

**2-2. test 커맨드를 입력할 경우)**

Stats. Txt 파일을 읽기모드로 열어서 train된 결과를 불러온다.

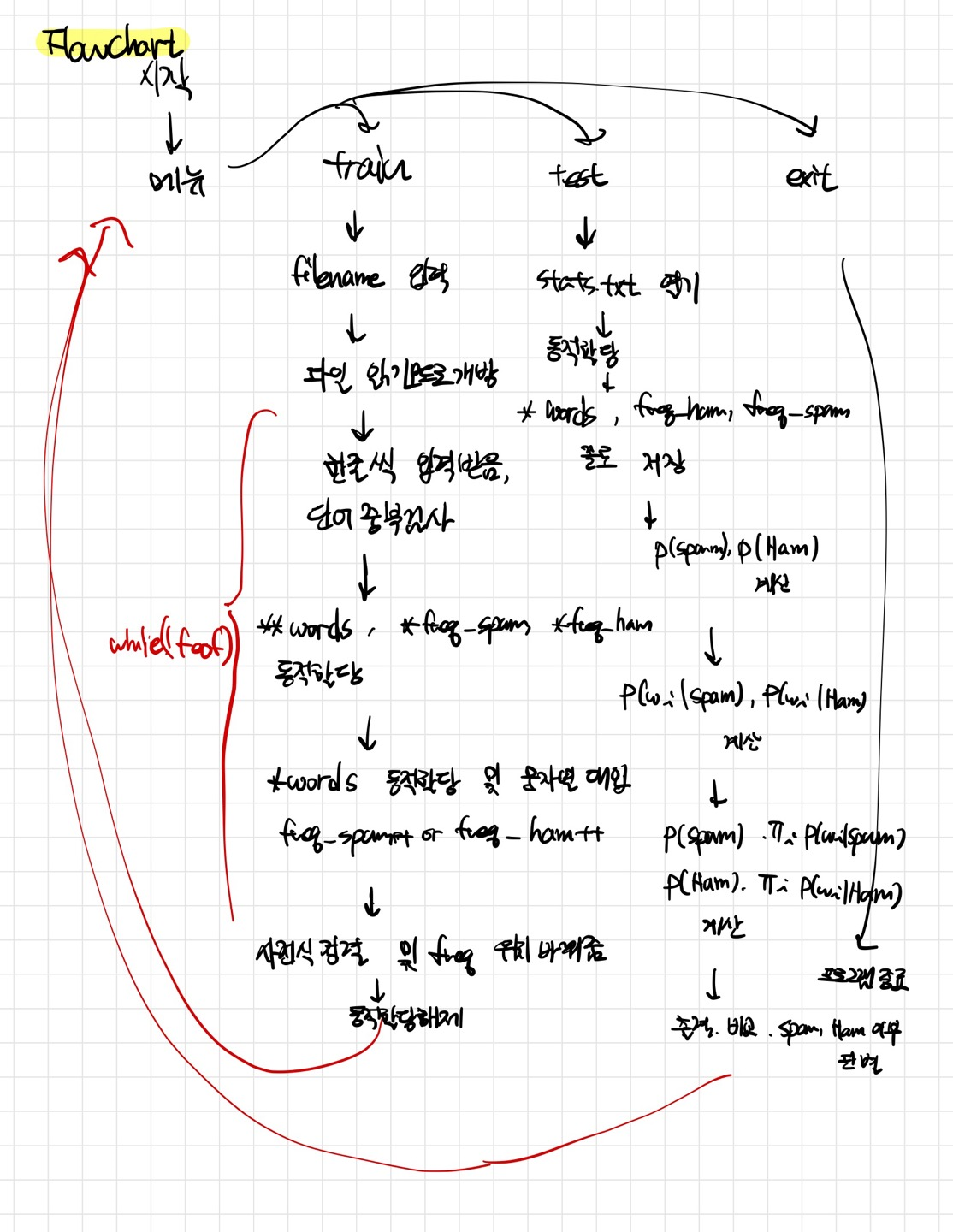
사용자에게 test할 문장을 입력받아 (ex: buy our new phone) train한 데이터를 바탕으로 조건부확률을 계산하여, spam인지 ham인지 판단한다.

**2-3. exit 커맨드를 입력할 경우)**

프로그램을 종료한다.

* **Flowchart**

이번 프로그램의 대략적인 Flowchart는 다음과 같다.



**함수 설명**

1. **void print\_menu()**

* menu를 출력해주는 함수이다.

