Available online at www.sciencedirect.com

ScienceDirect

Procedia Engineering 174 (2017) 788 – 792

Procedia Engineering

www.elsevier.com/locate/procedia

13th Global Congress on Manufacturing and Management, GCMM 2016

Development of Word Cloud Generator Software Based on Python

Yuping Jin*

Associate Professor, Department of Mathematics, Mudanjiang Normal University, Mudanjiang 157001, China

Abstract

A word cloud is a kind of weighted list to visualize language or text data, which gains increasing attention and more application opportunities as the big data time approaches. Currently, there has been some online word cloud generators available for users with simple requests, such as repeating the exact phrase, or collecting the text data from a web page. Moreover, most current word cloud generators cannot support characters other than English, which are limited in English-speaking users. There are also packages for programming languages (such as Python and R) to generate word clouds, which requires coding and is not user-friendly. This paper is focusing on developing a graphical user interface (GUI) software to generate word cloud maps with easy operations. The Python programming language is involved and some details are discussed. Finally the software is released for further application.

Peer-review under responsibility of the organizing committee of the 13th Global Congress on Manufacturing and Management

Keywords: Word cloud; GUI software; Python

1. Introduction

A word cloud is a kind of weighted list to visualize language or text data. It is an important branch of data mining, which has gained increasing attention and more application opportunities in the Big Data time [1, 2]. In early times, the prototype if word cloud is usually used as graphical maps to show the relative size of territories (such as cities, towns, regions) in terms of relative typeface size. Word cloud come from nowhere but it can be found in an early printed version of weighted English keywords in Douglas Coupland's Microserfs [3].

* Corresponding author. Tel.: +86-453-6516169; fax: +86-453-6516169.
E-mail address: jyjin@mdjnu.cn

Peer-review under responsibility of the organizing committee of the 13th Global Congress on Manufacturing and Management
doi:10.1016/j.proeng.2017.01.223

In the 21th century, as the development of internet technology, especially websites and blogs are commonly used, word cloud was used to put tags on the internet pages as navigation aids for readers' information through visualizing the frequency of each keyword by the font size.

There are 3 major word cloud maps applied in social networks distinguished by their algorithm instead of appearance,

- **Frequency** - In the frequency type, the size of font represents the number of keywords that appears in the collection. The frequency type is the most basic type used in mining text data.
- **Categorization** – In the categorization type, the size of font indicates the number of subcategories of a collection. The categorization type is commonly used in geographical mappings. However, the categorization type can be transmitted to the frequency type with regular coding.
- **Mixed** - In the mixed type, the data contains both frequency and categorization, which requires logically analyzing the complicated data before arranging the word cloud maps.

As with the tools to generate a word cloud map, the development is introduced as follows. Early times when artists drew graphical words has passed. Currently, there are some online word map generators such as "Wordclouds.com" [4] and "jasondavies.com" [5]. Using these generators, the users can upload a file or page, adjusting the word list, fonts, masks, colors or even scale and orientations. As word placement can be quite slow for more than a few hundred words, the layout algorithm can be run asynchronously, with a configurable time step size. This makes it possible to animate words as they are placed without stuttering.

Beyond the online tools, packages in programming languages provide much more flexible developing tools for the coders to generate a satisfaction word cloud according to their demands. On the Java, R and Python platform, the packages for word cloud are available. In this paper, the authors choose python as an illustration, a graphical user interface (GUI) was developed, then the program was packaged into an executive file for application.

2. Layout Algorithm

The basic layout algorithm in generating a word cloud is to estimate the frequency level of a keyword by their frequencies or categories. In principle, the font size of a word in the word cloud is determined by its appearing frequency. For a word cloud of categories number assigned to the exact category. For smaller frequencies, one can directly set the font sizes, from s_0 to whatever the maximum font size. For larger values, a scaling should be made. In a linear normalization, the weight t_i of a descriptor is mapped to a size scale of 1 through f, where $t_{\min}$ and $t_{\max}$ are specifying the range of available weights.

$$s_i = \begin{cases} \left\lceil \dfrac{f_{\max}(t_i - t_{\min})}{(t_{\max} - t_{\min})} \right\rceil & t_i > t_{\min} \\ 1 & \text{otherwise} \end{cases} \tag{1}$$

Since the number of indexed items per descriptor is usually distributed according to a power law ([6]), for larger ranges of values, a logarithmic representation makes sense [7].

