ORIGINAL RESEARCH PAPER

Data Science

KEY WORDS: Text visualization, Summarization, Text mining, Word-cloud, Open-Source Visualization tools, Term Weighting, Feature Weighting

USING OPEN-SOURCE TOOLS FOR TEXT VISUALIZATION EFFECTIVELY

Gowri R Choudhary* Computer science Department, Career Point University*Corresponding Author

Dr. Iti Sharma Faculty of Computer Science, Government Polytechnic college

ABSTRACT

Researchers from many fields produce data in text form and require its visualization for further analysis or application. Availability of open-source tools has made it convenient for them, yet there is a challenge in choosing a suitable tool and preparing a compatible input. This paper describes 10 popular open-source text visualization tools for word-clouds, compares them and suggests a priority window technique to convert corpus into a small single text file that retains the data characteristics of corpus. The visualizations produced by various methods to convert corpus into a single text file are compared to show the effectiveness of our proposal.

INTRODUCTION

Text is the most comprehensible form for human beings. Humans developed the script for communication because people communicate majorly in the text form not in the numerical form. Thus, text being the major part of human communication, comprehension and interaction, a copious amount of data is available in textual form like emails, webpages, newsfeeds, articles, stories etc. All these texts have different patterns inside it and when a professional derives those patterns through automated process, it is termed as text analytics [1]. Actually, text analytics enabling the user to convert the text into information so that the major five tasks [2] can be achieved through text analytics and also benefit to society, science and business are as follows: (i) Information extraction [3], (ii) Information retrieval, (iii) Clustering / Categorization, and (iv) Summarization. Summarization [4, 5] is a task of condensing large amount of text data into its most important parts such that information loss is minimal. The major summarization approaches are categorized into two forms [6]: First is Text-to- Text form [7, 8] is also called Text summarization which shortening up the long text by recognizing the important point and summarized into a smaller text without altering the meaning of the text and important approaches are deeply described in [9, 10]. The second is Text-to-Graphical form [11] is extracting the important words from a large amount of text and summarizing into a graphical form. This graphical form is termed as Text visualization. Text visualization is a collective term for methods used to convert results of text analysis in a visual form. [12, 13] describes it as transforming the text information into a visual form by considering the words, sentences and their relationship to make the user understand better and reduces the mental workload from facing massive text. Currently [14, 15] presented the survey on text visualization with a visualization browser and SoSVis [16]. Text visualization can be represented in many forms Tagcloud [17, 18], Word- cloud [19-21], Graph [22, 23], Graph-of-Words [24], chart [25], Map [26], Text data stream [27], Social networks [28] and others like timeline, tree-map [29], head-map, and spark-line. Users who have only working knowledge of computers prefer those text visualization which is capturing and attractive to the human eye like word-cloud. Because other forms like histograms, require a deep mathematical knowledge to interpret them. This makes word-clouds [30], a popular choice. It is used to depict words of input text document as arranged in space varied in size, color, and position based on word frequency, categorization, or significance [31]. A word that appears more in the text will be bigger and bolder in the word-cloud.

Several applications of text visualization are material science [32], health care [33], social media [34-36], services management [37], consumer reviews [38], Theme-crowds [39], and business performance analysis. Majority of users here are not well-versed with text mining and require only a visualization of their textual data. For such researchers ready-to-use visualization tools are very useful. Open-source tools are available for such purposes and this paper describes few such easy-to-use tools. The approach towards text visualization can be categorized into two based on the form of input [40]: (i) single text approach or (ii) collection of text approach. The entire corpus can be input to the visualization tool in the latter approach while the former limits the input to a single text file. This single file is also of limited size, hence the challenge of converting a corpus into a single text file that retains its text data characteristics.

In this paper, we have focused on this challenge. Various open-source text visualization tools are available for a single text representation only. So, we suggest here how to convert the corpus into a single text file and then visualize it with the help of visualization tools. The rest of paper first describes few popular and easy-to-use visualization tools, and then discusses "priority-window" techniques of converting a corpus into single text file. The methods are used to produce word-clouds of a corpus and those word-clouds are compared to demonstrate the effectiveness of our proposal.

TOOLS DESCRIPTION

This section briefly introduces some popular open-source text visualization tools, especially for word-clouds, with their salient features and major drawbacks.

