**CSGE601020**

**Dasar-Dasar Pemrograman 1**

**Foundations of Programming 1**

# Programming Assignment 2: Word Clouds

LAST DAY for uploading the result of your work to SCeLE: Thursday 5 Oct 2017 (11:55 PM).

Don't forget to write enough comments in your source code.

Please contact your TA (teaching assistant) for giving a demo of your work as soon as possible. The TA will give you a mark after the demo.

Please start working on this assignment immediately. If you have any questions, please ask the TA or the professor.

**Marking scheme:**

60 % correctness
30 % explanation in demo session
10 % program documentation (comments, neatness)

## Task Description

### Assignment Overview

This assignment will give you useful experience with functions, strings, lists, text processing, as well as file I/O.

### The Problem

A common element seen on web pages these days are word clouds or tag clouds of a document. A word cloud is a visual representation of frequency of words, where more frequent words are represented in larger font.

You are going to analyze a text document given by the user and create a word cloud for it, where the size of the font in the cloud indicates the frequency of the words.

### Background

You are provided with a number of elements to help with this assignment:

- text documents for processing: banKimoonSpeech.txt, JokoWidodoSpeechAPEC2014.txt, Hello_by_Adele.txt
- a file containing stop words.
- some functions for creating an HTML (hypertext markup language) file which can be

displayed by a standard browser.

Stop Words
Not all words are worth counting. ‘a’, ‘the’, ‘was’, ‘and’, etc. are just junk. A list of such words is provided in the file **’stopWords.txt’.** Each line has a single word. No word in the stop word list should be counted in the word cloud.

Functions
Three functions and an example are provided in **htmlFunctions.py**. Use the functions in your program. That file contains:

- `make_HTML_word(word, count, high, low)`: This function takes a `word` and wraps it in a font tag with a specific size and a random color — returning that string whose `fontsize` is between `htmlBig` and `htmlLittle` (two local variables in the function that you can change to be whatever you like). The parameters are:
    - `word` (string), the word to be wrapped,
    - `count`, how many times it occurred in the input document,
    - `high`, the highest word count in the input document,
    - `low`, the lowest word count in the input document.

  The function returns the word as a HTML-formatted string.

- `make_HTML_box(body)`. This function takes a single string named `body` that contains all the font-wrapped words from `make_HTML_word` and places them in an HTML box to be displayed. It returns a string which is the HTML code for the box.
- `print_HTML_file(body,title)`. Takes the `body` (string) returned from `make_HTML_box`, wraps a standard HTML web page tags around it, and creates a file. The string `title` is used in the HTML. The title is also used as the file name with an ‘.html’ suffix. The file is created and written to secondary storage.

**Program Specification**

- Prompt the user for the text file to be processed.
- Print the top 50 counts (word and count pairs). Sort the words by count (frequency) for printing. (For the next step you will sort the words alphabetically for the HTML file.)
- Generate an HTML file (with suffix .html) using the provided functions to generate a word-cloud file with the top 50 words. For the word cloud sort the top 50 words alphabetically—the word cloud looks more interesting that way. You can view the files in your favorite browser.

**Assignment Hints:**
There are a couple of problems here. Think about each one before you start to program.

1. You have to read in the file and separate the contents into words.

2. Once you have the words separated, you must remove all stop words (using the provided file “stopWords.txt”). Also, remember to remove punctuation from words, just because a word comes at the end of a sentence and has a period at the end of it doesn’t make it a different word. (Importing `string` and using `string.punctuation` is a useful way to specify

punctuation.)

3. You must then count the frequency of each word collected. Use the list data structure. You are not allowed to use the dictionary data structure.

4. Once you have a complete list, sort the list. Put count first in the tuple because sorting (using either `sort` or `sorted`) will sort on the first item.

5. Use the provided functions to turn the words and counts into an HTML page (see the example in the file **htmlFunctions.py** ).

**Happy Programming! 'Met ngoding!**

**L. Y. Stefanus**

## Sample Ouput:

1. First is what is printed to the IDLE shell (note: sorted by count).

```
Program to create word cloud from a text file
The result is stored as an HTML file

Please enter the file name: banKimoonSpeech.txt

banKimoonSpeech.txt :
50 words in frequency order as (count:word) pairs

 13:weapons        10:people         9:year           9:international
  8:human           8:development     7:world          7:rights
  7:peace           7:chemical       6:end            5:syria
  5:nations         5:action         4:women          4:united
  4:states          4:future         4:climate        3:violations
  3:time            3:system         3:syrian         3:sustainable
  3:support         3:strong         3:set            3:security
  3:region          3:poverty        3:peoples        3:parties
  3:member          3:mdgs           3:leadership     3:leaders
  3:law             3:humanitarian   3:government     3:goals
  3:global          3:face           3:dignity        3:deadly
  3:community       3:commitment     3:challenge      3:century
  3:build           3:arms

Please type Enter to continue ...
```

