International Journal Of Engineering Research And Development
e-ISSN: 2278-067X, p-ISSN: 2278-800X, www.ijerd.com
Volume 13, Issue 11 (November 2017), PP.56-61

Automated Chat Response and Query Grouping using Data Mining Techniques

*Sunil Bhutada[1], Sandhya Yedama[2]
[1] *Department of IT, SNIST, Hyderabad, India.*
[2]*M.Tech(CNIS), SNIST, Hyderabad, India.*
*Corresponding Author: *Sunil Bhutada*

ABSTRACT:- We are living in a world full of data. The amount of data that is created each day is increasing rapidly. Every day, people encounter a significant amount of information and store or represent it as data, for further analysis and management. One of the essential means of dealing with this data is to organize or group them into a set of clusters or categories. In our paper, we discuss, how an organization's internal chat messaging system can be automated. In the chat messaging system, employees of the team can post queries and also other employees who know the solution to the query can answer it. The queries may not be immediately followed by a solution, as many users post queries and solutions. These queries and solutions are mixed up, as the users randomly post them. To make the chat messaging system more organized and responsive, we propose a novel method to handle a search query and automatically group relevant queries. Then every query and query result is analyzed. Similar/Related queries, as well as query results, are grouped using clustering method. Whenever a user enters a new search query, it is analyzed, and relevant query results are displayed from the query group it belongs.

Keywords:- Query grouping, Query Similarity, clustering, Automatic chat response, query relevance.

Date of Submission: 13-10-2017 Date of acceptance: 02-11-2017

I. INTRODUCTION

In today's world, most of the data generated is digital. Hence, the digital data needs to be converted into meaningful information. Everyone from an individual to an organization is dependent on information systems. Considering an organization, they employ information systems to store and process data. It is vital to organize and group related data. Feldman et al.[1] is the first who introduced Text Mining or knowledge discovery from text (KDT). It extracts quality information from text [2]. Information Extraction is a job which automatically extracts facts from unstructured or semi-structured documents [3]. Unsupervised learning methods are the methods which try to find unseen structure out of unlabeled data. They do not require any training phase. Hence can be applied to any text data without a lot of effort. Topic modelling and clustering are the two usually used unsupervised learning algorithms. Grouping is the job of segmenting a group of text data into divisions where text in the same group (cluster) is similar to each other. In topic modelling, a probabilistic model is used to define a soft clustering, in which every text data has a probability function over all the clusters as opposed to a tight grouping of texts. In topic models each text data is represented as probability distribution above topics and each topic is expressed as probability distributions above words [4]. Thus, a topic is similar to a cluster, and the relationship of a text data to a topic is probabilistic[1,5]. Supervised learning methods are machine learning methods to learn a classifier or understand a function from the training data to perform predictions on hidden data. There is a wide range of supervised methods for example probabilistic classifiers, nearest neighbour classifiers, rule-based classifiers and decision trees [6, 7].

Various probabilistic methods exist with unsupervised topic models such as probabilistic Latent semantic analysis (pLSA) [8] and Latent Dirichlet Allocation (LDA) [9], and conditional random fields of supervised learning [10] are used frequently in text mining.We propose a new method to handle queries and solutions of an internal messaging chat system and automatically provide better query results. The queries and solutions are uploaded by different employees of the organization. Every query and response is analyzed. Similar and related queries, as well as query results, are grouped using clustering method. Whenever a user enters a new query, it is analyzed, and relevant query results are displayed from the query group it belongs.

II. LITERATURE SURVEY

Clustering is a popular data mining algorithm and has been broadly studied in the context of text. It has an extensive variety of uses such as in classification [11, 12], visualization [13] and document organization [14].

Clustering is a task to find the groups of similar documents in a set of documents. The similarity is assessed by using a similarity function. Text clustering could be at different stages of granularities where clusters can be sentences, terms, documents or paragraphs. Clustering techniques are used for organizing documents to improve support browsing and retrieval, for example Cutting et al. [15][16]has used clustering to create a table of contents of a large set of documents Various algorithms are proposed to optimize text representation [17]. Text clustering algorithms are split into different types such as agglomerative clustering algorithms and partitioning algorithms. Clustering algorithms have various trade-offs regarding effectiveness and efficiency.

Hierarchical clustering algorithms get their name as they form a set of clusters. The hierarchy is constructed in bottom-up (called agglomerative) fashion or top-down (called divisive) fashion. Hierarchical clustering algorithms are a type of the Distance-based clustering algorithms, that is, a similarity function is used to determine the similarity among text documents. A general outline of the hierarchical clustering algorithms for text data is found in [18][19]. In the top-down approach, we start on with one cluster which includes the entire documents. We recursively divide this group into sub-clusters. In the agglomerative approach, every document is primarily considered as an individual cluster. Then in succession, similar clusters are combined until the entire documents are embraced in one cluster. Time complexity is high in hierarchal clustering which is the main drawback, in general, it's in the sort of O(n 2logn), n being the number of record points.

