IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 2, February 2017

Improved Tag Cloud by Cleaning Tags using Context Free Data Cleaning

Sohil D. Pandya[1], Rinku Chavda[2]

Assistant Professor, MCA Department, Sardar Vallabhbhai Patel Institute of Technology (SVIT), Vasad, India[1, 2]

Abstract: In this span of erudition of information, Universities can lead competitive advantage of searching of resources only by trained data analysis. This paper highlights context free data cleaning for improved tag cloud by correcting values of user defined "Tags", using different string similarity metrics, where "Tags" are assigned by users which related to referenced resource. Authors propose a procedure to scrutinize suitability of value to correct other values of Tags. Several string similarity metrics were used, to find distance of two different strings and generate results. Experimental results show how the approach can meritoriously clean the data without reference data.

Keywords: Context free data cleaning; Information Retrieval; String similarity metrics; Tag Cloud.

I. INTRODUCTION

As the degree of Internet is rising in recent ages, we need to expend more and more time on Internet to search for specific resource. Even to search within an organization document repository is a difficult and time-consuming task. Because they are unstructured in nature, unorganized in storage, naming conventions are different, but they are needed to be retrieved as and when required.

In an organization like University, various notifications, circulars, notes are published regularly in general. To become specific, authors have taken an example of Gujarat Technological University (GTU) from where the resources are tagged by users, which regularly updates its website contents.

In this paper, authors tried to implement improved tag cloud for information retrieval using data cleaning process, for which various string similarity metrics are used. Here tags with its frequencies which are mentioned by users are taken for this model as an input.

At the time of tagging of resource, users use their own word as tag for their comfort for future reference. Sometimes for one resource, multiple users use different or same tags to tag that resource [4]. While using same tags there may be possibility of correct spell of tag, incorrect spell of tag, similar king of tag or shortening of tag are used.

So, to come out from this situation, authors have used various similarity metrics where the tags are compared to find similarity between tags, perform replacement of tags and generate tag cloud for information retrieval from tag list which becomes fine-tuned list of tags after applying data cleaning process [5, 6, 11]. Actually the generated tag cloud is social signaller for how people uses words to tag [8, 10]. Hence, various socio-cultural aspects are by product for the researchers of other domain also [9].

II. DATA CLEANING FOR IMPROVED TAG CLOUD

Data cleaning is the process of noticing and altering corrupt or inaccurate tags from a record set and refers to identifying incomplete, incorrect, inaccurate or irrelevant parts of the data and then replacing data.
After cleaning, a reference dataset should be consistent among various users while tagging any resources. The inconsistencies detected or removed may have been originally caused by user entry errors or by different data dictionary definitions of similar entities in different stores.

A tag cloud (word cloud or weighted list in visual design) is a visual representation for text data, typically used to depict keyword metadata (tags) on websites, or to visualize free form text [1, 2].

Here, the core part of this paper is, based on the frequency of tags, in descending order, they are compared with other tags, and based on the compare value, it is decided that whether they should be replaced or not? [7, 13] The procedure described in next section examines appropriateness of tags to become member of reference dataset and/or replace the tags by other comparing tags which are frequently used among multiple users [7, 12].

III. CONTEXT FREE DATA CLEANING

The projected procedure has two major components: clustering and nearest string. It has an important parameter acceptable Dist, which is the minimum acceptable distance required during matching and transforming (ranges from 0.0 to 1.0, where 0.0 is not similar string and 1.0 is same string).To measure the distance we used following sequence similarity metrics:

1) Jaro Winkler Distance
2) Damerau Levenshtein Algorithm
3) Smith-Waterman Algorithm

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 2, February 2017

And for further process and validation we put emphasis on some of the above metrics based on initial results and their methodologies, which are discussed below:

The Jaro–Winkler [18] as in (1) is an extension of Jaro distance; it uses a prefix scale which gives more favourable ratings to strings that match from the beginning for a set prefix length.

