Dynamic Requirement Clustering of Requirement with Usable Test Cases by Cosine-Correlation

Amit Verma, PhD
Professor and Head,
Computer Science Department, Chandigarh University

Chetna
Research Scholar
Computer Science Department, Chandigarh University
Gharuan, Kharar, India

ABSTRACT
In software engineering testing plays an important role in development and maintenance of software. Component based software development gained a lot of practical importance in the field of Software engineering by the academic researcher and industry for finding reusable efficient test cases. It is the predominant problem in software engineering that clustering reduces the search space of the component of test cases by grouping of similar entities together ensuring reduce time complexity and reduce the search time for retrieve test cases according to requirement. In this research paper we investigate how k-mean work on the set of requirement and usable test cases we also define how to resolve the k-mean clustering static number of cluster when new requirement or test cases will come. In this research paper we investigate how k-mean work on the set of requirement and usable test cases we also define how to resolve the k-mean clustering static number of cluster when new requirement or test cases will come. Here we purposed an approach for dynamic clustering for test cases and requirement.

Keywords
Clustering, correlation, retrieval, K-mean

1. INTRODUCTION
Collections of large amount of data worldwide lead to the term Big data, and with the advent to extract meaningful information from this data leads to existence of data mining. Data can be text of words, conjunctions of sentences in documents. Text mining is one of the current research topic in NLP . Text mining is a process of extracting high quality information from the text. Text mining has many applications but its major applications are in software engineering which includes Feature extraction , code retrieval , traceability , bug locating , reverse engineering , refactoring and restructuring, reusability, sequencing .These tasks are based on the query formulated by the user. Text mining further defines text retrieval which means extracting or retrieving text by formulating or reformulating the query. Query states the feature and retrieval depends upon the text in the query and the related text in the database or reprisotry.

Text retrieval linked with the reusable test cases as a automated software for component retrieval is an important thing. In this paper we are purposing a algorithm for clustering the reusable test cases with the set of requirements given by the user in the form of query and then finding the correlation between the clusters. Clustering groups the similar features or properties in one cluster and dissimilar features in another cluster, by clustering we have quick search response time and easy retrieval.

2. RELATED WORK
There are many research studies that are related with text mining. [1] A generalized approach is used to cluster documents or components on the basis of similarity function i.e. hybrid XOR, which is deflagrate for finding the level of resemblance between documents, software components and modules. [3] evidence that LDA-GA is capable to name robust LDA form that extend to a mellower exactitude on various datasets for the described software engineering tasks as equated to antecedently released outcomes, heuristic and the outcome of a combinative search.[5] recommender (Refoqus) is measured emulously opposed to 4 baseline approaches that are applied in document retrieval. Recommender executed the baselines and its recommendation extend to query execution advancement and conservation in 84% of the cases on an average[6] Simulating the semantic resemblance between documents confront a substantial theorisation challenge for cognitive science; with clichéd applications deals with the text in information handling, retrieving and decision support systems. [7] A prelude empiric study demonstrate that the metric is a good cause forecaster for text retrieval concept position, exceeding previously existing techniques from the area of NLP.[8] the relative reveals that a simple text models i.e. UM and VSM are more effectual at correctly recalling the related files within a library as equated to the more convoluted models such as LDA.[10] Revealed that the retrieval efficiency of document can be importantly more efficacious than conventional ranked list approach.[11] This paper presented the technique to originate clusters and cluster delineation by using user point of view.[12] Hierarchal clustering applied to search outcomes, possibly increases the recovery effectiveness likened to other static clustering of traditional IFS.

3. METHODOLOGY
Our system consists of following steps:

a. Preprocessing
b. Clustering with Cosine similarity
c. Ranking the clusters
d. Sentence similarity index.
e. Cluster similarity index.
f. Normalize cluster similarity index.

