Analysis of Different Similarity Measures for Text Document Comparison

R. Anushya[1] Dr. Antony Selvadoss Thanamani[2]
[1]Research Scholar [2]Associate Professor
[1,2]Department of Computer Science
[1,2]NGM College, Pollachi, India

Abstract— Present days people are related with substantial measure of information on standard premise. The sole reason for produced information is to meet the quick needs and no endeavor in sorting out the information for later proficient recovery. Data mining is an idea of separating learning from such a gigantic measure of information. While a few grouping techniques and the related similitude measures have been proposed before, there is no deliberate near investigation of the effect of closeness measures on bunch quality. This might be on the grounds that the famous cost criteria don't promptly decipher crosswise over subjectively extraordinary measures. In this paper we look at four well known likeness measures (Euclidean, cosine, Pearson relationship and broadened Jaccard) related to various kinds of vector space portrayal (Boolean, term recurrence and term recurrence and reverse report recurrence) of archives. Grouping of reports is performed utilizing summed up k-Means; a Partitioned put together bunching strategy with respect to high dimensional inadequate information speaking to content records. Execution is estimated against a human-forced grouping of Topic and Place classifications. We led various analyses and utilized entropy measure to guarantee factual essentialness of results. Cosine, Pearson relationship and broadened Jaccard similitudes rise as the best measures to catch human classification conduct, while Euclidean measures perform poor.

Key words: Document Similarity, Jaccard Similarity, Cosine Similarity, Euclidean Distance, Pearson Coefficient

I. INTRODUCTION

With regularly expanding volume of content archives, the inexhaustible writings streaming over the Internet, enormous accumulations of reports in computerized libraries and storehouses, and digitized individual data, for example, blog articles and messages are heaping up rapidly consistently [1]. For content archives, grouping has turned out to be a powerful methodology and an intriguing exploration issue. Grouping of content archives assumes a crucial job in effective Document Organization, Summarization, Topic Extraction and Information Retrieval [2]. At first utilized for enhancing the accuracy or review in an Information Retrieval System, all the more as of late, bunching has been proposed for use in perusing a gathering of records or in arranging the outcomes returned by a web crawler because of client's question or help clients rapidly recognize and center around the important arrangement of results. Client remarks are bunched in numerous online stores, such asAmazon.com to give collective suggestions. In cooperative bookmarking or labeling, bunches of clients that share certain characteristics are recognized by their comments. Report grouping has additionally been utilized to consequently produce Hierarchical bunches of records [3].

The advancement of PC innovation in the couple of decades has prompted huge supplies of amazing and moderate PCs. Increment in the expansive electronic documentation it is difficult to picture these archives productively by putting manual exertion. These have brought difficulties for the productive and successful association of website page archives consequently [4]. Removing highlights from website pages is first errand found in mining. Based on removed highlights comparability between site pages will be measure. There is different comparability allots are pointed for work. Data mining is the method of mining the already obscure and conceivably valuable data from information. Archive bunching arranges reports into various groups. The reports in each bunch share some regular properties as indicated by similitude measure. Content bunching calculations assume a vital job in helping clients to successfully sort out, explore and condense the data. Because of unstable development of getting to data from the web, productive access of data is required fundamentally [5]. The Text preparing assumes an imperative job in data recovery, web inquiry and data mining.

In text mining, an archive is spoken to as a vector in which every part demonstrates the estimation of its comparing highlight in the report. The component esteem can be the quantity of events of a term showing up in the report (term recurrence), the proportion between the term recurrence and the aggregate number of events of the considerable number of terms in the record set (relative term recurrence), or a mix of term recurrence and converse archive recurrence (TFIDF) [6] . Normally, the greater part of the element esteems in the vector are zero, such high dimensionality and sparsity can be a noteworthy test for similitude measure which is an essential activity in content preparing calculations.