$$d_j = \begin{cases} 0 & \text{if } m = 0 \\ \frac{1}{3}\left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m}\right) & \text{otherwise} \end{cases} \quad \ldots (1)$$

Where, m is the number of Matching characters and t is half the number of transpositions

The Damerau-Levenshtein distance [18] as in (2) is a distance (string metric) between two strings, i.e., finite sequence of symbols, given by counting the minimum number of operations needed to transform one string into the other, where an operation is defined as an insertion, deletion, or substitution of a single character, or a transposition of two adjacent characters.

$$d_{a,b}(i,j) = \begin{cases} \max(i,j) & \text{if } \min(i,j) = 0, \\ \min \begin{cases} d_{a,b}(i-1,j)+1 \\ d_{a,b}(i,j-1)+1 \\ d_{a,b}(i-1,j-1)+1_{(a_i \neq b_j)} \\ d_{a,b}(i-2,j-2)+1 \end{cases} & \text{if } i,j > 1 \text{ and } a_i = b_{j-1} \text{ and } a_{i-1} = b_j \\ \min \begin{cases} d_{a,b}(i-1,j)+1 \\ d_{a,b}(i,j-1)+1 \\ d_{a,b}(i-1,j-1)+1_{(a_i \neq b_j)} \end{cases} & \text{otherwise.} \end{cases} \quad \ldots (2)$$

Where, each recursive call matches one of the cases covered by the Damerau–Levenshtein distance:

- da,b(i-1,j)+1corresponds to a deletion (from a to b).
- da,b(i,j-1)+1} corresponds to an insertion (from a to b).
- da,b(i-1,j-1)+1(ai ≠ bj)corresponds to a match or mismatch, depending on whether the respective symbols are the same.
- da,b (i-2,j-2)+1 corresponds to a transposition between two successive symbols.

The Smith-Waterman algorithm [18], as in (3) is well-known algorithm for performing local sequence alignment, i.e. for determining similar regions between two strings or tags. It compares segments of all possible lengths and optimizes the similarity measures using substitution matrix and gap scoring scheme [6].

$$H(i,0) = 0, 0 = i = m$$
$$H(0,j) = 0, 0 = j = n$$

$$H(i,j) = \max \begin{cases} 0 \\ H(i-1, j-1) + w(S_{1i}, S_{2j}), \text{ Mismatch} \\ H(i-1, j) + w(S_{1i}, -), \text{ Deletion} \\ H(i, j-1) + w(-, S_{2j}), \text{ Insertion} \end{cases} \quad \ldots (3)$$

Where S1, S2 are strings and m, n are their lengths; H (i, j) is the maximum similarity between strings of S1 of length i andS2 of length j; w(c,d) represents gap scoring scheme.

The algorithm consists of following steps:

1. Convert all the Alphanumeric values to Number format e.g. I,one,First,1st, 1 ST to1
2. Keep list of Domain Specific entries of tags e.g. Degree Engineering, Deg. Engi., Bachelor of Engineering to B.E.
3. Retrieve list of tags (listing) with its frequency in descending order.
4. Repeat while (listing has tags to compare)

a. Read tag to compare from listing
b. Retrieve list of tags (listj) with its frequency in descending order where freq(tagj) ≤ freq(tagi) and tagi∉ listj.
c. Repeat while (listj has tags to compare)
   i. Convert tagi and tagj to lowercase
   ii. Compare tagi with tagj
   iii. If the compare value is greater or equal 0.9 thresholds value, then perform replacement of tags else keep that two tags as a separate tags.

