CSE 360: Introduction to Software Engineering

Team One

Text File Formatter and Analyser

User Guide

Daniel Aguiar, Marcus Elia, Craig Knoblauch, Lance Tokuno

Spring 2018

Table of Contents

Introduction

The Text File Formatter and Analyser (TFFA) is a robust software application that will help users with all their formatting needs. This document contains the instructions for using TFFA, including pictures of how to navigate the user interface. We will go over the many formatting options available in TFFA and how to use them on some sample text files as an example.

Overview

With TFFA, you simply select the text file that you want formatted, select the options for formatting your document, and specify the location and name of the resulting formatted file.. In addition, each time you format a text file, TFFA will display statistics, such as the number of words processed, the number of lines in the formatted document, the number of blank lines removed from the input file, and the average number of words per line as well as the average line length, of the formatted document.

There are some basic formatting operations that TFFA will perform on every text file it processes. The formatted output will have at most 80 characters on any given line. As such, it will try to maximise the number of words that appear on each line, without going over the 80 character limit of any line. For example, if one line is "Hello", and the line below it is "world!", the formatter will merge the words together into "Hello world!" since they together do not contain more than 80 characters. In addition, TFFA will always remove any excess spaces around words and reduce it to one space (e.g. "Hello   world!" turns into "Hello world!"). Finally, TFFA will always remove blank lines from the formatted document. The operation where you can choose different formatting options is the text justification: you can have your document left- or right-justified.

TFFA is simple enough that everything can be done on one page: selecting the file to format, the location and name of the resulting formatted version, the formatting options, and the statistics all appear on one page.

Getting Started

Getting started with the TFFA software only takes a few simple steps:

1. Install the Java runtime environment if it is not already installed:
   https://java.com/en/download/
2. After purchasing TFFA, clients will receive a Dropbox link where the software can be downloaded. Download TFFA.jar from Dropbox on any company licensed computer.
3. Locate the directory where TFFA.jar has been downloaded:
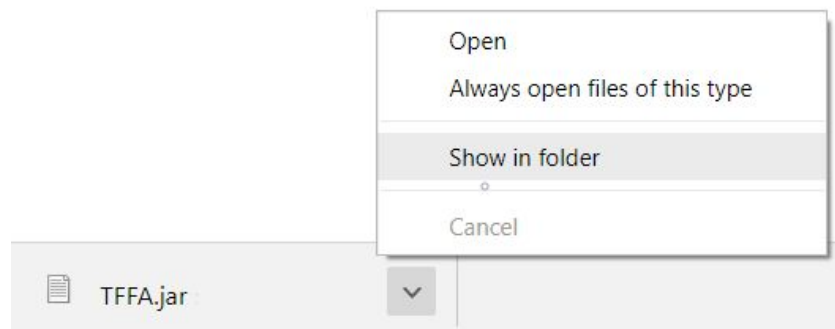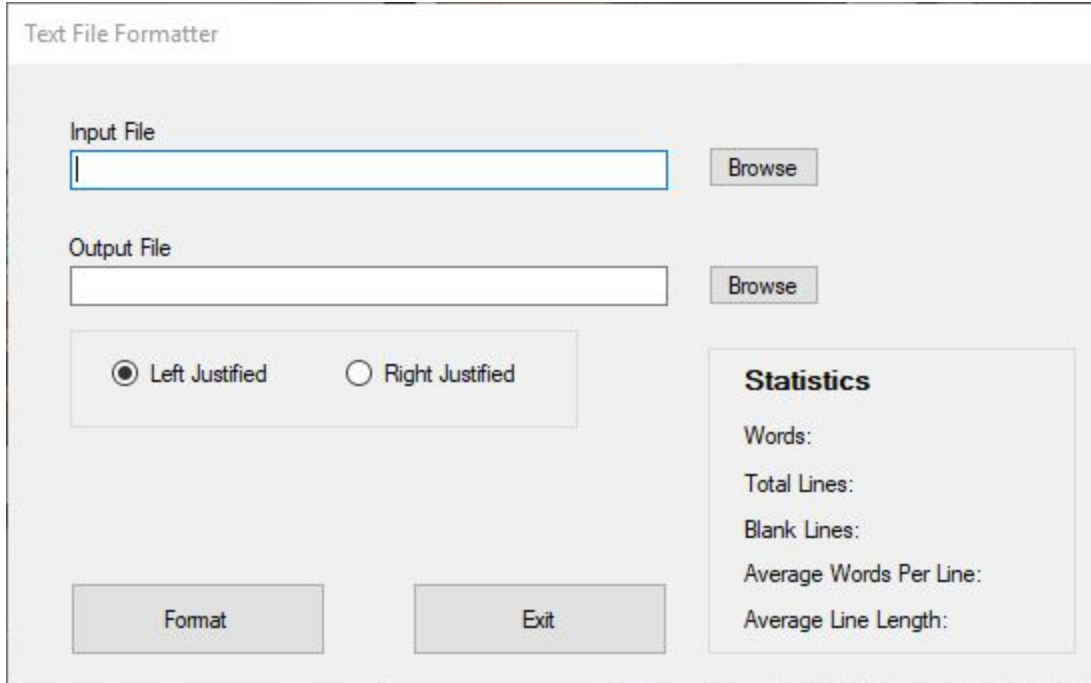

Locating the TFFA.jar file

After the download has completed, navigate to 'Show in folder' to easily find the correct directory. Move the TFFA.jar file to any desired location for easier access.

