Consolidating Client Names in the Lobbying Disclosure Database Using Efficient Clustering Techniques

[1]Rajan Kumar Kharel, [1]Niju Shrestha, [1]Chengcui Zhang, [2]Grant T. Savage, [2]Ariel Smith
[1]Department of Computer & Information Sciences
[2]COLLAT School of Business
University of Alabama at Birmingham
Birmingham AL, 35294

ABSTRACT
A fuzzy-matching clustering algorithm is applied to clustering similar client names in the lobbying Disclosure Database. Due to errors and inconsistencies in manual typing, the name of a client often has multiple representations including erroneously spelled names and sometimes shorthand forms, presenting difficulties in associating lobbying activities and interests with one single client. Therefore, there is a need to consolidate various forms of names of the same client into one group/cluster. For efficient clustering, we applied a series of preprocessing techniques before calculating the string distance between two client names. An optimized threshold selection has been adopted, which helps improve clustering accuracy. A single linkage hierarchical clustering technique has been introduced to cluster the client names. The algorithm proves to be effective in clustering similar client names. It also helps to find the representative name for a particular client cluster.

Categories and Subject Descriptors
I.5.4 [Application]: Text Processing

General Terms
Standardization, Algorithm, Performance

Keywords
String similarity, Levenshtein distance, lobbying disclosure data

1. INTRODUCTION
The US Lobbying Disclosure Act (LDA) instituted a requirement that organizations lobbying Congress and the executive branch register with Congress. This provides data to empirically research many of the popular claims and scholarly conclusions about interest group behavior and influence on congressional policymaking. In particular, the LDA dataset [4] provide a detailed and systematic account of the lobbying organizations' activity, including a general description of organization's business or activities, lobbying expenditures, the target of the lobbying activity (the House, the Senate, or the executive branch agency) and the issue for which the organization lobbied.

However, the characteristics of the LDA data present challenges because the files contain duplicates, missing values, variation in client and registrant names, and many other issues that require expertise in working with data. For example, there is no consistency in the completeness of the organization names or the way names are abbreviated or punctuated. A technique such as fuzzy string matching and clustering can be used to ensure that a collection of reports are appropriately attached to a single client rather than being attributed to more than one client that simply represent variations in the name that exist in the data.

Similarity/distance measures play an important role in fuzzy string matching and clustering. Choosing a proper preprocessing, distance measure, and clustering method is crucial for achieving expected clusters and often depends upon the data to be clustered. Consider this LDA dataset (consisting of names of companies, organizations, individuals, etc.) where each unique name string, though intuitively represent a unique entity, in many cases may be short hands or even erroneous forms of the actual representation (e.g., *ABC Private Limited* is the same as *ABC Pvt Ltd.* and *ABC Private Ltd.*). Even though it is relatively easy for a human to observe the obvious similarity between these variations, the sheer volume of the data prohibits manual cleanup.

We propose a similarity measure for a two-phase automated clustering of such name strings. The first phase, preprocessing, temporarily removes special characters, extra spaces, punctuation marks, stop words, and periods and replaces abbreviations and synonyms with a single consistent term in order to reduce data inconsistency prior to the clustering process. The second phase applies the proposed similarity measure in clustering. This phase involves finding a keyword that ought to match exactly in the string being compared in addition to satisfying certain threshold of a standard similarity measure such as Levenshtein's Distance [2]. For example, the keyword in the above example can be the first word 'ABC.' We refer to this constraint as 'first word matching' constraint.

The experiments were conducted on a large dataset of client names collected from the LDA dataset. The results from our proposed algorithm outperformed those received by employing standard similarity measures only. The rest of the paper explores the details of the proposed client name-clustering algorithm.

2. RELATED WORKS
String matching techniques have been applied in diverse areas. Cortelazzo et al. [1] used string-matching technique to measure the shape similarity of two-dimensional objects. Ristad et al. [5] used stochastic model to learn a string-edit distance function. Wei et al. [6] used a string-matching algorithm to cluster subject lines of spam emails, in which a modified scoring strategy implemented in dynamic programming was applied to detect similar textual patterns. Dinu et al. [3] used rank distance (RD) to measure the similarity between strings. Rank distance is metric which measures the similarity between two hierarchies based on the rank of the objects.

