Implementation of a Framework for Collecting and Analyzing Data Based on Facets

Mrs.K Vandana[1], Mrs.Lakshmi Lavanya[2], Mrs. Kshetravathi N S[3]
Assistant Professor[1,2,3]
Department of CSE
Global Institute of Engineering and Technology

Abstract — ***In this study, we provide a unified approach to gathering and analyzing text data that is based on Facets. First, the user interface; second, the web crawler; third, the data analyzer; and fourth, the database (DB) make up the integrated framework. Web crawling and text data analysis both make use of user interfaces, which allow users to define input Facet and option settings using a graphical user interface (GUI). Data visualization is really how it provides study results. Based on the input Facets, the web crawler retrieves text data from articles published on the web. When processing papers, the data analyzer divides them into "relevant articles" (containing word sets for these types of postings) and "nonrelevant articles" using established criteria. After that, it takes the text data from the relevant articles and makes a visual representation of the findings. In the end, the database stores the created text data, the user-defined knowledge, and the results of data analysis and visualization. Through the use of proof of concept (PoC) prototyping, we demonstrate that an integrated framework is feasible. The testing findings demonstrate that the prototype is able to reliably gather and evaluate the article text data.***

Keywords— Data Analysis, Integrated Framework, Intelligent Service, Text Data Collection, Web Crawling.

I. INTRODUCTION

For applications like media treatment and choice research and recommendation, intelligent systems have lately attracted a lot of attention from both the academic and business communities.(1) to (3) Such systems frequently glean the necessary information by mining text-data from online documents. The Web system's most crucial characteristics are, in general, data collecting and analysis. "Internet sensor" refers to a certain kind of network-centered infrastructure that can collect, store, and analyze the data. Key facilitators of the sensor web for such intelligent services are, then, web crawling for text data collection and data analysis for text data analysis.

So far, a few ongoing studies have sought to provide this capability using open-source languages like R, Python, and Scala (5-7) However, the vast majority have little experience with big data analytics or automated web crawling. Most previous research has shown that this is necessary to differentiate between web crawling and big data analytics. A unified design of several functionalities (such as web crawling, data analysis, and user application) is necessary to make smart services smooth be developed (10) and, as a result, smart targeted services are prone to unanticipated delays due to the fact that their practicality is heavily dependent on the developer's skill.

To facilitate the collection and interpretation of text material based on Facet, the authors of this study propose an automated web-crawling platform and data analysis. The following components make up the suggested framework: There are four parts: (1) the interaction with the user, (2) the web crawler, (3). Data is exchanged by means of these components' interaction. The end-user

This component of the interface allows users to configure the text data analysis and web crawling settings using a graphical user interface (GUI). Word clouds and word intensity charts are two examples of the many content visualization outcomes that emerged from the analysis of text data. The web crawler's text data component reads articles on the web and stores the content in a database so that it may be analyzed. Data analyzer component uses web crawler data sets to do data pre-processing and analysis. In order to prepare data for processing, object identification is performed.

The papers will often be classified as "relevant articles" or "nonrelevant objects" according to predetermined criteria, such as a list of terms to be included. Articles that are relevant to the user's search are those that are extremely connected to the topic at hand, whilst articles that are not relevant are those that are not at all related. Analyzing the data is the third phase. Words consisting of three or more characters are extracted from the contents of related articles as the first stage. The second method involves filtering terms in such a way that superfluous words are removed. Word clouds and word frequency charts show the outcomes of the data analysis. Three databases, DB, DB, and DB, make up DB in the most recent research. The database part includes three databases. The outcomes of data analysis and visualization, as well as any predetermined knowledge, are stored in every database.We validate the viability of the unified system via proof of concept (PoC) prototyping. The UI is built using Java Swing frames, while the web crawler and data analysis are done using open-source R programs (11,12). The integrated frame provides the features of web crawling and text data analysis reliably, according to the experimental findings.