IV. EXPERIMENTAL RESULTS

The algorithm is tested using a sample data derived from user account of http://www.delicious.com which is one of the popular website for social bookmarking over the Internet; it is also called web based tagging system. It not only allows adding URL as a bookmark, but it also allows adding some extra information related to the URL like title, keywords and remark [14, 15, 16, 17]. The data consisting of tags with its frequencies. Based on that tags and its frequencies, the tag cloud is as below:

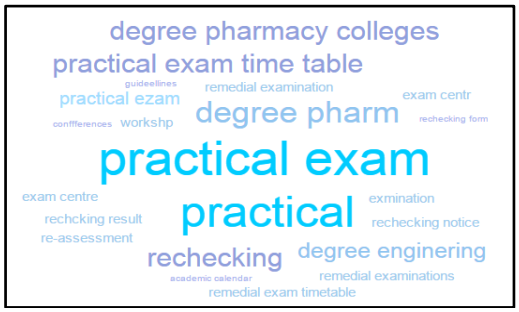

Fig. 1: Before applying the Data Cleaning process

For selected string similarity metrics several results, like how many records replaced (total, correctly, incorrectly, not replaced in context spelling mistake), were found and are discussed in this section. Here is an example given for one similarity metrics, i.e., for Jaro–Winkler algorithm with 0.9 similarity metrics value, we found replacement rules as shown in Table 1. The table is showing count for replacement of tags with correct tags where the tags are misspelled.

After applying Jaro-Winkler algorithm, TABLE 1 shows values of total records, replaced records, correctly replaced, incorrectly replaced, not replaced where the acceptableDis is greater or equal to 0.9. There were about 1556 records out of which 383 values were identified as

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 2, February 2017

correctly replaced (316: 82.51%) and incorrectly replaced (67: 17.49 %) and the generated tag cloud is as below:

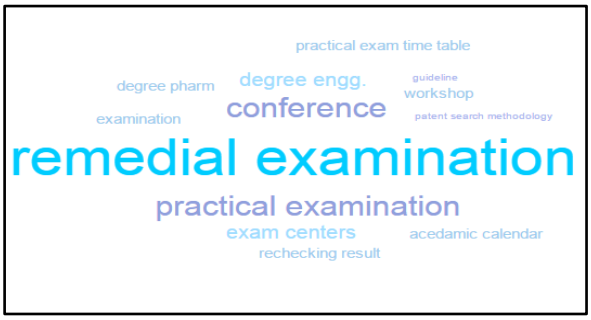

Fig. 2: After applying the Data Cleaning process

TABLE I Count of replacement as correctly, incorrectly and not replaced

total records	replaced records	correctly replaced	incorrectly replaced	not replaced
1556	383	83	17.49	1.57

There are 1556 records out of which 61 records contains incorrect tags which are entered by users. Using Jaro-Winkler algorithm, 55 records out of 61 records of incorrect tags are replaced with correct tags (TABLE II). Hence, from correctly replaced list of tags, 1.57 % tags remains unchanged.

TABLE II Incorrect tags which are altered with appropriate correct word

tag	tag frequency	replaced with	tag frequency
practical ezamination	10	practical examination	46
practical ezam	9	practical exam	35
rechcking Result	7	rechecking result	20
exmination	7	examination	16
workshp	5	workshop	15
exam centr	5	exam centers	30
patent search mathodology	4	patent search methodology	5
acedamic calendar	3	academic calendar	15
submision	3	-	-
guideellines	3	guideline	5
analyaia	2	-	-
conffferences	2	conference	50
remidial	1	-	-

Following results, percentage of correctly replaced (CR %), percentage of incorrectly replaced (IR %) and percentages of not replaced (NR %) were derived as in (4).

$$\left.\begin{aligned} CR(\%) &= \frac{CR * 100}{ReplacementDone} \\ IR(\%) &= \frac{IR * 100}{ReplacementDone} \\ NR(\%) &= \frac{NR * 100}{ReplacementDone} \end{aligned}\right\} \quad \ldots (4)$$

