

COMP 201 - Fall 2021

Assignment 2 - Strings: Spam Detection

Assigned: 21 October 2021 23:59, Due: 4 November 2021 23:59

Efehan Güner (efehanguner21@ku.edu.tr) is the lead person for this assignment.

1 Introduction

The purpose of this assignment is to advance your skills on manipulating strings and learn to use string functions efficiently on C. You will do this by working on a real life problem: Spam Detection. You will use a widely used Natural Language Processing measure called “tf-idf” for this assignment. You are provided with a real word data set called “SMS Spam Collection” which includes real word SMS messages open to the public research. This data set also includes previous spam detection results for each message as “ham”(non spam) and spam. You will process this file and do spam detection based on calculated “tf-idf” metrics. You can learn more about tdf-idf in the References section.

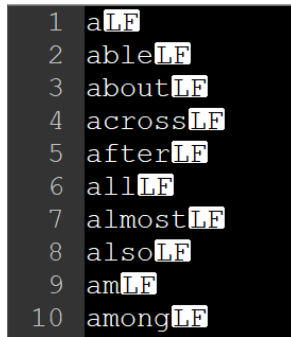
2 Datasets

“SMSSpamCollection.txt” has SMS messages and their spam status on each line. A line in this text file begins with ham/spam then there is a ‘\t’ character until the beginning of the message. Line ends with end line character ‘\n’.

```
20 spam→England v Macedonia - dont miss the goals/team news. Txt ur national team to 87077 eg ENGLAND to 87077 Try
21 ham→Is that seriously how you spell his name?1.5
22 ham→I'm going to try for 2 months ha ha only joking1.5
23 ham→So u pay first lar... Then when is da stock comin...1.5
24 ham→Aft i finish my lunch then i go str down lor. Ard 3 smth lor. U finish ur lunch already?1.5
25 ham→Ffffffffff. Alright no way I can meet up with you sooner?1.5
26 ham→Just forced myself to eat a slice. I'm really not hungry tho. This sucks. Mark is getting worried. He knows I'
27 ham→Lol your always so convincing.1.5
```

Figure 1: Sample SMS messages and their labels from SMSSpamCollection.txt

There is also a text file called “99webtools.txt” which includes common stop words in English language. This file includes all words as separate lines in the file, each line ends with end line character ‘\n’.



```
1 aLF
2 ableLF
3 aboutLF
4 acrossLF
5 afterLF
6 allLF
7 almostLF
8 alsoLF
9 amLF
10 amongLF
```

Figure 2: Sample words from 99webtools.txt

3 Logistics

This is an individual project. All handins are electronic. Clarifications and corrections will be announced on Blackboard.

4 Handout Instructions

4.1 How to start

- I Accept the GitHub Classroom assignment using the link: <https://classroom.github.com/a/06FSumvF>
- II Clone the GitHub repository created for you to a Linux machine in which you plan to do your work (We advice you to do your work on our linux servers [linuxpool.ku.edu.tr]. See Section 8 for details.) :

```
$ git clone https://github.com/COMP201-Fall21/assignment-2-USER.git
```


(Replace USER with your GitHub username that you use to accept the assignment)

4.2 Main Task

You will fill the *main.c* file with the given tasks. In the file, supposed locations of tasks are indicated as comments.

4.3 How to Read a File

1. First, open the text file using the *fopen()* function.
2. Then, use the function *fgets()* to read text from the stream and store it as a string. The newline or EOF character makes the *fgets()* function stop reading so you can check the newline or EOF file character to read the whole line. Note: *fgets()* also reads endlime characters.
3. When you are done reading all the contents of the file, close the text file using the *fclose()* function.

5 Tasks

5.1 Task 1: Frequency Analysis (40 pts)

In this task, you will enter a word as second argument (first is argument indicating task 1: “calculate-tf”) and calculate its tf score for both spam and ham messages. Then you will determine which is larger. tf score is calculated as:

$(\text{Number of times word occurs in } S \text{ messages}) / (\text{Total number of } S \text{ messages})$

When calculating for spam messages take S spam, for ham messages, take it ham. You will need to find the number of occurrences of the word in the each S message and sum them for both spam and ham messages (separately). You also need to count spam and ham messages separately. You can assume words in the message are separated using non alphabetical characters (including numbers). It is advised for you to use *isalpha()* function for this reason. **You also need to do a non-case sensitive search. It is advised that you convert both input word and words in the text file to the lowercase string.** Also don't forget to use floats for calculations, output resulting float values in 6 precision (use *%.6f*).