II. Functional Architecture of Integrated Framework

Figure 1 displays a user interface, internet crawler, information analyser and DB element operational structure in the suggested unified system. There are three logical elements in the user's interface component: input panel, display panel and a predefined information panel. The user feedback panel is used to set Facets and choices including the range of crawling sites, the minimum size of the word cloud and the word frequency rating. The scope of crawling pages is the list of domains on which the content of papers can be identified. The minimum frequency of the word cloud corresponds to the average frequency of the terms shown in the word cloud, one of the outcomes of information analysis. Note that the word "cloud" is an image of various words, each of them of different sizes. The regular word classification is a frequency metric for evaluating the terms in the word frequency map. After the data analysis, the result view panel is used to provide data view results. In the predefined knowledge panel, words are added or removed from the DB knowledge DB. The portion of the Web Cruiser involves data collector and file creator parts. The information collector practices sorting, aid for vocabulary and collection. Parsing includes scanning papers ' urls or browsing articles ' text information on certain blogs. For this function, the webpage of items based on the user-defined input Facet are checked with a standard Resource Locator (Rel) to count the number of articles ' web pages using their corresponding URL.

In fact, this generates a URL to crawl documents on the basis of the number of web pages for publications scanned through parsing. It then allows the relevant URL to be used as a hypertext markup (HTML) document for all posts in the database. Language aid requires UTF-8 encoding to avoid internet crawl data loss. Extracts text information from the HTML archive from the posts. The author of the database generates a derived data file and stores it in the DB component's ripple Database. There are three feature blocks in the data analyzer component: preprocessor, analyzer and visualize. The preprocessor extracts the collected text details of the papers contained in the crawling Database beforehand. By comparing the text data of the articles to a set of words in predefined knowledge, the preprocessor classifies these papers into relevant, non-relevant articles. If one of the terms is included in the document, it is defined as a corresponding article by the preprocessor feature section. In the DB research element review, the text information of the relevant articles are saved. In addition, non-relevant papers are discarded with text information.

Two steps are taken: selection of terms and sorting of phrases. In the extraction stage vocabulary is derived from a text information of the relevant articles, which consists of three or more characters. The unnecessary words (for example,' A," A," The,' and' The,') from the text data of the relevant articles are removed in the word filters step. The findings of the data analysis were saved in the DB element analyzer. The viewfinder visualizes information on the basis of the effects of data analysis and the alternative values defined through the user input table. The output is visualized as word clouds and frequency curves in the data visualization and stored in the analyzer DB. The DB modules are creeping DB, predefined DB and DB review. The crawling data collected from the Internet by the internet crawlers is stored by the Crawling DB. A set of terms is placed by the client in the predefined information DB via the predefined knowledge table. DB research holds the text information for the relevant posts, results of data analytics and results of data show.

III. Implementation

The Facet configurations of the output and the application to scan the articles ' webpage is performed in the team configuration box of the GUI. The field Query is used to pick the entry Facet in the category box and the button "Confirm" is used to check the articles ' WebPages based on the Facet data. This application was provided for the inspection for papers ' web pages. First of all, the Xml document with details is demanded about the total number of papers ' web pages.

In order to obtain the maximum number of web pages of the posts, the text information whose allocation is < select category= "search-header." The text data is then extract from the HTML document by xpathSApply). Last blanks and quotations are excluded from text information and text data is shown in the "Total Post" tab in a crawling community box (e.g., the maximum number of articles ' web pages). The scope of crawling pages is specified in the "Crawling Page" field in order to collect text information on posts. The internet crawling application is then performed with the "Crawling" key. The HTML file containing the content of posts on a given website and the document information with a div= "story Body" p > attribute meaning will be retrieved by the XPathSApply) (feature from the HTML file as you crawl through the site.

The whole list of crawling pages set by the client is constantly carried out with the site crawling method. The text information for all the papers published were eventually collected and stored in the DB. In the "Review" button, the information preprocessing and data analysis is done. The predefined information is used for identification of papers in software preprocessing. The terms can be inserted and omitted by the client in predefined information. In the predefined vocabulary community container, the "Insert" or "Delete" buttons are used to insert or delete terms from predefined information respectively. The str detect) (function tests whether the terms are found in the predefined information in the collected articles text content. In the case of a word contained in the article, the relevant article shall be classified as an article, or else it shall be classified as an article of no relevance. DB is used in research to store text information for relevant articles, thus non-relevant papers are excluded from the text data. Once data analysis is finished, information analytics are conducted to retrieve and filter terms.

For word extraction, the sapply) (method is used to extract terms consisting of three or more characters from the text information of the relevant articles. For word processing, the text information of relevant articles with the str detect) (feature delete unnecessary words. The data analysis findings are contained in the DB archive. Visualizing the information is done using the "View" key. In the form of words or term intensity charts, the effects of the data visualization are shown. This is done using the choice values in the team option box (i.e., term intensity rating and minimum word cloud size). Word clouds are created using the wordcloud() (function and the bar plot) function creates text frequency graphs. The DB research holds the output of data visualization.