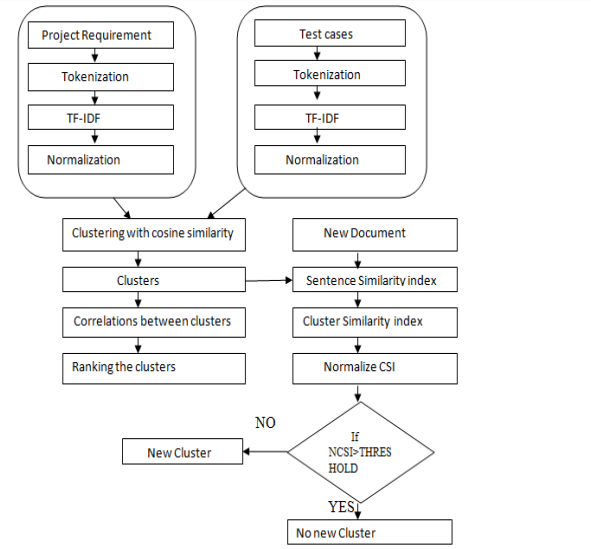

Fig 3.1: Diagram of the System Module

1. In the first step we will extract the text from the Requirement document and test document.
2. After extracting the text, pre-processing phase will start in which tokenization and stop world removal will be applied on the text.
3. In the third step, after pre-processing phase we will calculate the TD-IDF phase in which document, weightage will given to texts and represent in the vector form.
4. After completion of these steps, K-Mean with cosine similarity is applied this will result in cluster of documents and test cases.

   Above step results in cluster, in this step we find less duplicate information cluster by correlation.
5. In the last step we will check the similarity of new document by calculating the sentence similarity index and cluster similarity index, if the value of NCSI(Normalize cluster similarity index) is greater than threshold then new document will not make new cluster otherwise vice versa.
6. We will calculate the accuracy, precision, recall and f-score of K-Mean with cosine similarity.

ALGORITHM 1

For I=0 to Length (doc)

{

Tokenization;

Stop word Removal;

TF-IDF vector space;

}

For I=0 to Length(doc) Feature set

{

K-mean with cosine similarity

}

K-number of cluster with document (d1, d2, d3 .....Dn)

ALGORITHM 2

Input: K-number of cluster with document and new requirement.

Output: Cluster new requirement.

$$s1 \dots sn \begin{bmatrix} sim(d1, si) & \cdots & sim(s1, dn) \\ \vdots & \ddots & \vdots \\ sim(sn, d1) & \cdots & sim(sn.dn) \end{bmatrix}$$

For I=0 to Length(Sentence) in new requirement.

$$SSI = \sum_{i=0}^{i=no.\ of\ lines\ in\ a\ document} SiDi$$

$$CSI = \sum_{i=0}^{i=no.\ of\ document\ in\ cluster} SSIi$$

NCSI= CSI/ Number of total document.

Cluster new requirement=max(NCSI1, NCSI2……..NCSIn)

}

Cluster with max NCSI

If NCSI<Threshold.

Make new cluster

}

4. EXPERIMENTAL RESULTS

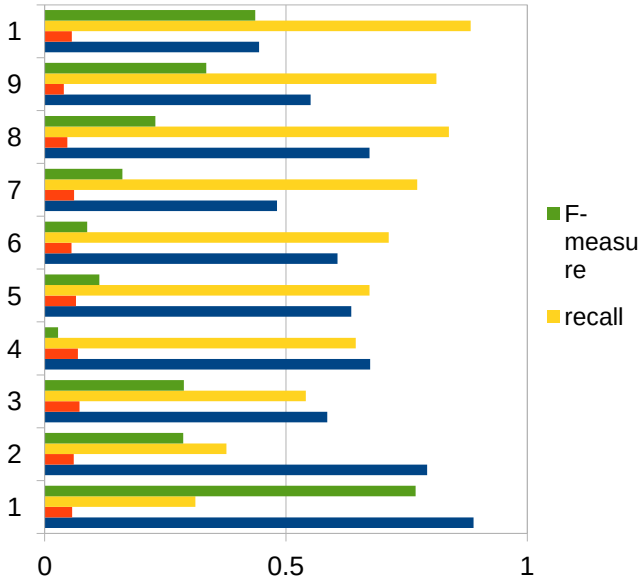

Figure 5.23 Mixed graph Showing accuracy, precision, recall, f-measure at various cluster size.

Cluster	Precision	Accuracy	Recall	F-measure
Two	0.05652	0.8889	0.311	0.768
Three	0.06038	0.79253	0.3762	0.2868
Four	0.07222	0.58587	0.3762	0.2884
Five	0.06862	0.67431	0.6446	0.02789
Six	0.06458	0.63535	0.6728	0.11296
Seven	0.05523	0.606808	0.7129	0.0882

Eight	0.06057	0.4815	0.7723	0.1606
Nine	0.04660	0.6732	0.8376	0.2296
Ten	0.03974	0.5513	0.8116	0.3347
Eleven	0.05641	0.4440	0.8827	0.4365

Table 4.1: Varying Accuracy, precision, recall, f-measure with varying cluster size.