ACMSE'14, Month 1–2, 2010, City, State, Country.

3. ALGORITHMS

3.1 String Matching

Levenshtein distance [2] is one of the common ways that calculates the edit distance between two strings, i.e., the minimum number of single-character insertions, deletions, or replacements required to convert one string into another.

After a thorough analysis of the data, it was observed that often times, client names representing the same entity show up as unique with the presence of extra characters or stop words. For effectiveness of the clustering process, these extra characters are temporarily removed before name strings are compared while they remain unchanged in the output. After applying the pre-processing techniques, we check the "first word matching" constraint as mentioned in Section 1. In the next step, Levenshtein distance is calculated in the following way.

Consider x and y are two strings with length m and n. x_i and y_j are two characters of x and y strings at i^{th} and j^{th} positions, respectively. M is the distance matrix initialized with m+1 rows and n+1 columns. The entry $M_{i,j}$ represents the cost of converting substrings $x_{0\ldots i-1}$ to $y_{0\ldots j-1}$ ($x_{0\ldots 0}$ and $y_{0\ldots 0}$ both represent empty strings). Steps involved in computation of the distance matrix are described as follows.

$M_{1,1} \leftarrow 0$ // the cost of converting one empty string to another is 0
$M_{1,j} \leftarrow j$; $M_{i,1} \leftarrow i$
// $x_{0\ldots 0}$ (empty string) can be converted into $y_{0\ldots j-1}$ by inserting j characters at a cost of j
// $x_{0\ldots i-1}$ can be converted into $y_{0\ldots 0}$ (empty string) by dropping i characters at a cost of i
$M_{i,j} \leftarrow$ Min ($M_{i-1,j}+1$, $M_{i,j-1}+1$, $M_{i-1,j-1}$+D (x_{i-1}, y_{j-1}))
D (x_i, y_j) = 0 if $x_i = y_j$, otherwise 1.

If the characters x_{i-1} and y_{j-1} of the strings x and y are equal then to convert the string $x_{0\ldots i-1}$ to $y_{0\ldots j-1}$ is $M_{i-1,j-1}$. If they are not, either use substitution, deletion, or insertion at the cost of $M_{i-1,j-1}+1$, $M_{i-1,j}+1$ and $M_{i,j-1}+1$, respectively to equalize. The edit effort that yields the minimum cost will be selected as the optimal edit. The matrix entry value at the position $M_{m+1,n+1}$ gives the minimum edit distance between the strings x and y.

3.2 Clustering

String clustering is to associate/cluster similar strings based on their structural similarity in an unsupervised way. We used Hierarchical Agglomerative Clustering technique to cluster similar strings. The details of the clustering process are summarized as follows.

Input: Strings set $S = \{s_1, s_2, s_3 \ldots s_n\}$, Threshold = t

Output: m clusters $\{C_1, C_2 \ldots C_m\}$

Step 1: Each string forms a cluster by itself and serves as the cluster centroid.
Step 2: Calculate the inter-cluster distance of each pair of clusters by calculating the distance between their cluster centroids.
Step 3: Merge the two clusters (into one) with the minimum centroid distance less than the threshold value.
Step 4: Update the centroid of new (merged) clusters. To select the new centroid, pair-wise distances between cluster members are calculated. And the cluster member that yields the minimum sum of distance with all the other cluster members will be selected as the new centroid for the merged cluster.
Step 5: Continue with Steps 2~4 until there are no more clusters that can be merged.

3.3 Experimental Dataset

The experimental dataset is selected from the lobby registration files collected over the period of January 1999 to October 2013 [4] that consists of 65,088 client names. For threshold selection purpose, a randomly selected set of 1,010 client names was used to create 300 ground truth client name classes manually. An optimum threshold determined from the process detailed below was used to cluster the entire dataset. For testing and evaluation purposes, we randomly selected 1,000 client clusters with 2,625 client names (no overlap with the training set) based on which the cluster quality was evaluated.