Sample Outputs:

```
$ ./main calculate-tf meet
Total Ham: 4827
Word in Ham: 75
Calculated tf value for this word: 0.015538
Total Spam: 747
Word in Spam: 8
Calculated tf value for this word: 0.010710
This word is mostly used for non spam messages
```

```
$ ./main calculate-tf Word
Total Ham: 4827
Word in Ham: 13
Calculated tf value for this word: 0.002693
Total Spam: 747
Word in Spam: 25
Calculated tf value for this word: 0.033467
This word is mostly used for spam messages
```

```
$ ./main calculate-tf fdwkdfawhd
This word doesn't occur in the text!
```

5.2 Task 2: Spam Detection (55 pts)

This time 5 words will be entered instead of 1 word. Each word is a separate console argument. You will calculate tf-idf score for each of the words (for both spam and ham), Then you will sum them up to find total tf-idf score. Finally, you will determine if the sentence is spam or not spam according to your results (which is larger spam score or ham score?). For each word tdf-idf score is calculated as $tf * idf$, tf calculation is the same as task 1. To calculate idf you will use this formula:

$\ln((\text{total number messages}) / \text{total number of messages having at least 1 occurrence of given word})$.

As seen in the formula, idf scores of spam and ham messages will be the same (it is a measure of general text not spam/ham messages specifically). So, only the tf score changes between spam and ham, as seen in the task 1. To use \ln function, *math.c* is included in *main.c*, you can use *logf()* function for more accurate floating type calculations. You need to handle the cases which denominator inside of \ln becomes 0. Ex: $\ln(0/0)$, in such cases *logf()* function outputs NaN, breaking all of the calculations. To solve this you have to take $\text{idf}=0$. **All restrictions from Task 1 also apply here.**

Sample Outputs:

```
$ ./main predict urgent help wanted contact us
Total tf-idf score from non spam messages for the sentence: 0.134213
Total tf-idf score from spam messages for the sentence: 0.920415
This sentence is spam.
```

```
$ ./main predict heLlo what are U doin
Total tf-idf score from non spam messages for the sentence: 0.857701
Total tf-idf score from spam messages for the sentence: 0.837070
This sentence is not spam.
```

Note: To examine the case when both of the scores are 0 refer to the Task Extra. (Only the final output text.)

5.3 Task Extra: Stop Words (10 pts)

For this task you can use your Task 2 code. But you have to modify it to consider stop words in calculations. For each word in the sentence, compare it with stop words in the “99webtools.txt” file. If it exists in the file, do not use it in $\text{tdf} \times \text{idf}$ calculations. In other words, consider that word’s tf-idf score 0. If a stop word is found, you also have to alert it in the console. **All restrictions from Task 1 and Task 2 also apply here.**

Sample Outputs:

```
$ ./main predict-wo-stopwords heLlo what are U doin
what is a stop word, it is not used in tf-idf calculations.
are is a stop word, it is not used in tf-idf calculations.
Total tf-idf score from non spam messages for the sentence: 0.470386
Total tf-idf score from spam messages for the sentence: 0.494196
This sentence is spam.
```

```
$ ./main predict-wo-stopwords does either else ever has
does is a stop word, it is not used in tf-idf calculations.
either is a stop word, it is not used in tf-idf calculations.
else is a stop word, it is not used in tf-idf calculations.
ever is a stop word, it is not used in tf-idf calculations.
has is a stop word, it is not used in tf-idf calculations.
Total tf-idf score from non spam messages for the sentence: 0.000000
Total tf-idf score from spam messages for the sentence: 0.000000
Tf-idf scores are found to be 0, spam detection failed!
```

6 Evaluation

Your score will be computed out of a maximum of 100(+10 from extra) points based on the following distribution:

35 Task 1.

55 Task 2.

5 Style points.

5 Effective use of version control points.

10 Extra task.

Task Points. Your exact outputs will be matched to some selected test cases (all are different from samples). Therefore, to ease auto grading try to match the outputs to the outputs in the samples. You will get points separately from each of the test cases. So, you can get partial points. There is also an optional extra task to get some bonus points. To compile *main.c* use the given Makefile. Run your code using “calculate-tf”, “predict”, “predict-wo-stopwords” arguments for tasks 1,2,3 respectively, just like in the samples.

Effective use of version control points. You are required to push your changes to the repository frequently. If you only push the final version, even if it is implemented 100% correctly, you will lose a fraction of the grade because you are expected to learn to use Version Control Systems effectively. You do not have to push every small piece of change to Github but every meaningful change should be pushed. For example, each of the implemented features can be one commit (like counting the occurrences of the word in the file). But try to keep commit messages meaningful.

Style points. Finally, we’ve reserved 5 points for a subjective evaluation of the style of your solutions and your commenting. Your solutions should be as clean and straightforward as possible. Your comments should be informative, but they need not be extensive.