The layout algorithm itself is incredibly simple. For each word, starting with

- Attempt to place the word at some starting point: usually near the middle, or somewhere on a central horizontal line.
- If the word intersects with any previously-placed words, move it one step along an increasing spiral. Repeat until no intersections are found.

The hard part is making to enhance the performing efficiency. According to Jonathan Feinberg, Wordle uses a combination of hierarchical bounding boxes and quad trees to achieve reasonable speeds [8].

Nomenclature

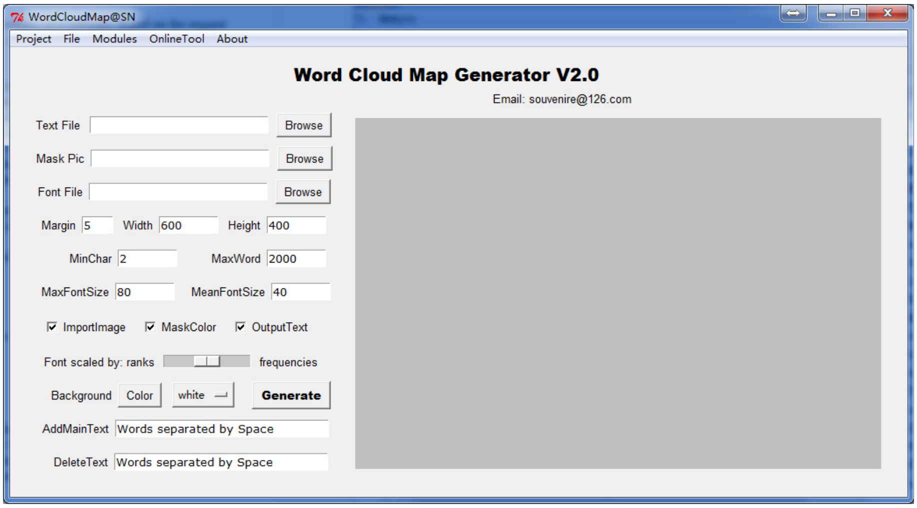
s_i	display font size of item *i*
$f_{\max}$	maximum font size
t_i	count of item *i*
$t_{\max}$	maximum of count
$t_{\min}$	minimum of count

Based on the layout algorithm mentioned above, the “wordcloud master” package [9] on Python platform was selected. For Chinese characters “jieba” (Chinese for “to stutter”) package [10] is selected to deal with Chinese words. The basic thought is to split Chinese words using a dictionary which can be edited or downloaded from the package website.

3. Graphical user interface (GUI) designation

The graphical user interface is a thought that satisfy the user friendly demand of a software. Based on the request of users, algorithm parameters, setting data saving and acquiring modes, a GUI can be determined. In this project, the following factors are considered.

- File loading: loading text data; mask picture and fonts (usually loaded from the system folder).
- Parameter setting: margin of text; width and height of picture; minimum character (easy excluding short word like “a”, “an”, “the” or function words such as “of”, “on”); maximum words considered; checking for using image, mask color and output the split text for further editing; also a parameter to determine whether the drawing uses ranks or frequencies: using ranks, one can get the font scale exact by the ranking of data, while using frequencies, the margin can be somehow adjusted by the frequency orders; background color can be selected or set; while one wanted to add some text words in the major place, or delete some unwanted words, the adjustments can be made in the software interface.
- Plot area: to demonstrate the loaded mask or the output figure is requested.
- Auxiliary functions: such as help; contract; and demo should be placed in the menu area.