K-means clustering is the partitioning algorithm which is used in the data mining. The k-means clustering divides n documents in the framework of text data into k clusters, represents about the clusters on which they are built. The important form of k-means algorithm is: It is difficult to find a solution for k-means clustering (NP-hard). However, there are well-organized heuristics such as [20] that are employed in array to unite quickly to a local best possible. The disadvantage of k-means clustering is that it is susceptible to the primary choice of the number of k.

III. PROPOSED METHODOLOGY

ARCHITECTURE:

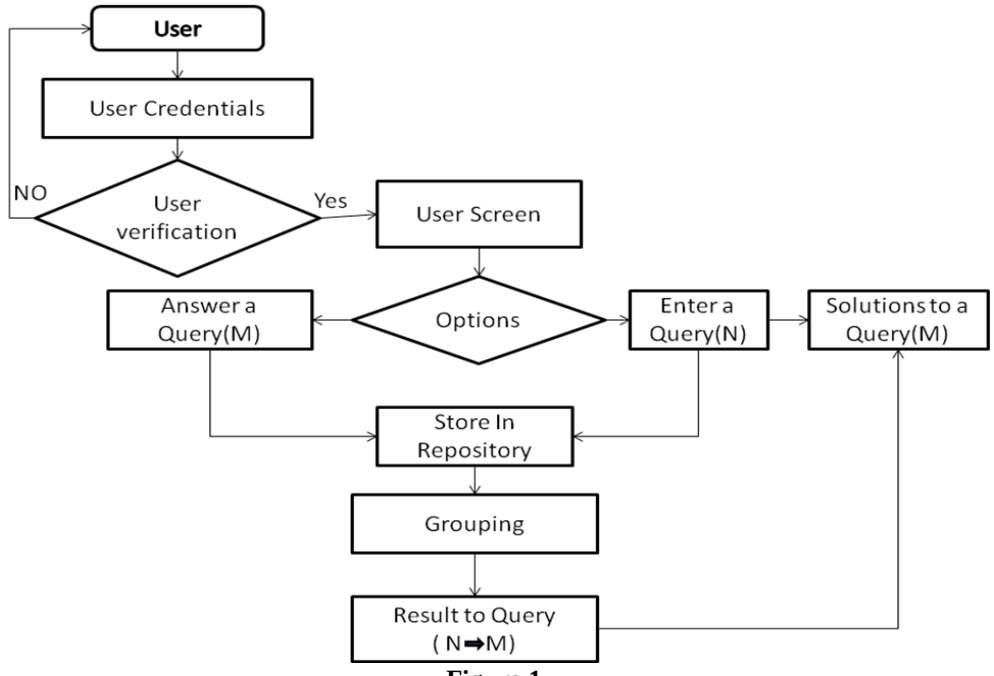

Figure 1

The proposed system architecture is discussed for optimization and organizing user search query and responses. It gives a detailed description of working functionality of the system. The algorithm operates as follows. Initially, user logins with user credentials. If credentials are valid, it goes to the chat screen if not it reverts to user login page. In the chat screen, the user can search a query, which is denoted by N or answer a query, which is denoted by M. whenever a user enters a search query, it is stored and analyzed. Then it is added into a relevant query group. Result's M corresponding to query N in the same group is provided to the user as a response. Whenever a user answers a query, it is analyzed and stored in the corresponding query group.

We consider a ten-minute chat, where many users have posted queries and solutions. Query 1 has four solutions (S1_1, S1_2, S1_3, S1_4), Query 2 has two solutions (S2_1, S2_2), Query 3 has three solutions (S1_1, S1_2, S1_3), Query 4 has one solution (S4_1), Query 5 has one solution (S5_1), Query 6 has two solutions (S6_1,

S6_2), Query 7, 8 have one solution each (S7_1), (S8_1) respectively and Query 9 has no solution yet received from the users.

Assuming queries (1,5,7) are related, (2,4) are related, query 3 is single, and query(6,8,9) are related. Working of the below is discussed in the experimental setup.