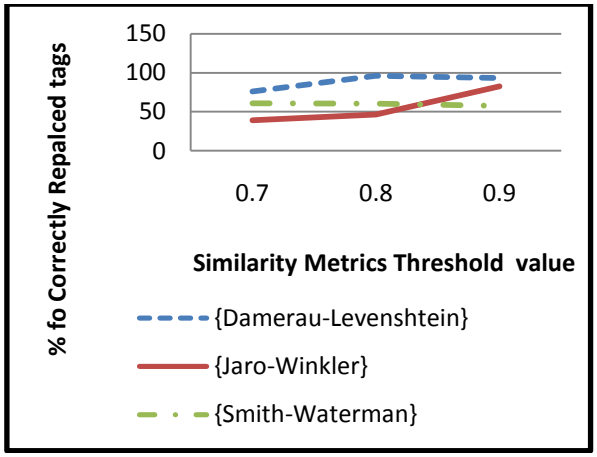

Fig. 3. Percentage of correctly replaced tags

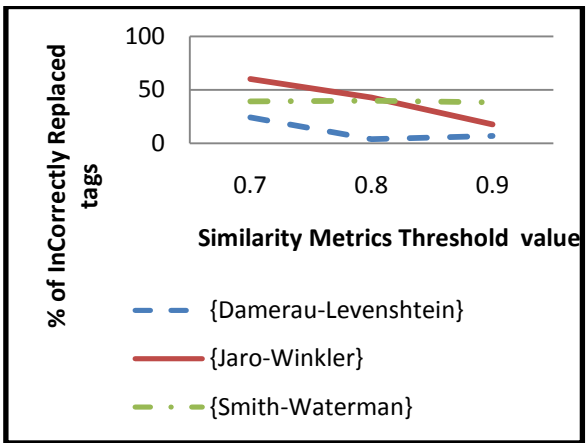

Fig. 4. Percentage of incorrectly replaced tags

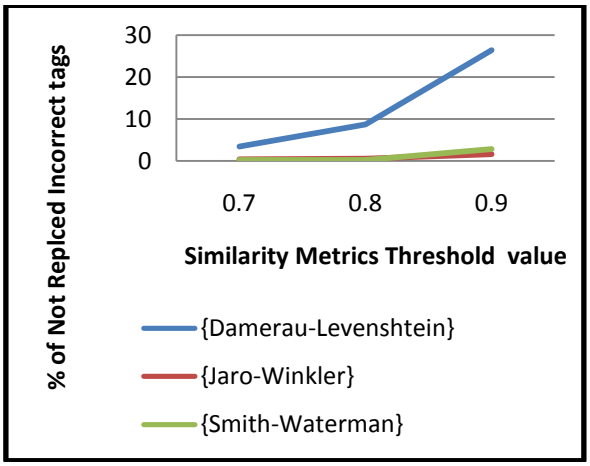

Fig. 5. Percentage of not replaced tags

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 2, February 2017

Consequences found on testing of algorithms are:

1. By looking further we found that percentages of correctly replaced values are increased and percentages of not replaced values are decreased as acceptableDist for various string similarity metrics is increasing but the percentages of incorrectly replaced values are also being decreased as shown in Fig. 3, Fig. 4, and Fig. 5.
2. For instance, using Jaro-Winkler algorithm with distance values 0.9, 0.8, 0.7 there were 82.51%, 46.57%, 39.26% values were replaced correctly respectively, 17.49%, 42.58%, 59.92% values replaced incorrectly respectively, 1.57%, 0.46%, 0.34% values are not replaced respectively, with respected to total replaced values.
3. The major disadvantage of the algorithm is to incorrectly classify some values (generally in earlier passes) even if they are correct in real world context.

V. CONCLUSION

The outcomes of the experiments verify the correctness of the algorithm and which inspire to use it for context free data cleaning.

In above experiments various string similarity metrics were used. It is possible that other metrics or functions and/or various combinations of them, as per the requirements, may give better results and this should be explored in further experiments.

REFERENCES

[1] A.W. Rivadeneira, D.M. Gruen, M.J. Muller, and D.R. Millen, "Getting our head in the clouds:Toward evaluation studies of tagclouds", Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 2007, pp. 995–998.