Important Note: We use automated plagiarism detection to compare your assignment submission with others and also the code repositories on GitHub and similar sites. Moreover, we plan to ask randomly selected 10% of students to explain their code verbally after the assignments are graded. And one may lose full credit if he or she fails from this oral part.

7 Handin Instructions

As with Assignment 1, we use GitHub for the submissions as follows. Note that we want you to get used to using a version management system (Git) in terms of writing good commit messages and frequently committing your work so that you can get most out of Git.

I Commit all the changes you make: `$ git commit -a -m "commit message"`

Note: please use meaningful commit messages because

II Push your work to GitHub servers: `$ git push origin main`

8 How to use linuxpool.ku.edu.tr linux servers ¹

I Connect to KU VPN (If you are connected to the KU network, you can skip this step.)

See for details: <https://confluence.ku.edu.tr/kuhelp/ithelp/it-services/network-and-wireless/vpn-access>

II Connect to linuxpool.ku.edu.tr server using SSH (Replace USER with your Koç University username):

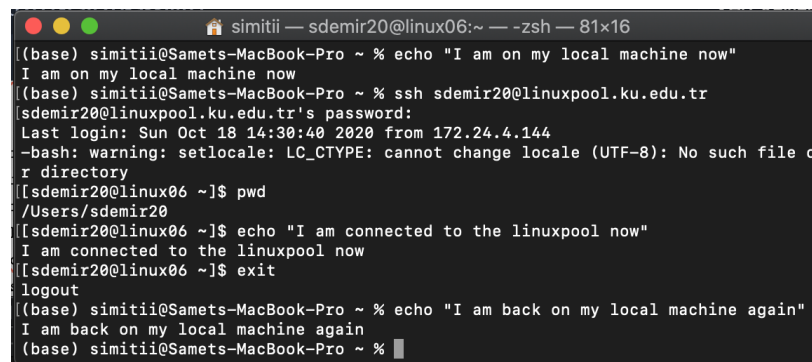
`$ ssh USER@linuxpool.ku.edu.tr`

(It will ask your password, type your Koç University password.)

III When you are finished with your work, you can disconnect by typing: `$ exit`

Your connection to the server may drop sometimes. In that case, you need to reconnect.

We advice you to watch the following video about the usage of SSH, which is used to connect remote servers, and SCP, which is used to transfer files between remote servers and your local machine: <https://www.youtube.com/watch?v=rm6pewTcSro>



```
simitii — sdemir20@linux06:~ — zsh — 81x16
(base) simitii@Samets-MacBook-Pro ~ % echo "I am on my local machine now"
I am on my local machine now
(base) simitii@Samets-MacBook-Pro ~ % ssh sdemir20@linuxpool.ku.edu.tr
sdemir20@linuxpool.ku.edu.tr's password:
Last login: Sun Oct 18 14:30:40 2020 from 172.24.4.144
-bash: warning: setlocale: LC_CTYPE: cannot change locale (UTF-8): No such file o
r directory
[sdemir20@linux06 ~]$ pwd
/Users/sdemir20
[sdemir20@linux06 ~]$ echo "I am connected to the linuxpool now"
I am connected to the linuxpool now
[sdemir20@linux06 ~]$ exit
logout
(base) simitii@Samets-MacBook-Pro ~ % echo "I am back on my local machine again"
I am back on my local machine again
(base) simitii@Samets-MacBook-Pro ~ %
```

Figure 3: How to connect and disconnect using SSH

9 Academic Integrity

All work on assignments must be done individually unless stated otherwise. You are encouraged to discuss with your classmates about the given assignments, but these discussions should be carried out in an abstract way. That is, discussions related to a particular solution to a specific problem (either in actual code or in the pseudocode) will not be tolerated. In short, turning in someone else's work, in whole or in part, as your own will be considered as a violation of academic integrity. Please note that the former condition also holds for the material found on the web as everything on the web has been written by someone else. See [Koç University - Student Code of Conduct](#).

¹For details, please see the guide on linuxpool that we have announced on Blackboard.

10 Late Submission Policy

You may use up to 7 grace days (in total) over the course of the semester for the assignments. That is you can submit your solutions without any penalty if you have free grace days left. Any additional unapproved late submission will be punished (1 day late: 20% off, 2 days late: 40% off) and **no submission after 2 days will be accepted.**

11 References

SMS Messages Data Set: [Link](#)

Stop Words Data Set: <https://github.com/igorbrigadir/stopwords>

tf-idf source: <http://www.tfidf.com/>

tf-idf source: <https://en.wikipedia.org/wiki/Tf%E2%80%93idf>

NLP source: [Link](#)