4. To run the program, either double click on the TFFA.jar file or use the command line:

   java -jar TFFA.jar

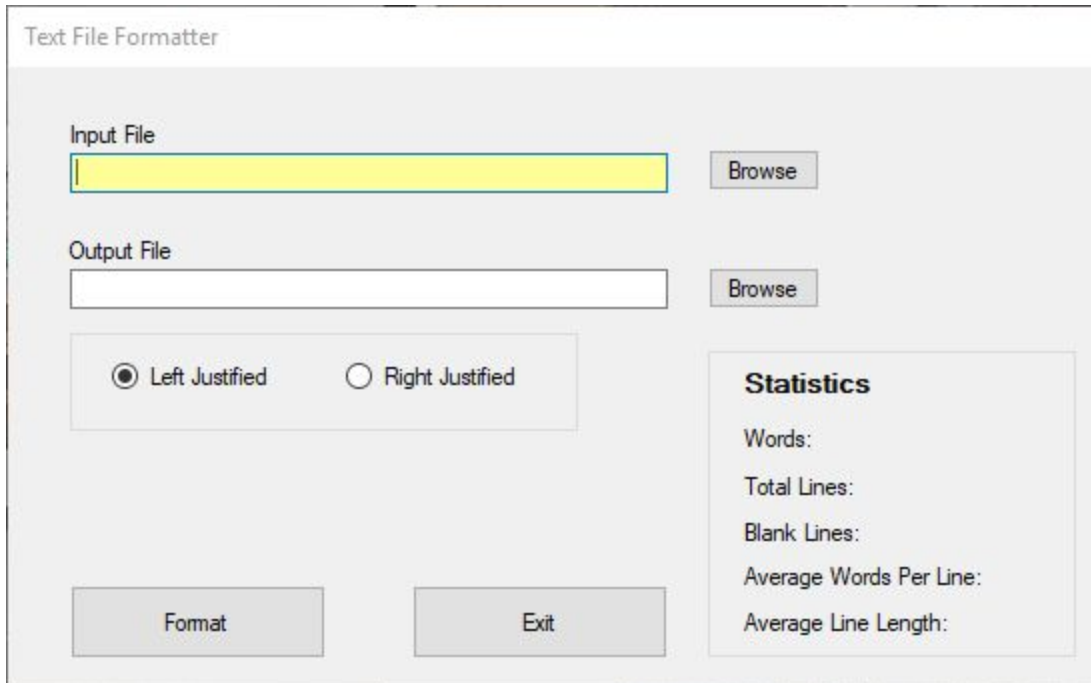# User Interface

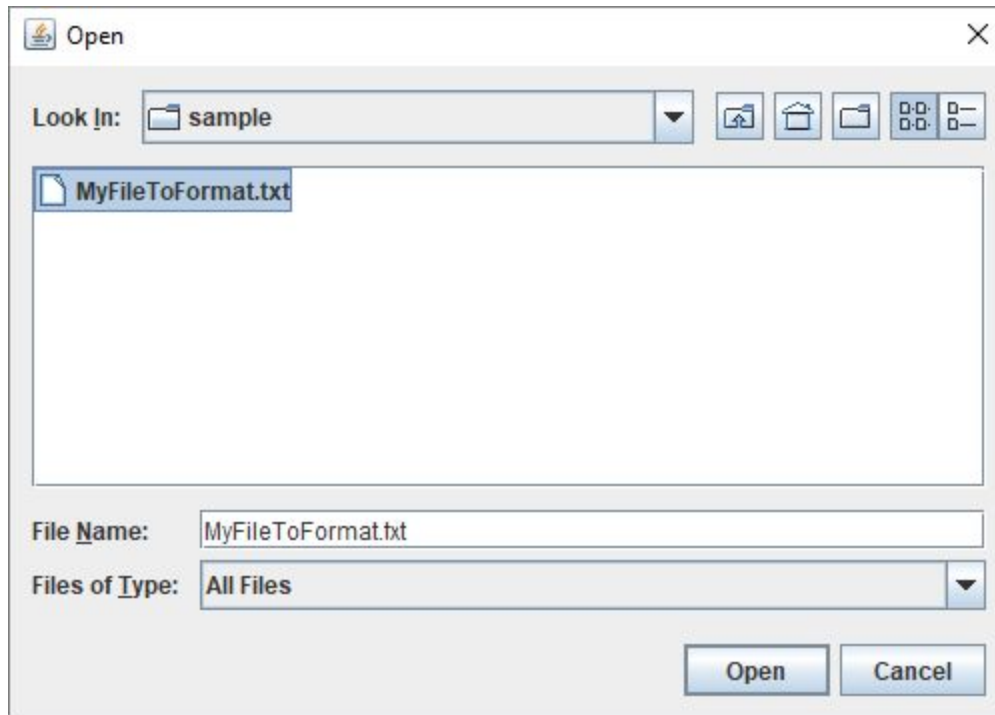Below, the starting page of TFFA is shown. You may find that operating this application is very simple.