[2] Hearst, Marti A., and Daniela Rosner. "Tag clouds: Data analysis tool or social signaller?" In Hawaii International Conference on System Sciences, Proceedings of the 41st Annual, pp. 160-160. IEEE, 2008.

[3] James Sinclair, and Michael Cardew-Hall, "The folksonomy tag cloud: when is it useful?" in Journal of Information Science, Vol 34 No 1, pp. 15-29, May 2007.

[4] Kim, Hak-Lae, John G. Breslin, Sung-Kwon Yang, and Hong-Gee Kim. "Social semantic cloud of tag: Semantic model for social tagging." In Agent and Multi-Agent Systems: Technologies and Applications, pp. 83-92. Springer Berlin Heidelberg, 2008.

[5] Knautz, Kathrin, Simone Soubusta, and Wolfgang G. Stock. "Tag clusters as information retrieval interfaces." In System Sciences (HICSS), 2010 43rd Hawaii International Conference on, pp. 1-10. IEEE, 2010.

[6] Kuo, Byron YL, Thomas Hentrich, Benjamin M. Good, and Mark D. Wilkinson. "Tag clouds for summarizing web search results." In Proceedings of the 16th international conference on World Wide Web, pp. 1203-1204. ACM, 2007.

[7] Lukasz Ciszak "Application of Clustering and Association Methods in Data Cleaning", in Proc. of Int. Multiconference on Computer Science and Information Technology, Vol. 3, pp. 97-103, 2008.

[8] M.A. Hearst, and D. Rosner, "Tag clouds: Data analysis tool or social signaller?", Proceedings of 41st Hawaii International Conference on System Sciences (HICSS 2008), Social Spaces minitrack, 2008.

[9] Satoshi Niwa, Takuo Doi, and Shinichi Honiden, "Web Page Recommender System based on Folksonomy Mining" in Proc. of the Third International Conference on Inforrmation Technology: New Generation (ITNG'06), pp. 388-393, April 2006.

[10] Seifert, Christin, Barbara Kump, Wolfgang Kienreich, Gisela Granitzer, and Michael Granitzer. "On the beauty and usability of tag clouds." In Information Visualisation, 2008. IV'08. 12th International Conference, pp. 17-25. IEEE, 2008.

[11] Sohil D. Pandya, Paresh V. Virparia and Rinku Chavda "Implementation of Folksonomy based Tag Cloud Model for Information Retrieval from Document Repository in an Indian University", International Journal on Soft Computing, Artificial Intelligence and Applications (IJSCAI), Vol.5, No.1, February 2016.

[12] Sohil D Pandya, Dr. Paresh V Virparia "Data Cleaning in Knowledge Discovery in Databases: Various Approaches", in Proc. of National Seminar on Current Trends in IT (CTICT) – 2009, February 2009.

[13] W Cohen, P Ravikumar, S Fienberg "A Comparison of String Distance Metrics for Name-Matching Tasks" in Proc. of the IJCAI-2003.

[14] https://delicious.com/developers

[15] https://github.com/lucaong/jQCloud

[16] http://jquery.com

[17] http://www.w3schools.com/jquery

[18] http://en.wikipedia.org

BIOGRAPHIES

Dr. Sohil D. Pandya is working as an Asst. Professor at Sardar Vallabhbhai Patel Institute of Technology (SVIT), Vasad. He has 12+ years experience of teaching in MCA. He has published more than 6 Papers in International/ National Journals and presented more than 6 papers in International/ National conferences. His research interest includes data mining and information retrieval. He is an editor and editorial review board member in some journals.

Prof. Rinku Chavda is working as an Asst. Professor at Sardar Vallabhbhai Patel Institute of Technology (SVIT), Vasad. She has 6+ years experience of teaching in MCA. She has published 2 Papers in International/ National Journals. She has expertise in Android App development and has guided several Android App projects.