There are a few comparability measures proposed by various scientists for the undertaking of report grouping. The most fundamental one is Euclidean measure that utilizes the separation metric to process the comparability of the archives. Another generally utilized likeness measure is cosine comparability measure, it utilizes the archives that are spoken to in vector space and it computes the point between the directional vectors of these records [7]. These two measures don't consider the semantic of the terms (words) in the record. So as to compute the semantic similitude, scientists proposed Word Net based semantic likeness; they first concentrate the semantic words from reports and construct the semantic class progressive system of the terms to ascertain closeness dependent on shared ideas chain of command.

The likeness measurements between reports can be characterized in a few different ways relying upon the portrayal of the records; if the archives are spoken to as vectors where every component is a word at that point approaches dependent on Simple coordinating coefficient can

be utilized. This dataset contained all reports made by a whole class bunch amid one semester. This implies in the greater part of the cases an educator needs to peruse and recollect the substance of every one of their understudies work [8]. Despite the fact that, contingent upon the situation the person in question may just need to know the principle substance of the records for class exercises, for example, doing class bunches dependent on the comparability of the reports. Be that as it may, the need of looking through the structures to envision, think about or remark the substance it is additionally critical in this social setting.

II. DOCUMENT SIMILARITY MEASURES

Comparability is a mind boggling idea which has been generally talked about in the phonetic, philosophical and data hypothesis networks. Likeness has been a subject of extraordinary enthusiasm for mankind's history since quite a while back. Indeed, even before PCs were made, people have been keen on discovering closeness in all things. Every single field of study gives their own meaning of what likeness is. In Psychology comparability "… alludes to the mental closeness or nearness of two mental portrayals." while in music it's "… a specific likeness between at least two melodic pieces" and in geometry "Two geometrical items are called comparable on the off chance that they both have a similar shape." Definitions for similitude are distinctive for each field yet what props up in every one of them is the utilization of one major field to demonstrate the similitude:

Math is utilized to compute likeness where it overwhelms the field. After the beginning of two new fields in a century ago, Information Theory and Computer Science, the theme of comparability has not turned out to be littler at all [9] . Rather by utilizing the PC it has been less demanding to discover how comparable at least two things are to one another. This possibility caused by these fields where arithmetic can be connected and the effectiveness to make the counts quick, have influenced people to imagine calculations to make better approaches to figure similitude less demanding, quicker and as right as could be allowed. The instinctive ideas of closeness ought to be about the equivalent for the most part everybody.

The natural ideas of comparability ought to be about the equivalent. One of the predetermined classifications is literary comparability; taking at least two strings and contrasting them with one another with discover how comparative they are. This class is so intriguing and loaded up with advantage that numerous college educators and understudies have considered far into this classification and composed numerous a paper on the printed likeness [10]. Explanation behind this is people are extraordinary; they got distinctive thoughts and diverse limits with regards to how comparative something depends on a scale.

The on-request administration of cloud prompts the requirement for new booking systems. The new booking techniques to be proposed should join the customary planning ideas with new planning parameters, for example, transmission capacity, vitality utilization, work movement and cost for effective planning. Coming up next are where new methodologies can be proposed utilizing different improvement procedures, machine learning strategies or fluffy frameworks.

A. Vector Space Model

Likewise called term vector demonstrate is a math show for speaking to content records as vectors of identifiers, similar to terms or tokens. Obviously, the term relies upon what is being thought about yet are typically single words, watch words, expressions or sentences. A report gathering comprising of Y records, ordered by Z terms can be appeared in a Y x Z network M [11]. Accordingly, the inquiries and the archives are speaking to as vectors and each measurement compares to a different term subsequently every component in the lattice M is a load for the term Z in the report Y. In the event that the term shows up in the archive, the incentive in the framework for the explicit component changes, generally not. Utilizing this together with the suspicion of record similitude's hypothesis, the likeness between two archives can be determined by looking at the contrast between the holy messengers of each report vector and the question vector. This computation winds up given an outcome extending from 0 to 1, the two numbers notwithstanding. On the off chance that the record and the question vector are symmetrical the outcome is 0 and there will be no match otherwise known as the inquiry term does not exist in the archive. On the off chance that the outcome is 1 it implies that both the vectors are equivalent to one another. To figure these qualities numerous ways have been discovered, some more known than others.