10:50:00	Query 1	10:54:27	Solution 3_3
10:50:21	Solution 1_1	10:54:52	Query 5
10:50:55	Query 2	10:55:17	Solution 4_1
10:51:11	Solution 2_1	10:55:48	Solution 5_1
10:51:42	Solution 1_2	10:56:32	Query 6
10:51:59	Solution 2_2	10:56:57	Solution 6_1
10:52:15	Query 3	10:57:29	Query 7
10:52:36	Solution 3_1	10:57:47	Solution 6_2
10:52:54	Solution 1_3	10:58:33	Solution 7_1
10:53:20	Query 4	10:59:02	Query 8
10:53:44	Solution 3_2	10:59:34	Solution 8_1
10:54:05	Solution 1_4	11:00:00	Query 9

Figure 2

After Query Grouping:

Group 1
Query 1
Solution 1_1
Solution 1_2
Solution 1_3
Solution 1_4
Query 5
Solution 5_1
Query 7
Solution 7_1

Group 2
Query 2
Solution 2_1
Solution 2_2
Query 4
Solution 4_1

Group 3
Query 3
Solution 3_1
Solution 3_2
Solution 3_3

Group 4
Query 6
Solution 6_1
Solution 6_2
Query 8
Solution 8_1
Query 9

Figure 3

IV. EXPERIMENTAL SETUP

The proposed work aims to group the queries and solutions in a chat messaging system of an organization, where the employees use internal messaging (chat system) to resolve some technical issues when they face. Queries are posted anonymously on the chat messenger. Answers are also provided by other employees.

1. User chat screen: :

After user logged into the system, a chat screen appears, where the user can either enter a query or a response. We used a database to store all the information. There are two significant tables, one table stores user credentials and another table stores queries/responses which are maintained by the server of the organization, Whenever a user enters a query or a response, it is immediately stored into a table with following attributes namely query/response, user id, time, date.

2. Analyse and group similar queries/responses:

After the above step, the query or response is analyzed and put into the same query group. If there is no query group existing, a new query group is formed.

3. Provide automatic query results to queries:

As users enter lots of queries and responses, our algorithm groups these queries /responses. After enough groups with sufficient queries and responses are formed, the chat messaging system can be made automatic, i.e., whenever a user enters a query, the system analyses the query with already existing groups, if the query is related to a group then responses nearly matching the query in that particular group are provided as responses.

Query Group [21]:
A query group is an ordered list of queries, q_i, collectively with related set of responses r_i of q_i. A query group is represented as $g=(\{q_1, r_{1\_1}, r_{1\_2\ldots}\},\{ q_2, r_{2\_1}, r_{2\_2\ldots}\},\ldots.,\{ q_n, r_{n\_1}, r_{n\_2\ldots}\})[]$.
An approach to the identification of query groups is to first consider every query or response as a singleton group, and then iteratively merge these singleton groups (in a k-means or agglomerative way [8]). Given g_c, we check if there are existing query groups relevant to g_c. If so, we merge g_c with the query group g having the highest similarity t_{max} above or equal to the threshold t_{sim}. Otherwise, we put g_c as a new singleton query group and insert it into G.
Algorithm:
Selecting the query group that is the most similar to the given query or response.
Input:
The current singleton query group g_c containing the current query q_c or responses r_c.
2) An existing query groups $g = \{g_1, \ldots, g_m\}$
3) A similarity threshold t_{sim}, $0 \leq t_{sim} \leq 1$

Output:
The query group G that best suit g_c, or a new one if needed
a. G =0
b. $t_{max} = t_{sim}$
c. **for** i = 1 **to** m
d. **if** $sim(g_c, g_i) > t_{max}$
e. $g = g_i$
f. $t_{max} = sim(g_c, g_i)$
g. **if** g =Φ
h. $G = G \cup g_c$
i. $g = g_c$
j. **return** g

Query Relevance or Query Group Relevance:
To group only the related queries or responses, we need a relevance measure simm between current query group g_c and already present query group g_i € G.
In our algorithm we consider the combination of time and text similarity to group related queries and responses.
Time:
One may imagine that g_c or r_c and g_i are relevant if the queries or responses of g_c appear close to g_i in time. In other words, we assume that related queries or responses are generally issued within a short period of time. A time-based relevance metric $simm_{time}$ is defined so that, it can be used instead of simm to group related responses or queries.

$Sim_{time}(g_c, g_i)$ is the inverse of the time difference between the times that q_c/r_c and q_i/r_i are issued:

$$Simmtime(gc, gi) = \frac{1}{|time(qc) - time(qi)|} \text{ or } \frac{1}{|time(rc) - time(qi)|} \text{ or } \frac{1}{|time(rc) - time(ri)|}$$

Higher $simm_{time}$ values state that queries or responses are temporally close.

Text:
We can also assume that two groups are related, if their queries or responses are textually similar . Textual similarity between sets of words can be measured using metrics such as the fraction of overlapping words (Jaccard similarity [10]). We can use text similarity relevance metrics instead of simm.

$$Simm\_jac(gc, gi) = \frac{|words(gc) \cap words(gi)|}{|words(gc) \cup words(gi)|}$$

Using the above algorithm, related queries and responses can be placed into a query group. After enough groups are formed, whenever user enters a new query, it is mapped to a relevant group and responses of that particular group can be automatically provided as results.