To specify your input file, simply click 'Browse' next to the 'Input File' box (shown below).
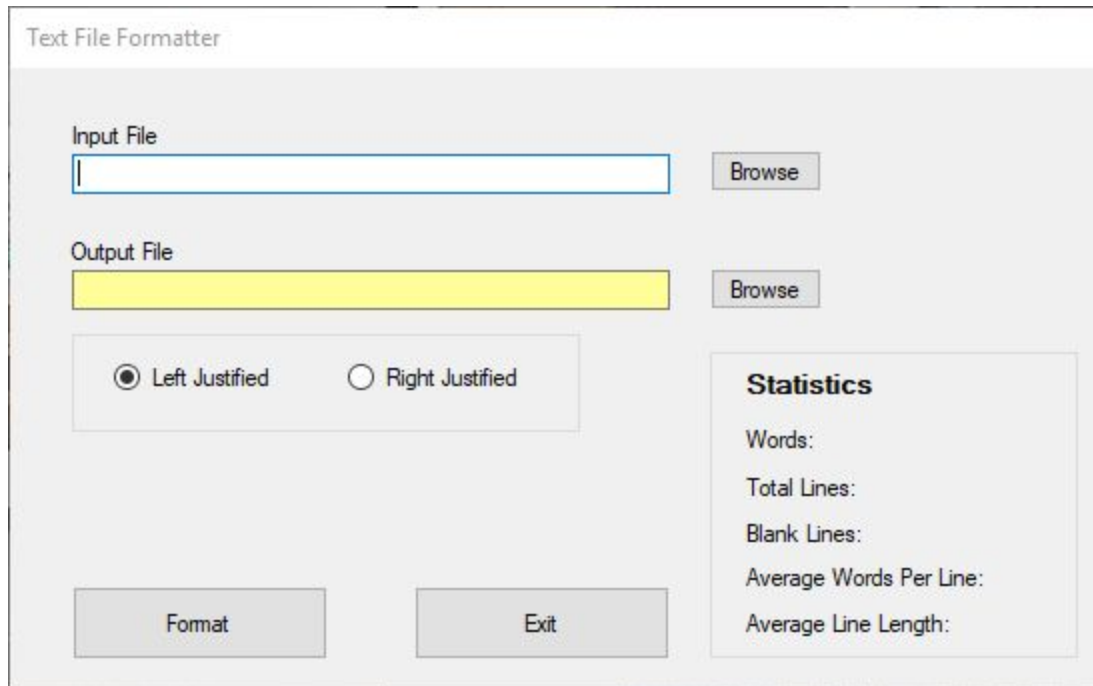
A file browser for your system, similar to the one shown below, will appear. Here you can select your input file, and approve the selection by pressing 'Open' in the browse window.

If you already know the location and name of your input file, you can enter it directly in the 'Input File' field.

You'll need to specify your output file as well. This is the same process as specifying an input file. The 'Output File' selection box is shown below.

If you incorrectly specify an input or output file, the error popups shown below will appear.
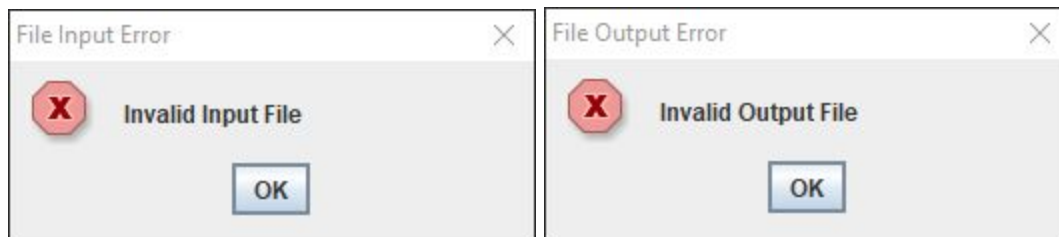
Before you format your file, make sure you choose your justifications from the highlighted section shown below.