3.4 Threshold Selection

The main goal of clustering is to group similar client. The accuracy of clustering thus depends on the distance threshold t, as shown in Section 3.2. This signifies why choosing an optimal threshold is essential in clustering. In this study, an optimal threshold value is determined by using the training dataset. We carried out 10 rounds of clustering for the dataset varying the distance threshold from 1 to 10 with a step size of 1. At each round, we collected the number of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). Finally, we chose a threshold corresponding to a round that gave us the highest F measure score.

A false positive (FP) is a wrong grouping/pairing of two names that belong to two different ground truth classes into a cluster. A false negative (FN) is a wrong separation of two names that belong to the same ground truth class into two separate clusters. The definitions of TP and TN follow the same principle. Precision, Recall and F measure are calculated using the following equations.

$$Precision = TP/(TP + FP) \quad (1)$$

$$Recall = TP/(TP + TN) \quad (2)$$

$$F(N) = (1 + N^2)\frac{Precision \times Recall}{N^2 \times Recall + Precision} \quad (3)$$

Since precision is considered more important than recall in this particular application, we attach as three times the importance to precision as to recall, i.e., N=3 in calculating F measure (Equation 3).

Table 1. FP, FN, TP, TN, Precision, Recall and F-scores under different thresholds without 'first word matching'

t	TP#	FP#	FN#	Precision	Recall	F-3
1	620	12	1150	0.9810	0.3502	0.8312
2	772	29	998	0.9637	0.4361	0.8596
3	**852**	**31**	**918**	**0.9648**	**0.4813**	**0.8767**
4	910	69	862	0.9295	0.5135	0.8598
5	1278	753	500	0.6292	0.7187	0.6371
6	1372	1706	406	0.4457	0.7716	0.4653
7	1405	2019	377	0.4103	0.7884	0.4309
8	1427	6717	355	0.1752	0.8007	0.1900
9	1506	14752	286	0.0926	0.8404	0.1016
10	1568	27522	224	0.0539	0.8750	0.0594

From Table 1 and 2, we observed that when the threshold value increases from 1 to 4, there is no significant difference in their performance in terms of F-3. However, the performance of the algorithm without 'first word matching' drops sharply after a threshold of 5 and above (due to a big increase in FPs), while the performance of the one with 'first word matching' remains

relatively stable. The best threshold for 'no first word matching' is at threshold *t*=3, and that for 'first word matching' is at t=5. These two threshold values were then used in testing the test dataset.

Table 2. FP, FN, TP, TN, Precision, Recall and F-scores under different thresholds with 'first word matching'

t	TP#	FP#	FN#	Precision	Recall	F-3
1	620	12	1150	0.9810	0.3502	0.8312
2	728	26	1042	0.9655	0.4112	0.8508
3	800	26	970	0.9685	0.4519	0.8691
4	852	26	920	0.9703	0.4808	0.8806
5	**1192**	**82**	**586**	**0.9356**	**0.6704**	**0.8999**
6	1303	148	475	0.898	0.7328	0.8782
7	1319	167	463	0.8876	0.7401	0.8702
8	1370	209	412	0.8676	0.7687	0.8565
9	1395	238	387	0.8542	0.7828	0.8464
10	1432	247	350	0.8528	0.8035	0.8476

4. RESULTS AND DISCUSSIONS

To evaluate the performance of the clustering algorithm, both with and without 'first word matching,' was performed on the test dataset as mentioned in Section 3, with their respective optimal threshold values used.