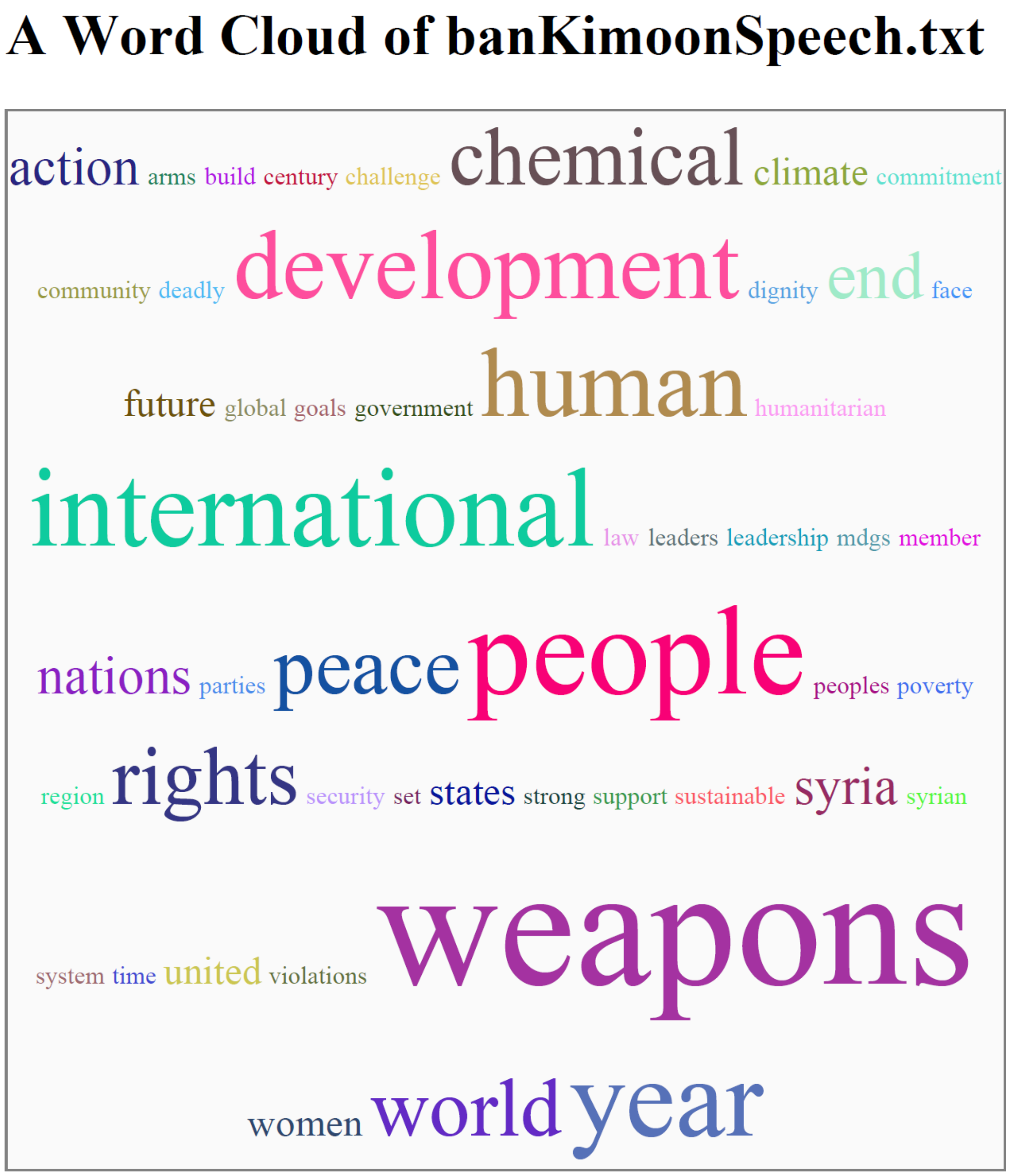
2. Next is the generated HTML file: banKimoonSpeech.txt.html.

```
    <html> <head>
    <title>A Word Cloud of banKimoonSpeech.txt</title>
    </head>

    <body>
    <h1>A Word Cloud of banKimoonSpeech.txt</h1>
<div style="
    width: 560px;
    background-color: rgb(250,250,250);
    border: 1px grey solid;
    text-align: center" ><span style="color: rgb(209, 148, 5); font-
size:30px;">action</span> <span style="color: rgb(9, 107, 6); font-
size:14px;">arms</span> <span style="color: rgb(71, 151, 221); font-
size:14px;">build</span> <span style="color: rgb(13, 102, 176); font-
size:14px;">century</span> <span style="color: rgb(77, 82, 229); font-
size:14px;">challenge</span> <span style="color: rgb(41, 174, 122); font-
size:46px;">chemical</span> <span style="color: rgb(23, 140, 5); font-
size:22px;">climate</span> <span style="color: rgb(85, 112, 28); font-
size:14px;">commitment</span> <span style="color: rgb(178, 11, 119); font-
size:14px;">community</span> <span style="color: rgb(77, 197, 113); font-
size:14px;">deadly</span> <span style="color: rgb(239, 75, 238); font-
size:55px;">development</span> <span style="color: rgb(154, 129, 120); font-
size:14px;">dignity</span> <span style="color: rgb(164, 135, 223); font-
size:38px;">end</span> <span style="color: rgb(151, 33, 46); font-
size:14px;">face</span> <span style="color: rgb(8, 146, 190); font-
size:22px;">future</span> <span style="color: rgb(72, 88, 98); font-
size:14px;">global</span> <span style="color: rgb(62, 40, 248); font-
size:14px;">goals</span> <span style="color: rgb(216, 44, 48); font-
size:14px;">government</span> <span style="color: rgb(111, 210, 157); font-
size:55px;">human</span> <span style="color: rgb(232, 125, 127); font-
size:14px;">humanitarian</span> <span style="color: rgb(34, 77, 178); font-
size:63px;">international</span> <span style="color: rgb(0, 82, 51); font-
size:14px;">law</span> <span style="color: rgb(186, 21, 125); font-
size:14px;">leaders</span> <span style="color: rgb(55, 69, 49); font-
size:14px;">leadership</span> <span style="color: rgb(79, 110, 207); font-
size:14px;">mdgs</span> <span style="color: rgb(86, 81, 32); font-
size:14px;">member</span> <span style="color: rgb(194, 202, 181); font-