A.WordArt /Tagul

WordArt is a good tool which provides several settings such as text color, shape of the word-cloud, size, and density of the word, fonts and more. Input is to be provided as a list of keywords. It has a much fancier look than most of the other tools. The main drawback of this tool is it neither allows single text file upload nor a corpus. This tool available at https://wordart.com/.

B.Tagcrowd

An easy-to-use tool that allows only pasting of text or single file upload. There are no options for changing shape, color, etc. The words are shown in alphabetical order with variation in size and thickness for emphasis. Tool is available at https://tagcrowd.com/

C.Word-It-Out

A simple tool with preset design options for word-clouds. The major drawbacks of this tool are (1) no option of file upload, (2) limited design options and (3) style and fonts are not up to the mark. This tool available at https://worditout.com/

D.Voyant

It is versatile tool that generates (1) Cirrus: It is a 'cloud' generated from the most frequent words; (2) Bubble lines: A word frequency graph throughout the text; and (3) Text arc: for word distribution and their interconnection. Besides pasting single text, user can upload single text file as URL, pdf, or MSWord format or entire corpus. This tool is available at https://voyant-tools.org/

E. WordSift

WordSift is collection of text analysis tools. Major features are word-cloud and visual Thesaurus. It allows only to paste text, with recommended limit up to 10000 words for analysis. The main advantage is that we can control the scale of the words, density, and orientation. This tool available at https://wordsift.org/

F. Wordclouds

Wordclouds is simple and much similar to the WordArt tool. The set of shapes for word-clouds are as per fields like traffic, love, pets etc. It has many settings for font, style, color palettes, size of cloud, invert, and masking options. The main drawback is (1) slow speed and (2) single text input. This tool available at https://www.wordclouds.com/

G. Jasondavies Word-Cloud

This tool is one of the best online word-cloud generators because it provides funny and exciting shapes. It allows only to paste text input. Another drawback is that we cannot change the settings like shapes and colors. This tool is available at https://www.jasondavies.com/wordcloud/

H. Daniel Soper's Word-cloud Generator

It is a free tool and easy to use because of its simple interface. The output cloud can be customized using the options panel. The drawbacks are (1) no option of uploading file and (2) not many options for shape, color, and design. This tool available at https://www.danielsoper.com/wordcloud/

I. Lexos [49]

Lexos is a web-based platform tool which has great resources for visualizing large text. This site allows uploading the entire corpus i.e., multiple files. Then prepare the data to visualize and analyze it. The output can be word-clouds, multi-cloud, bubbleviz, rolling-window graph and analysis like statistical analysis, clustering, similarity query, and top word. The drawback is that we have to login every time before using the tool. This tool available at http://lexos.wheatoncollege.edu/upload

J. Vizzlo

This tool has more potential than any other word-cloud generator because it has many paywalls. It allows the user to upgrade, change shapes, front and can fix the maximum numbers of words. One of the main features of this tool is used for removing the watermark from the word-cloud. If you can pay upfront, this tool allows you to change more settings and download files in PNG form with a transparent background. This tool available at https://vizzlo.com/create/wordcloud

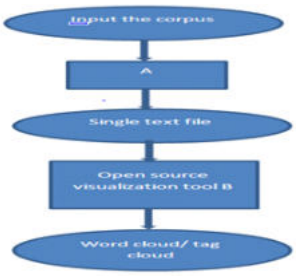
PROBLEM STATEMENT

The main challenge is that most of the tools are not taking text corpus as input rather these tools allow uploading single text file or giving the option of pasting the text to generate a word-cloud. Therefore, this is a very big gap for a research scholars and scientists because often the application is only for a corpus. They need a cloud or a visualization tool for entire corpus instead of a single text file. We present how this challenge can be overcome in the next section.

PROPOSED SCHEMATIC

The proposed schema for using the tools with corpus is explained with the help of flowchart in Figure

Figure 1: Flowchart of the proposed schema.