Word Cloud Map Generator V2.0

Email: souvenire@126.com

Text File

Mask Pic

Font File

Margin 5 Width 600 Height 400

MinChar 2 MaxWord 2000

MaxFontSize 80 MeanFontSize 40

☑ ImportImage ☑ MaskColor ☑ OutputText

Font scaled by: ranks frequencies

Background white

AddMainText Words separated by Space

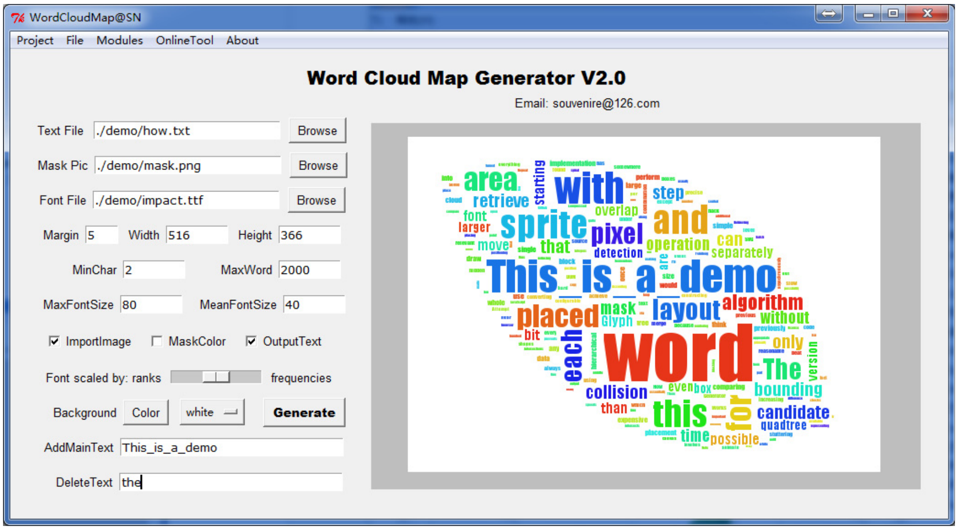
DeleteText Words separated by Space

Fig. 1. Graphical interface of the software.

Based the request analysis above, the software interface are designed as Fig. 1. To realize the functions, the classical “Tkinter” package was adopted to build the dialog interface. For the Input/Output of data flows, sub-packages such as “tkFileDialog” and “tkMessageBox” are adopted.

4. Illustrations

Using the online paragraph in [5] as an illustration. The software is applied using a leaf-like mask and “impact” font, a graphical picture can be obtained. The settings and the demo are shown in Fig. 2



Word Cloud Map Generator V2.0

Email: souvenire@126.com

Text File ./demo/how.txt

Mask Pic ./demo/mask.png

Font File ./demo/impact.ttf

Margin 5 Width 516 Height 366

MinChar 2 MaxWord 2000

MaxFontSize 80 MeanFontSize 40

☑ ImportImage ☐ MaskColor ☑ OutputText

Font scaled by: ranks frequencies

Background Color white

AddMainText This_is_a_demo

DeleteText the

Fig. 2. Using the software to generate the demo.

In order to compare with other tools, one can generate the word cloud using the tools mentioned in [4] and [5], the illustration is shown in Fig. 3.