B. Cosine Similarity

The standard method for evaluating the likeness between two archives x_1 and x_2 is to PC the cosine similitude of their vector portrayals and measure the cosine of the edge between the vectors

$$CosineSimilarity(x_1 + x_2) = \frac{V(x_1) \cdot V(x_2)}{|V(x_1)||V(x_2)|} \qquad (1)$$

While the denominator is the product of the vectors', $V(x_1)$ and $V(x_2)$, Euclidean lengths, the numerator shows the dot product which is the inner product of the vectors. The effect of the denominator is to length-normalize the vectors $V(x_1)$ and $V(x_2)$ to unit vectors:$v(x_1) = \frac{V(x_1)}{|V(x_1)|}$ and $V(x_2) = \frac{V(x_2)}{|V(x_2)|}$The result of the angle will show the result. If the angle is 0 between the document vectors then the cosine function is 1 and both documents are the same. If the angel is any other value then the cosine function will be less than 1 [12]. Does the angle reach -1 then the documents are completely different? Thus, this way by calculating the cosine angle between the vectors of x_1 and x_2 decides if the vectors are pointing in the same direction or not.

C. Term Frequency-Inverse Document Frequency

The term frequency– opposite record recurrence or TF-IDF weight is an approach to give words that show up in content a load. Thusly a few words ends up heavier than different words and influence the similitude score to be progressively exact in other words it makes a few words increasingly imperative that different words in the content that are being thought about [13]. To comprehend TF-IDF it is best to part up and takes one piece at any given moment; TF and IDF. TF

or term recurrence, as it says the recurrence a term shows up in a given record. Generally the term is standardized so an inclination towards longer reports, which would have a higher term recurrence for the explicit terms, to get a proportion of the imperative term t inside the record d. There are numerous sorts of term recurrence variations like sub straight TF scaling or greatest TF standardization yet this paper will work with the essential and most known one. The term recurrence would then be able to be characterized as:

$$TF_{i,j} = \frac{n_{i,j}}{\sum_k n_{i,j}} \quad (2)$$

$n_{i,j}$ is recurrence of the term d in the archive d. The dominator is the whole of the recurrence of the considerable number of terms showing up in the record d meaning the span of the archive d, otherwise known as |d|. There is a major issue with just having TF alone since all terms are viewed as similarly critical with regards to their pertinence on an inquiry. A few terms will have no or almost no segregating force in deciding importance [14]. For instance, if the reports are on winged creatures, the records will have the term flying creature commonly in the archives. This will underline reports which happen to utilize "fowl" all the more much of the time, without looking on the heaviness of progressively imperative terms. In the event that the imperative terms are "yellow" "winged creature" "yellow" will be not get any overwhelming weight on the off chance that it happens once in a while yet it will at present be a decent term to separate the pertinent archives from the non-important. That is the reason there is required an instrument for weakening the impact of terms that happen over and over again in the accumulation of archives to mean for assurance of importance.

That is the purpose behind the backwards report recurrence factor to be added to the TF condition. It will reduce the heaviness of terms that happens to happens too often in the reports and builds the heaviness of terms that just happens once in a while.

$$IDF_i = log\frac{|D|}{1 + |\{j: t_i \in d_i\}|} \quad (3)$$

|D| is the number of documents in the document set. $|\{j: t_i \in d_i\}|$ is the number of documents where the term " appears. Though if the term " is not found in the document, it will equal in a zero division thus it is common to add 1. So the TF-IDF equation will be:

$$TF - IDF = TF_{i,j} * IDF_i = \frac{n_{i,j}}{\sum_k n_{i,j}} * log\frac{|D|}{1 + |\{j: t_i \in d_i\}|} \quad (4)$$

TF-IDF will filter out the frequent terms when it gets a high weight term. This happens when a term has a high frequency in a document but a low frequency in all the documents put together. The TF-IDF value will be greater than 0 if IDF is greater than 1.