From Figure 1, 2, query 1 be "what does error 404 mean?" and the solution 1_1 is "The HTTP 404 Not Found Error implies that the webpage you were trying to load could not be found on the server" , after sometime the user might ask another query and let it be query 5 " How to Fix the 404 Not Found Error?" and another user posts the solution 5_1 as "Clear your browser's cache because it may indicate that the 404 Not Found error message might be just yours" From the above conversation it is clear that the queries 1, 5 and solutions 1_1, 5_1 are related, grouping is done based on time and text similarity.

V. CONCLUSIONS

In this paper we proposed a new method to automatically answer a query using query grouping techniques in a chat messaging system. Initially, we grouped all related queries and responses then using these groups, a new query is analyzed and mapped to a existing group and responses of that particular group are provided as answers. Our algorithm works for small organizations where the number of queries and responses are limited to only a few thousands, to adapt this to larger organizations, the algorithm may be supported by other relevant measures as applicable.

REFERENCES

[1]. Ronen Feldman and Ido Dagan. 1995. Knowledge Discovery in Textual Databases (KDT).. In KDD, Vol. 95. 112–117.

[2]. Ming-Syan Chen, Jiawei Han, and Philip S. Yu. 1996. Data mining: an overview from a database perspective. IEEE Transactions on Knowledge and data Engineering 8, 6 (1996), 866–883.y: Springer, 1989, vol. 61.

[3]. Jim Cowie and Wendy Lehnert. 1996. Information extraction. Commun. ACM 39,1 (1996), 80–91

[4]. M. Allahyari, S. Pouriyeh, M. Assefi, S. Safaei, E. D. Trippe, J. B. Gutierrez, and K. Kochut. 2017. A Brief Survey of Text Mining: Classification, Clustering and Extraction Techniques. ArXiv e-prints (2017). arXiv: 1707.02919

[5]. Mark Steyvers and Tom Griffiths. 2007. Probabilistic topic models. Handbook of latent semantic analysis 427, 7 (2007), 424–440

[6]. Tom M Mitchell. 1997. Machine learning. 1997. Burr Ridge, IL: McGraw Hill 45 (1997)

[7]. Fabrizio Sebastiani. 2002. Machine learning in automated text categorization. ACM computing surveys (CSUR) 34, 1 (2002), 1–47.

[8]. Thomas Hofmann. 1999. Probabilistic latent semantic indexing. In Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 50–57

[9]. DavidMBlei, AndrewY Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. the Journal of machine Learning research 3 (2003), 993–1022

[10]. John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.(2001)

[11]. L Douglas Baker and Andrew Kachites McCallum. 1998. Distributional clustering of words for text classification. In Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 96–103.

[12]. Ron Bekkerman, Ran El-Yaniv, Naftali Tishby, and Yoad Winter. 2001. On feature distributional clustering for text categorization. In Proceedings the 24th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 146–153.

[13]. Igor Cadez, David Heckerman, Christopher Meek, Padhraic Smyth, and Steven White. 2003. Model-based clustering and visualization of navigation patterns on a web site. Data Mining and Knowledge Discovery 7, 4 (2003), 399–424.

[14]. Douglass R Cutting, David R Karger, Jan O Pedersen, and John W Tukey. 1992. Scatter/gather: A cluster-based approach to browsing large document collections. In Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 318–329

[15]. Douglass R Cutting, David R Karger, and Jan O Pedersen. 1993. Constant interaction-time scatter/gather browsing of very large document collections. In Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 126–134

[16]. Peter G Anick and Shivakumar Vaithyanathan. 1997. Exploiting clustering and phrases for context-based information retrieval. In ACM SIGIR Forum Vol. 31. ACM, 314–323

[17]. Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. Information processing & management 24, 5 (1988), 513–523

[18]. Fzionn Murtagh. 1983. A survey of recent advances in hierarchical clustering algorithms. Comput. J. 26, 4 (1983), 354–359
[19]. Fionn Murtagh. 1984. Complexities of hierarchic clustering algorithms: State of the art. Computational Statistics Quarterly 1, 2 (1984), 101–113
[20]. Paul S Bradley and Usama M Fayyad. 1998. Refining Initial Points for K-Means Clustering.. In ICML, Vol. 98. Citeseer, 91–99
[21]. Heasoo Hwang, Lauw, H., Getoor, L. and Ntoulas, A. (2012). Organizing User Search Histories. IEEE Transactions on Knowledge and Data Engineering, 24(5), pp.912-925.

*Sunil Bhutada. "Automated Chat Response and Query Grouping using Data Mining Techniques." International Journal Of Engineering Research And Development , vol. 13, no. 11, 2017, pp. 56–61.