To format your file, click the 'Format' button highlighted below.

Statistics about your output file will appear in the highlighted section shown below.

To exit TFFA, simply click the 'Exit' button. Again, this is highlighted below.

Input / Output Format

Input File: Must be a .txt file of any size with ASCII encoded text.

Output File: A text file containing the formatted input in ASCII encoded text.

- 80 characters per line.
    - If a single word larger than 80 characters, this limit will be exceeded so that the word is printed on a single line.
    - Words will be merged to the same line as long as they follow the rule of 80 characters per line. Words are read sequentially from top to bottom, left to right.
- Excess space after words are reduced to a single space.
- Blank lines are removed.
- Left or right justification.

Sample Run

```
Text File Formatter

  Input File
  ┌─────────────────────────────────────┐   ┌──────────┐
  │ c:\MyFiles\sample1.txt              │   │ Browse   │
  └─────────────────────────────────────┘   └──────────┘

  Output File
  ┌─────────────────────────────────────┐   ┌──────────┐
  │ c:\MyFiles\sample_out.txt           │   │ Browse   │
  └─────────────────────────────────────┘   └──────────┘

  ┌──────────────────────────────────────┐   Statistics
  │ ◉ Left Justified   ○ Right Justified │
  └──────────────────────────────────────┘   Words: 704

                                              Total Lines: 79

                                              Blank Lines: 6

                                              Average Words Per Line: 9.2
  ┌──────────────┐      ┌──────────────┐
  │    Format     │      │     Exit     │       Average Line Length: 79.4
  └──────────────┘      └──────────────┘
```

If the input file (c:\MyFiles\sample1.txt) has these lines in it:

the quick brown fox! jumps over the lazy dog

the quick brown fox jumps over the lazy dog!

the quick. brown fox! jumps   over the lazy dog.

the output file (c:\MyFiles\sample_out.txt) will contain these lines:

the quick brown fox! jumps over the lazy dog the quick brown fox jumps over the

lazy dog! the quick. brown fox! jumps over the lazy dog.

Appendix 1: Sample Input Files

Here are a couple sample input files. We'll see their respective formatted versions in Appendix 2. The first:

```
the quick brown  fox! jumps over the lazy dog
the quick brown  fox jumps over the lazy dog!
the quick. brown  fox! jumps    over the lazy dog.
theQuickBrownFoxJumpsOverTheLazyDog!TheQuickBrownFoxJumpsOverTheLazyDog!TheQuickBrownFoxJumpsOverTheLazyDog!
```

Here is he second input file:

```
Lorem ipsum dolor sit amet, consetetur sadipscing elitr
, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua.
     At vero eos et accusam et justo duo dolores et ea rebum.
```

Take note of the punctuation and the spaces, as well as the length of the last line in the first file.

Appendix 2: Sample Output Files

Now we will see how TFFA will process these two files. The following is the result of formatting the first input file with left justify.

```
the quick brown fox! jumps over the lazy dog the quick brown fox jumps over the
lazy dog! the quick. brown fox! jumps over the lazy dog.
theQuickBrownFoxJumpsOverTheLazyDog!TheQuickBrownFoxJumpsOverTheLazyDog!TheQuickBrownFoxJumpsOverTheLazyDog!
```

The formatter was able to squeeze in some words from line 2 into line 1, and some words from line 3 into line 2. However, due to line 4 containing one really long word (above 80 characters), no more words were added to line 3 of the original file from line 4, and line 4's length is above 80 characters. Let us now see how it looks right justified.

```
               the quick brown fox! jumps over the lazy dog the quick brown fox jumps over the
                                   lazy dog! the quick. brown fox! jumps over the lazy dog.
BrownFoxJumpsOverTheLazyDog!TheQuickBrownFoxJumpsOverTheLazyDog!TheQuickBrownFoxJumpsOverTheLazyDog!
```

The following file is the result of formatting the second input file above, left justified first.

```
Lorem ipsum dolor sit amet, consetetur sadipscing elitr , sed diamnonumy eirmod
tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At
vero eos et accusam et justo duo dolores et ea rebum.
```

Even punctuation here counts as words, so there is a space between "elitr" and "sed" in the first line. Now look at the right justified version.

```
Lorem ipsum dolor sit amet, consetetur sadipscing elitr , sed diamnonumy eirmod
   tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At
                     vero eos et accusam et justo duo dolores et ea rebum.
```

Appendix 3: Glossary of Terms

- **Blank Lines**: Lines that consist of a only a newline character or whitespace.
- **Left Justified**: The formatting of a document so that the text is aligned with the left margin.
- **Right Justified**: The formatting of the document so that the text is aligned with the right margin.
- **Statistics:** Attributes of the output file e.g average line length.
- **TFFA**: Text File Formatter and Analyser, the name of the application.
- **Word**: Any sequence of non-space characters.