1. **char\* process\_str(char\* str)**

* ASSN 3-1 에서 구현한 mystrlower 함수에 특수문자를 공백으로 바꿔주는 기능을 추가로 구현한 함수이다.

1. **void dic\_sort(char\*\* words, int cnt\_word, int\* freq\_spam, int\* freq\_ham)**

* words를 사전식으로 정렬하고, 그에 맞춰 freq 배열도 위치를 바꿔주는 함수이다.

1. **void train(char\*\* words, int\* freq\_spam, int\* freq\_ham, char\* fname, int\* ham, int\* spam)**

* train의 기능을 구현한 함수이다.

**프로그램 실행 방법 및 예제**

1. **초기 선택 메뉴**

초기 선택 메뉴에서 command를 입력받는다. Train, test, exit 의 유효한 3개의 커맨드가 있고, 그 이외의 커맨드를 입력받으면 error 문구를 출력하고 다시 입력받는다.

텍스트이(가) 표시된 사진

자동 생성된 설명

* 1. **train를 입력받을 경우**

Train 함수를 호출하여 train을 시작한다.

먼저, 파일의 이름을 입력받는다.

텍스트이(가) 표시된 사진

자동 생성된 설명

이 때, 입력한 파일을 읽기모드로 열어주고, 정상적으로 열리지 않는다면 error 문구를 출력한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

파일이 정상적으로 열렸다면, train을 완료한 결과를 출력하는 동시에 같은 내용을 stats.txt에 fprintf 함수를 통해 저장한다. 맨 윗 줄은 Ham과 Spam인 label의 개수이고, 그 다음부터는 각각 사전식으로 정렬한 단어, 그 단어의 ham에서의 빈도, spam에서의 빈도 순으로 출력한다.그 뒤 동적할당을 해제한다.

그 다음 enter(그 이외의 입력은 고려하지 않음) 를 입력받으면 메인화면으로 돌아간다.

텍스트이(가) 표시된 사진

자동 생성된 설명

* 1. **Test를 입력받을 경우**



Test라는 command를 입력받으면 stats.txt 를 읽기 모드로 열어준다. 그 뒤 test할 문장을 fgets 함수를 통해 입력받는다. 이 때, 입력받은 문장에서 중복되는 단어가 있는 경우 한 번에 대해서만 확률값을 계산한다.

가장 먼저 p(ham), p(spam)을 입력받아 계산하고, 그 뒤 유효한 단어(stats.txt 속 데이터와 겹치는 단어들)에 대해서 확률 값을 계산하고, 그 결과를 모두 곱해서 최종적인 확률을 계산하여 출력한다. 출력이 완료되면 동적할당을 해제해준다. enter키를 입력받을 때까지 대기하며, 입력받은 뒤에는ㅁ 메인화면으로 돌아간다.

텍스트이(가) 표시된 사진

자동 생성된 설명

* 1. **Exit을 입력받을 경우**

텍스트이(가) 표시된 사진

자동 생성된 설명

프로그램을 종료한다. (동적할당에 대한 해제는 train 단계와 test 단계에 각각 해주므로 고려하지 않아도 됨)

* 1. **그 외의 command를 입력받을 경우** 텍스트이(가) 표시된 사진

     자동 생성된 설명

오류 메시지를 출력하고 메인 화면으로 돌아가 command를 다시 입력받는다.

**토론**

문자열을 사전식으로 정리하는 부분의 알고리즘이 strcmp의 알고리즘을 그대로 따라가서 제작하기 수월했다. Test 단계에서는 동적할당을 원활히 하고, 해제도 원활히 해 주었지만 train 단계에서 calloc 함수를 사용해서 동적할당을 할 때 이론적으로는 strlen(단어)+1 만큼의 공간만큼 동적 할당을 해 주면 되지만 계속해서 알 수 없는 오류가 발생하여 strlen(단어)+4 만큼을 동적할당 해주었다. 이 오류의 원인을 찾고 싶었으나 찾지 못해서 아쉽다.

**결론**

이번 assignment를 통하여 문자열을 다루는 여러 사용자 정의 함수를 제작해볼 수 있었고, string의 길이만큼 메모리를 동적 할당하여 저장 공간을 효율적으로 사용하는 실습을 해 볼 수 있었다.

**개선 방향**

사용자 정의 함수를 더 많이 사용해서 코드의 가독성을 높이고 코드를 더 간단하게 만들 수 있을 것 같아서 개선해보고 싶다. 특히, train 부분은 함수로 만든 것에 비해 test 부분을 메인 함수에 구현해서 조금 아쉽다.