IV. Result Analysis

A word frequency chart showing 10 of the most frequently extracted words from the articles with 10 words. The graph's and y axes are the word forms and terms rate, respectively. The frequency and intensity values of every term show on the chart on the line. In the figure,' cloud' frequency is 2132, and' cloud' frequency is 2%.

Parameter Values	Facet Data Analysis
Crawling page	300
Predefined	knowledge Data, IoT, Hadoop, Cloud
Filtering words	Good, will, then, the, are, with, and, that, this, but, have, has, can, for, you, from, been, more, they, said, what, its, about, how, was, which, their, into, these, when, there, those
Word cloud minimum frequency	150
Word frequency ranking	10

Table 1 Experimental parameters.

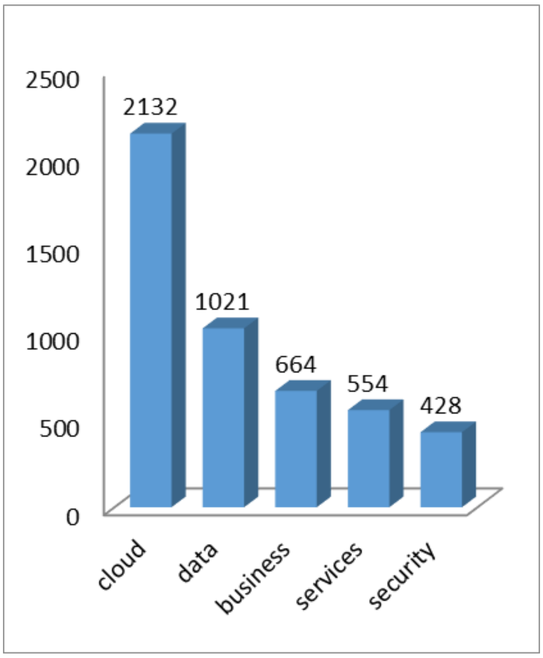

Fig1: Word frequency graph

V. Conclusions

In this paper they presented an automated site crawling and computational model for the processing and evaluation of Facet-based text content. Four components comprise the integrated frameworks: user interface, web crawling, data analyzation and DB. The user interface component helps the user to set the input Facet and the GUI option values. And the input Facet text data is compiled by a webcasting component. The information analyzer is used to preprocess, analyze data and model data. Finally, the DB component saves the text data collected, knowledge predefined, analysis results and results of data visualization. PoC prototyping was conducted to check the feasibility of the integrated framework. For the experiment, the text data of ZDNet items have been collected using the user's input Facet and text data gathered to monitor their frequencies have been analyzed. In a words-cloud and word frequency chart, the analysis results were visualized. The results showed that the integrated framework provides web crawling and text data analysis with reliable functions. The results show.

References

[1] Referenced in [1] Future Gener. Comp. Syst. 37 (2014) 267 by C. Dobre and F. Xhafa.

[2] Health Information Science System 2 (2014) by W. Raghupathi and V. Raghupathi.

[3] In the proceedings of the 2013 IEEE/ACM sixth international conference on utility and cloud computing, Z. Khan, A. Anjum, and S. L. Kiani were published on page 381. 78 in IEEE Internet Computing, 12 (2008) by A. Sheth, C. Henson, and S. S. Sahoo.

[4] Referenced in J. Enormous Data 2 (2015) 24 by S. Landset, T. M. Khoshgoftaar, A. N. Richter, and T. Hasanin.

[5] This is from the 2012 Springer publication "Proc. European Conf. Item Oriented Programming" by authors F. Morandat, B. Slope, L. Osvald, and J. Vitek, page 104.

[6] J. Funct. Program. 20 (2010) 303, B. C. D. S. Oliveira and J. Gibbons. Mesbah, A. V., and Lenselink, S. (2012): ACM Transactions on the Web, 6(1).

[7] Y. Zhang: IEEE Transactions on Service Computing, 9 (2016) 786.

[8] (Springer, 2016) Proc. International Conference on Modern Internet of Things Technologies and Applications, S. Wang, C. Zhang, and D. Li.

[9] (R) Foundation: https://www.r-project.org/ [11]continued till July 2017.

[10] http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html is Oracle's link for reference [12]. arrived in July 2017. Click here to visit ZDNet: http://www.zdnet.com/made it until July of 2017