size:30px;">nations</span> <span style="color: rgb(227, 85, 132); font-
size:14px;">parties</span> <span style="color: rgb(15, 177, 209); font-
size:46px;">peace</span> <span style="color: rgb(72, 140, 206); font-
size:71px;">people</span> <span style="color: rgb(110, 122, 1); font-
size:14px;">peoples</span> <span style="color: rgb(82, 65, 26); font-
size:14px;">poverty</span> <span style="color: rgb(181, 141, 182); font-
size:14px;">region</span> <span style="color: rgb(241, 192, 247); font-
size:46px;">rights</span> <span style="color: rgb(104, 197, 146); font-
size:14px;">security</span> <span style="color: rgb(90, 122, 80); font-
size:14px;">set</span> <span style="color: rgb(105, 86, 174); font-
size:22px;">states</span> <span style="color: rgb(231, 224, 167); font-
size:14px;">strong</span> <span style="color: rgb(117, 155, 28); font-
size:14px;">support</span> <span style="color: rgb(12, 109, 180); font-
size:14px;">sustainable</span> <span style="color: rgb(125, 184, 185); font-
size:30px;">syria</span> <span style="color: rgb(98, 153, 85); font-
size:14px;">syrian</span> <span style="color: rgb(47, 206, 227); font-
size:14px;">system</span> <span style="color: rgb(198, 1, 64); font-
size:14px;">time</span> <span style="color: rgb(35, 146, 211); font-
size:22px;">united</span> <span style="color: rgb(11, 119, 70); font-
size:14px;">violations</span> <span style="color: rgb(130, 111, 191); font-
size:96px;">weapons</span> <span style="color: rgb(108, 39, 193); font-
size:22px;">women</span> <span style="color: rgb(86, 1, 63); font-
size:46px;">world</span> <span style="color: rgb(239, 77, 135); font-
size:63px;">year</span></div>

    </body> </html>
```

Here is what the banKimoonSpeech.txt.html file looks like in a browser (note: the words are sorted alphabetically).



# A Word Cloud of banKimoonSpeech.txt

action arms build century challenge chemical climate commitment
community deadly development dignity end face
future global goals government human humanitarian
international law leaders leadership mdgs member
nations parties peace people peoples poverty
region rights security set states strong support sustainable syria syrian
system time united violations weapons
women world year