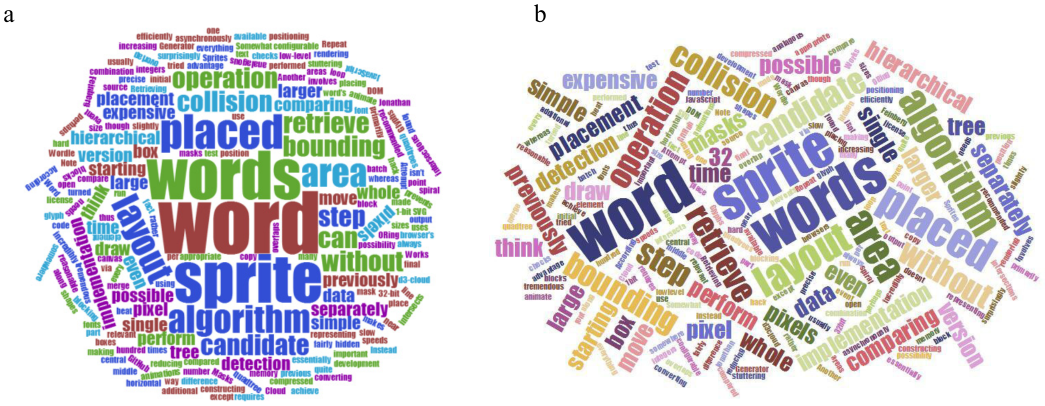


Fig. 3. Using the online tools to generate the demo (a) with online tool [4]; (b) with online tool [5].

It can be inferred from the figure that, the developed software can flexibly dealing with the text formats, adding or deleting the major key texts. The following is another demo to generate a text with Latin proverbs where Chinese characters and Latin word are mixed. The result is shown in Fig. 4, which indicated that, the software is compatible with mixing languages.

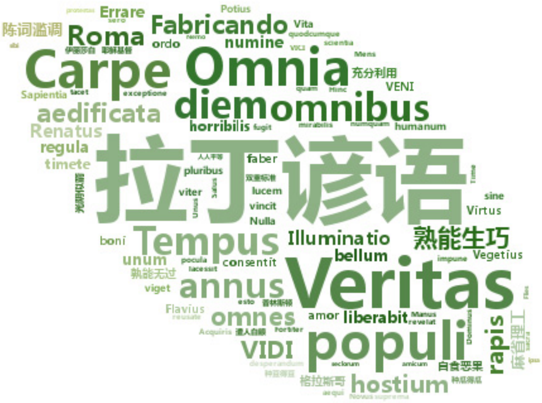


Fig. 4. Using the software to generate a word clod with mixed languages.

5. Concluding remarks

This paper presents the development of a word cloud map generator software, which can be used in mining big text data from the website, article or artworks. The software was based on Python, the algorithm is based on basic linear, power and logarithmic representation of font sizes, providing flexible, adjustable and user-friendly tool for users with text mining tasks. The software is proved to support mixed multiple language environment, which provides wider ranges of application.

Acknowledgements

This work was supported by the project of A Study on Branch Predictor of Multi-core Processor Based on Data-reusing (Grant No.1351MSYYB010) and JAVA project training course to the application of transformation and development (Grant No 16-JG18037)

References

[1] P. Kinnaird, I. Talgam-Cohen. "Big Data". XRDS: Crossroads, The ACM Magazine for Students. Association for Computing Machinery, 19(2012).

[2] V. Mayer-Schönberger; Kenneth Cukier (2013). Big Data: A Revolution that Will Transform how We Live, Work, and Think. Houghton Mifflin Harcourt, 2013.

[3] D. Coupland. Microserfs, in: USA hardback., New York, 1995.

[4] http://www.wordclouds.com/ . Retrieved October 2016

[5] https://www.jasondavies.com/wordcloud/about/ . Retrieved October, 2016.

[6] Jakob Voss: Collaborative thesaurus tagging the Wikipedia way. April 2006.

[7] Kentbyte: "Tag Cloud Font Distribution Algorithm". June 2005". Echochamberproject.com. Retrieved 2013-07-27.

[8] Steffen Lohmann, Jürgen Ziegler, and Lena Tetzlaff (2009). Comparison of Tag Cloud Layouts: Task-Related Performance and Visual Exploration. T. Gross et al. (Eds.): INTERACT 2009, Part I, LNCS 5726, (2009) 392–404.

[9] https://amueller.github.io/word_cloud/ . Retrieved October , 2016.

[10] https://pypi.python.org/pypi/jieba/ . Retrieved October, 2016.