At first the entire corpus is given an input to block A, we get single text file. This single text file is then given to any open-source visualization tools (B) as mentioned above and finally we get a word-cloud/ tag cloud. Our main focus is on the block A. Converting a corpus of n text files to a single text file can be approached as follows:

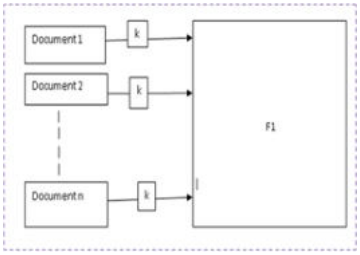
Naive – concatenate all the files and merge into a single file. The obvious drawback is the size of file, which already is a restriction in most of open-source tools. Also, the number 'n' itself may be too large to make a simple concatenation a memory-intensive operation. Hence, we propose two approaches of the following: There are three basic approaches already exists in text mining where a word is taken has its face value, a term is taken proportional to the frequency, and a term is taken proportional to the tf-idf. So, these three approaches have been listed here with one concept that we have introduce as priority window. The approaches are (1) selecting something from the bag which is priority window, (2) selecting something from the bag proportional to the frequency, and (3) selecting something from the bag which is proportional to its tf-idf.

1. Priority Window (KWExtract-k)

The first approach is the priority window (pw) approach that aims at collecting most important words from each file and that number of words (say k) is kept fixed and same for each file. Let us consider the size of window is k, that is k most common occurring words are to be considered from each document. Here the size of k can be set according to the user requirement. Thus, for a corpus of n documents extract k keywords from every file and concatenate into a single file of size kn. Value of k can be adjusted for tools having restriction on input-size. In our experiments we have observed that accepting at most 8 or 10 words from each document in the corpus is sufficient. By extracting those top k words from every document, we construct a single concatenated file containing all those words. Here the glitch is that suppose a particular term occurs in first document of the priority window may not be in the priority window of other document but occurring in the document then that term will get eliminated. Thus, those terms which are common to all documents and most occurring in all documents will get higher priority and words which rarely occur will not get priority in the entire corpus. So, we are eliminating the document bias because it might be that a long document has very higher frequency of particular word which is not occurring or very rarely occur in the other documents may gain a total higher frequency in the corpus. Therefore, only those words which are common to all documents and mostly occur in all documents will gain the priority, hence named as priority window. The main advantage of this approach is the elimination of document bias and the drawback is that only a document frequency and the popularity of the considered term plays a role in the priority window but not the individual term frequency is being considered. The formula is

F_1=concatenate (pw)

Figure 2: Schematic representation of priority window (KWExtract-k) approach.

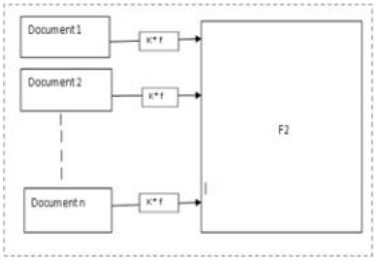


2. Priority Window with Frequency (KWExtract-f)

In order to overcome the drawback of the first approach that every priority word appears only once per document, we construct a single text file by picking all the documents, construct a priority window, pick all the words/terms that appear in the priority window and then repeat those words as many times as their frequency in that particular document so that the frequency of the term also becomes a consideration in this approach. In this single text file, the frequency of each word is going to be different. Here we have involved the frequency factor besides eliminating the document bias. We can observe in this approach that the frequency of the term in the corpus is not showing but the frequency of the term in the particular document is visible. So, the drawback cannot be clearly stated but a loophole can be identified in this approach where a term may be just out of the priority window of certain article and that frequency may not be contribute in the final single text file frequency of that particular term. Thus, it is possible in certain cases that a document having a particular term of very high frequency is able to affect the frequency of that term in final single text file that will in turn affect the visualization. The formula is

F_2=concat [pw*frequency]

Figure 3: Schematic representation of priority window with frequency (KWExtract-f) approach.



3. Priority Window with inverse document frequency (KWExtract- idf)

This approach is based on the inferences that have been already done in text mining in so many years. Several researchers agree that using inverse document frequency is a component eliminates the document bias and also it gives weightage to the terms which hold the importance in all over the corpus instead of in a single article. Just for the comparison to all the other approaches, we take an approach by taking the priority window and multiply all the terms with their inverse document frequency, so that we get a file which is equivalent to the tf- idf form of the corpus bag. The inverse document is computed using following the expression

$$idf = \log\left[\frac{N}{1+df}\right]$$

Where df is the document frequency. Therefore, the priority window with inverse document frequency can be defined as

F_3 =concat[pw*idf]

priority words are completely vanished from the visualization. Thus, reduces the visual noise and emphasis the important words. So, the advantage of using a frequency-based approach is very clear visual. The effect of our proposed schema KWEtract-f reduces the words which are not importance.