Table 3. Comparison results

	First word matching (*t*=5)	No match of first word (*t*=3)
TP #	2120	1553
FP #	116	67
FN #	2009	2538
Precision	0.9481	0.9586
Recall	0.5134	0.3796
F-3	**0.8740**	**0.8317**

It can be clearly seen from Table 3 that the proposed clustering algorithm with the 'first word matching' constraint outperformed the algorithm without the constraint, measured by F-3 score. In particular, the precision of the algorithm with the constraint is very close to that of the one without the constraint, but the former exhibits a much higher recall. We also performed experiments on threshold selection and clustering using different F measures such as F-2 and F-5, and in all cases the clustering algorithm with the 'first word matching' constraint outperformed the other.

"ANHEUSER-BUSCH COMPANIES INC"
"ANHEUSER-BUSCH COMPANY"
"ANHEUSER-BUSCH COMPANIES, INC"
"ANHEUSER-BUSCH COMPANIES"
"ANHEUSER-BUSCH COMPANIES, INC."
"ANHEUSER-BUSCH COS"
"ANHEUSER-BUSCH COS INC"

Figure 1. A sample cluster

Figure 1 shows a sample cluster. All the client names in that cluster represent the same unique client. Using the centroid selection technique described in Section 3.2, the centroid of this cluster is selected to be "ANHEUSER-BUSCH COMPANIES, INC."

"International Text Group"
"International Textile Group"

Figure 2. A FP example

One example of FP is given in Figure 2. "International Textile Group" has been wrongfully assigned to the "International Text Group" cluster, while they actually represent two different clients.

"Intercontinental Exchange, Inc."
"IntercontinentalExchange, Inc."

Figure 3. FN examples

An example of FN is given in Figure 3. Two clients "Intercontinental Exchange, Inc" and "IntercontinentalExchange, Inc." have been clustered into two different clusters, because these two names failed the first-word matching criterion.

5. CONCLUSIONS AND FUTURE WORKS

An automatic clustering algorithm for consolidating US house client names was developed. The two-phase clustering approach is comprised of the pre-processing phase and the clustering phase. The latter phase was based on the proposed similarity measure, which requires exact matching of the first word plus satisfying a cluster distance criterion. Furthermore, an optimal threshold selection process was incorporated to reduce the sum of FPs and FNs, improving the clustering accuracy.

In future, we will further explore various patterns of client names. There are a few entities that in due course of time changed their names but append their former name to the new names, an issue not completely solved. Further, using one single threshold value may introduce biasness towards the strings that have a length close to the majority of the string lengths in the dataset. Therefore, it is more desirable to use an adaptive threshold technique that takes into account the length of the specific centroid string in clustering.

6. ACKNOWLEDGMENT

This work is supported in part by Natural Science Foundation under grant IOS-1127911.

7. REFERENCES

[1] Cortelazzo, G., Mian, G. A., Vessi , G., and Zamperoni, P. 1994. Trademark shapes description by string-matching techniques. *Pattern Recognition*, 27(8):1005-1018, 1994.

[2] Levenshtein, V. I. 1966. Binary codes capable of correcting insertion and reversal. *Sov. Phys. Dokl.,* 10, 707-710.

[3] Dinu, L. P. and Ionescu, R.-T. Clustering Methods based on Closest String via Rank Distance. 14th International Symposium on Symbolic and Numeric Algorithm for Scientific Computing, SYNASC 2012, Timisoara, Romania, September 26-29, 2012. IEEE Computer Society 2012 ISBN 978-1-4673-5026-6 http://www.computer.org/csdl/proceedings/synasc/2012/4934/00/4934a207-abs.html

[4] Lobbying disclosure databases: http://www.senate.gov/legislative/Public_Disclosure/database_download.htm

[5] Ristad, E.S. and Yianilos, P.N. 1998. Learning string-edit distance. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 20(5):522-531, 1998.

[6] Wei, C., Sprague, A., and Warner, G. 2009. Clustering malware-generated Spam Emails with a Novel Fuzzy String Matching Algorithm. In *Proceedings of the 2009 ACM symposium on Applied Computing* (Honolulu, Hawaii, USA, Mar. 9-12, 2009). SAC'09. ACM, New York, NY, 889-890. DOI= 10.1145/1529282.1529473