Experimental results shows the above stated four parameters i.e. Accuracy, Precision, Recall, F-measure for various cluster sizes. These results are more efficient than the earlier clustering methods and in this clustering is dynamic which further results in more efficient clustering.

5. CONCLUSION AND FUTURE SCOPE

It is concluded that cluster formed by K-mean by cosine similarity gives a domain knowledge of text with help of base similarity of text but this work does not give ranking of unique information contained in cluster because while in clustering , clusters created contains duplicity of information and for that we used correlation and find highly correlated cluster . All the work done above doesn't overcome the problem of static number of cluster in k-mean and for that we proposed algorithm which may or may not create new cluster it works by finding the cosine similarity between the document and previously formed cluster, by finding the sentence similarity index and cluster similarity index in which every line of documents is checked against every line of clusters, if the value of greater then threshold then new cluster is not formed otherwise vice versa. Enhance the work by dynamic counting the threshold by using Exception-Maximization algorithm and use adaptive method for threshold optimization. One can enhance the work by using other dataset and validate our approach.

6. REFERENCES

[1] Radhakrishna, Vangipuram, Chintakindi Srinivas, and CV Guru Rao. "Document Clustering Using Hybrid XOR Similarity Function for Efficient Software Component Reuse." *Procedia Computer Science* 17 (2013): 121-128.

[2] Milios, E., et al. "Automatic term extraction and document similarity in special text corpora." *Proceedings of the sixth conference of the pacific association for computational linguistics*. 2003.

[3] Panichella, Annibale, et al. "How to effectively use topic models for software engineering tasks? an approach based on genetic algorithms." *Proceedings of the 2013 International Conference on Software Engineering*. IEEE Press, 2013

[4] Wang, James Z., and William Taylor. "Concept forest: A new ontology-assisted text document similarity measurement method." *Web Intelligence, IEEE/WIC/ACM International Conference on*. IEEE, 2007.

[5] Haiduc, Sonia, et al. "Automatic query reformulations for text retrieval in software engineering." *Proceedings of the 2013 International Conference on Software Engineering*. IEEE Press, 2013.

[6] Lee, M., Brandon Pincombe, and Matthew Welsh. "An empirical evaluation of models of text document similarity." *Cognitive Science* (2005).

[7] Haiduc, Sonia, et al. "Evaluating the specificity of text retrieval queries to support software engineering tasks." *Software Engineering (ICSE), 2012 34th International Conference on*. IEEE, 2012.

[8] Rao, Shivani, and Avinash Kak. "Retrieval from software libraries for bug localization: a comparative study of generic and composite text models."*Proceedings of the 8th Working Conference on Mining Software Repositories*. ACM, 2011.

[9] Bouras, Christos, and Vassilis Tsogkas. "A clustering technique for news articles using WordNet." *Knowledge-Based Systems* 36 (2012): 115-128.

[10] Leuski, Anton. "Evaluating document clustering for interactive information retrieval." *Proceedings of the tenth international conference on Information and knowledge management*. ACM, 2001.

[11] Bhatia, Sanjiv K., and Jitender S. Deogun. "Conceptual clustering in information retrieval." *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on* 28.3 (1998): 427-436.

[12] Tombros, Anastasios, Robert Villa, and Cornelis J. Van Rijsbergen. "The effectiveness of query-specific hierarchic clustering in information retrieval."*Information processing & management* 38.4 (2002): 559-582.

[13] Poshyvanyk, Denys, Malcom Gethers, and Andrian Marcus. "Concept location using formal concept analysis and information retrieval." *ACM Transactions on Software Engineering and Methodology (TOSEM)* 21.4 (2012): 23.

[14] Maitah, Wafa, Mamoun Al-Rababaa, and Ghasan Kannan. "Improving the Effectiveness of Information Retrieval System Using Adaptive Genetic Algorithm." *International Journal of Computer Science & Information Technology* 5.5 (2013): 91-105.

[15] Kettinger, William J., et al. "Strategic information systems revisited: a study in sustainability and performance." *MIS quarterly* (1994): 31-58.

[16] Dharmarajan, A., and T. Velmurugan. "Research Scholar, Research & Development Centre, Bharathiar University, Coimbatore-046, India."*Computational Intelligence and Computing Research (ICCIC), 2013 IEEE International Conference on*. IEEE, 2013.

[17] Lew, Michael S., et al. "Content-based multimedia information retrieval: State of the art and challenges." *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 2.1 (2006): 1-19.

[18] Eriksen, Hallvard Andreas. "Requirements to ultrasound imaging workstations in a Chinese hospital.