Table 1: Wordcloud obtained by using KWExtract-k, KWExtract-f and KWExtract-idf approaches from different open-source text visualization tools. (A)WordArt, (B) Tagcrowd, (C) WordItOut, (D) Voyant, (E)Wordsift, (F) WordCloud, (G) Jasondavies, (H) Daniel Soper's Word cloud, (I) Lexos, (J) Vizzol.

Tools	KWExtract-k	KWExtract-f	KWExtract-idf
A			
B			
C			
D			
E			
F			
G			
H			
I			
J			

Also, if we want to see particularly which terms have been emphasized and how the priority window approach affected the visualization of corpus, then we can see in Figures 5, 6, 7. On the x-axis, the numbers are the sequence numbers of different terms and the y-axis represent the term frequency. The peaks are showing the important words. Here the priority window runs y axis from max value to lower values and terms that occur in priority window are highlighted for any document. We have shown these figures for the entire corpus. In the figure, we can see that the words at the peaks are different in both tf and idf.

Observation of word-clouds

If manually we see that actually which words are of more importance or relevance to the entire article, then the word-clouds obtained through KWExtract-f approach are much

better than KWExtract- idf approach. So, it leads to the conclusion that KWExtract- idf approach cannot be used directly as a formula for creating single text files but term frequency can be used. This affects the visualization by making it more semantically effective.

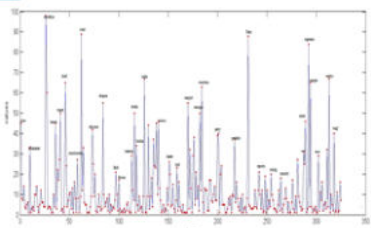


Figure 5: Priority Window (KWExtract-k).

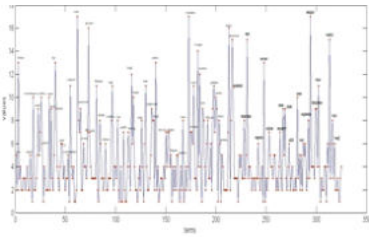


Figure 6: Priority Window with Frequency (KWExtract-f).

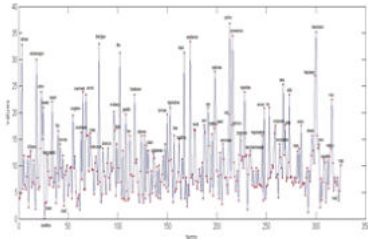


Figure 7: Priority Window with inverse document frequency (KWExtract-idf).

CONCLUSION

Text visualization is need of several non-technical fields too where researchers and users lack either knowledge or tools to produce effective visualization. Open-source tools are available to help people in such situations. Still there is a challenge to convert the corpus (collection of several text files) into a single text file such that its data characteristics are retained, because majority of visualization tools take a limited size single text file as input. This paper has proposed three "priority window approaches" using frequency and inverse frequencies of words in corpus. Using the proposed methods corpus can be converted to a small text file of only important terms and hence visualization produced is effective. For demonstration, we produce word-clouds from different tools using the priority window techniques and compare them. It is observed that a combination of word-importance and its frequency in individual documents gives effective output.

REFERENCES:

[1] D. Sarkar, "Text analytics with Python: Apractitioner's guide to natural language processing", 2nd edition, Apress, May 22, 2019. ISBN-10: 1484243536

[2] S. S.Bhoslay, M. Bali, "Text Analytics by Business Analytics Specialization", Book - Data Geek Text Analytics by Business Analytics Specialization School of Business and Management, Volume 3, Issue 1, April 2021.

[3] Rai, A. What is Text Mining: Techniques and Applications, 3rdi- 5 Common Techniques Used in Text Analysis Tools, Retrieved from 3rdi, 2018, September 12. https://www.3rdisearch.com/5- common-techniques-used-in-text-analysis-tools, June 01, 2019.

[4] S. Syed, T. Yousef, K. Al-Khatib, S. Jänicke, M. Potthast, "Summary Explorer Visualizing the State of the Art in Text Summarization", Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 185–194 August 1–6, 2021.

[5] S. Ying, Y. Zheng, W. Zou, "LongSumm 2021: Session based automatic summarization model for scientific document", Proceedings of the Second Workshop on Scholarly Document Processing, pages 97–102 June 10, 2021.

[6] Bhargav, A. Choudhury, S. Kaushik, R. Shukla and V. Dutt, "A comparison study of abstractive and extractive methods for text summarization", Advances in Intelligent Systems and Computing, March 2021.

[7] Dr. S. J. Mandal, "Deep Learning Powered Text Summarization Framework for Creating a Highly Accurate Summary", whitepaper, Data-matics Global Services Ltd., 2021.

[8] N. Iskender, T. Polzehl, S. Moller, "Reliability of human evaluation for text summarization: Lessons learned and challenges ahead", Proceedings of the Workshop on Human Evaluation of NLP Systems (HumEval), pages 86–96 Online, April 19, 2021.

[9] Awasthi, K. Gupta, P. S. Bhogal, S. S. Anand and P. K. Soni, "Natural Language Processing (NLP) based Text Summarization - A Survey", 2021 6th International Conference on Inventive Computation Technologies(ICICT), 2021, pp. 1310-1317, doi: 10.1109/ICICT50816.2021.9358703.

[10] A. P. Widyassari, S. Rustad, G. F. Shidik, E. Noersasongko, A. Syukur, A. Affandy, D. R. Ignatius, M. Setiadi, "Review of automatic text summarization techniques & methods", Journal of King Saud University - Computer and Information Sciences, 2020.

[11] A.K.Yadav, A.K.Maurya, Ranvijay, R.S. Ranvijay, "Extractive text summarization using recent approaches: A survey", Ingénierie des Systèmesd'Information, Vol. 26, No. 1, 2021, pp. 109-121.

[12] Q. Gan, M. Zhu, M. Li, T. Liang, Y. Cao and B. Zhou, "Document visualization: a verview of current research", WIREs Computational Statistics, vol 6, 2014 6:19–36. doi: 10.1002/wics.1285.

[13] N. Elmqvist, M. Hlawitschka, and J. Kennedy, "Visualizing Translation Variation of Othello: A Survey of Text Visualization and Analysis Tools", Supplementary Material, Eurographics Conference on Visualization (EuroVis), Eurographics Association 2014.

[14] K. Kucherand A. Kerren, "Text Visualization Browser: A Visual Survey of Text Visualization Techniques", poster abstract, IEEE Information Visualization(Infovis'14), Paris, France, 2014

[15] K. Kucher, A. Kerren, "Text Visualization Revisited: The State of the Field in 2019", Proceedings in the Eurographics Association, 2019.

[16] M. Alharbi and R. S. Laramee, "SoSTextVis: A Survey of Surveys on Text Visualization", Proceedings in the Eurographics Association, EG UK Computer Graphics & Visual Computing (2018).

[17] M. Hearst., D. K. Rosner, "Tag clouds: Data analysis tool or social signaller?", In Proceedings of the Hawaii International Conference on System Sciences (2008), pp. 160–160.

[18] S. Jänicke, J. Blumenstein, and M. Rücker, D. Zeckzer1 and G. Scheuermann, "TagPies: Comparative Visualization of Textual Data", International Conference on Information Visualization Theory and Applications, January 2018. DOI: 10.5220/0006548000400051.

[19] T. Kulahcioglu, G. Melo, "Paralinguistic Recommendations for Affective Word Clouds", IUI '19, March 17–20, 2019, Marina del Rey, CA, USA, ACM ISBN 978-1-4503-6272-6/19/03.

[20] B. Lee, N. H. Riche, and A. K. Karlson, S. Carpendale, "SparkClouds: visualizing trends in tag clouds", IEEE Transactions on Visualization and Computer Graphics 16, 6 (2010), 1182–1189.

[21] W. Cui and Y. Wu, S. Liu, F. Wei, and M. X. Zhou, H. Qu, "Word Cloud Visualization", Published by the IEEE Computer Society, 0272-1716/10/$26.00 © 2010 IEEE.

[22] S. Havre, E. Hetzler, P. Whitney, L. Nowell, "ThemeRiver: Visualizing thematic changes in large document collections", IEEE Transactions on Visualization and Computer Graphics 8, 1 (Jan. 2002), 9–20.

[23] F. B. Viegas, M. Wattenberg, J. Feinberg, "Participatory visualization with Wordle", IEEE Transactions on Visualization and Computer Graphics 15, 6 (2009), 1137–1144.

[24] J. Antoine, P. Tixier, K. Skianis, and M. Vazirgiannis, "GoWvis: a web application for Graph-of-Words-based text visualization and summarization", Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics—System Demonstrations, pages 151–156, Berlin, Germany, August 7-12, 2016.

[25] K. Koh, B. Lee, B. Kim, J. Seo, "ManiWordle: Providing flexible control over wordle", IEEE Transactions on Visualization and Computer Graphics 16, 6 (2010), 1190–1197.

[26] K. Kucher, C. Paradis and A. Kerren, "DoSVis: Document Stance Visualization", In Proceedings of the 13th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP '18) - Volume 3: IVAPP, pages 168-175, Funchal, Madeira - Portugal, 2018. SciTePress.

[27] F. Wanner, A. Stoffel, D. J¨ackle, B. C. Kwon, A. Weiler, and D. A. Keim, "State-of-the-Art Report of Visual Analysis for Event Detection in Text Data Streams", Computer Graphics Forum, 33(3), 2014.

[28] B. Preim, P. Rheingans, and H. Theisel,"Wordonoi: Visualizing the Structure and Textual Contents of Knowledge Networks", Eurographics Conference on Visualization (EuroVis) 2013, Volume 32 (2013), Number 3.

[29] M. Wattenberg, "Visual exploration of multivariate graphs", In Proceedings of the ACM Conference on Human Factors in Computing Systems (2006), pp. 811–819.

[30] F. Heimerl, S. Lohmann, S. Lange and T. Ertl, "Word Cloud Explorer: Text Analytics Based on Word Clouds", 2014 47th Hawaii International Conference on System Sciences, 2014, pp. 1833-1842, doi: 10.1109/HICSS.2014.231.

[31] J. N. Vilaplana, M. P. Montoro, "How we draw texts: a review of approaches to text visualization and exploration", Elprofesional de la informaci´on, mayo-junio, v. 23, n. 3, 2014 pp. 221-235.

[32] O. Kononova, T. He, H. Huo, A. Trewartha, E. A. Olivetti, and G. Ceder, "Opportunities and challenges of text mining in materials", *iScience 24, 102155, March 19, 2021

[33] M. Elbatta, E. Arnaud, M. Gignon, and G. Dequen, "The Role of Text Analytics in Healthcare: A Review of Recent Developments and Applications", Proceedings of the 14th International Joint Conference on Biomedical Engineering Systems and Technologies (BIOSTEC 2021) - Volume 5: HEALTHINF, pages 825-832, DOI: 10.5220/0010414508250832

[34] S. Chen, L.Yuan, and X. Yuan, "Social Media Visual Analytics", Computer Graphics Forum, June 2017 36(3):563-587, DOI: 10.1111/cgf.13211

[35] A. I. Kabir, R. Karim, S. Newaz , M. I. Hossain, "The Power of Social Media Analytics: Text Analytics Based on Sentiment Analysis and Word Clouds on R", InformaticaEconomica, April 2018, DOI:

10.12948/issn14531305/22.1.2018.03
[36] Y. Wu, N. Cao, D. Gotz, Y. P. Tan, and D. A. Keim, "A Survey on Visual Analytics of Social Media Data", IEEE Transactions on Multimedia, 18 (2016), 11. - S. 2135-2148, https://dx.doi.org/10.1109/TMM.2016.2614220.
[37] S. Kumar, A. K. Kar, P. Vigneswarallavarasan, "Applications of text mining in services management: A systematic literature review", International Journal of Information Management Data Insights, Volume 1, Issue 1, 2021, 100008, ISSN 2667-0968, https://doi.org/10.1016/j.jjimei.2021.100008.
[38] B. Alper, H. Yang, E. Haber, E. Kandogan, "OpinionBlocks: Visualizing Consumer Reviews, Conference", IEEE VisWeek Workshop on Interactive Text Analytics for Decision Making, October 2011.
[39] D. Archambault, D. Greene, J. Hannon, P. Cunningham, N. Hurley, "ThemeCrowds: Multiresolution Summaries of Twitter Usage", Conference PaperSMUC'11, October 2011, DOI: 10.1145/2065023.2065041
[40] J.W. Hong and S.B. Park, "The Identification of Marketing Performance Using Text Mining of Airline Review Data", Mobile Information Systems, Hindaw, Volume 2019, Article ID 1790429, 8 pages, https://doi.org/10.1155/2019/1790429