D. Jaccard Similarity Coefficient

The Jaccard Similarity Coefficient calculates the similarity between sets and is defined as:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} \quad (5)$$

1) Simple Calculation

The size of intersection of A and B divided by the size of the union of A and B. Jaccard Distance which instead of similarity measures dissimilarity between can be found by subtracting Jaccard Similarity Coefficient from 1[15]:

$$JD(A,B) = 1 - J(A,B) or JD(A,B) = \frac{|A \cup B| - |A \cap B|}{|A \cup B|} \quad (6)$$

Tanimoto Coefficient is an "extended" Jaccard Similarity Coefficient and Cosine Similarity put together. Meaning by having Cosine Similarity yield Jaccard Similarity Coefficient, Tanimoto Coefficient can be represented:

$$T(A,B) = \frac{A \cdot B}{||A||^2 + ||B||^2 - A \cdot B} \quad (7)$$

This is in case of binary attributes which are a special case of discrete attributes that only have 2 values, normally 0 or 1. Tanimoto Coefficient runs from -1/3 to 1 unlike the Cosine Similarity that runs from -1 to 1.

E. Euclidean Distance

Euclidean separation otherwise known as L2 remove otherwise known as Euclidean standard is another likeness measure in the vector space demonstrates. Euclidean separation is common to the point that the discussing Distance is about dependably alluded to this separation [16]. This comparability measure separates from the other vector space show likeness measures by not made a decision from the point like the rest but rather than the direct Euclidean separation between the vector inputs. To streamline it on the off chance that there are 2, Euclidean separation computes the separation between those focuses rather than the course of the vectors like it is done in Cosine Similarity. Euclidean separation analyzes the foundation of square contrasts between the organized of the sets in the vectors x and y:

$$|x \to y| = \sqrt{\sum_{i=1}^{M} (x_i - y_i)^2} \quad (7)$$

F. Hamming Distance

The Hamming Distance takes two strings of equivalent length and ascertains the quantity of positions at the spots where the characters are unique. It computes minimal number of substitutions expected to make one string into another [17]. Whenever took a gander at a few precedents it turns out to be evident that Hamming just use substitution and that's it. Hamming is for the most part utilized in mistake remedying codes in the fields like media transmission, cryptography and coding hypothesis. Since it is every one of the a matter of discovering where the distinctions in various information are. Whenever taken media transmission then it is just an issue if the number is 0 or 1 and in this way effectively determined how extraordinary the two information bundles of equivalent lengths are.

III. PROPOSED APPROACH

A. Text Similarity with N-Grams

One of the basic methods for finding the likeness between two records depends on Term-Frequency (TF) highlight exhibit in genuine cases. In this method, just the quantity of shared words is tallied and a likeness proportion is created dependent on this counter esteem. Another procedure is called n-gram [18]. A basic n-gram is an adjacent succession of words. They for the most part give higher grouping

exactness in contrast with TF since words can pass on various semantics as per their specific situation (i.e. polysemy, equivalent word). Perusing self-assertive length input records. You have to store these records into two diverse connected records. Since the quantity of basic terms and 2-grams will be self-assertive; we additionally need to see the outcomes will be put away in another connected rundown.

An increasingly broad methodology is to shingle the archive (or make k-grams). This takes continuous words and gatherings them as a solitary question. A k-gram is a successive arrangement of k words. In this way, the arrangement of every one of the 1-grams is actually the pack of words demonstrate. An elective name to k-gram is a k-shingle; these mean a similar thing.

D1: I am Sam.
D2: Sam I am.
D3: I do not like green eggs and ham.
D4: I do not like them, Sam I am.
The (N = 1)-grams of D1∪D2∪D3∪D4 are: {[I], [am], [Sam], [do], [not], [like], [green], [eggs], [and], [ham], [them]}
The (k = 2)-grams of D1∪D2∪D3∪D4
1 are: {[I am], [am Sam], [Sam Sam], [Sam I], [am I], [I do], [do not], [not like], [like green], [green eggs], [eggs and], [and ham], [like them], [them Sam]}.
The set of k-grams of a document with n words is at most n-k. The takes space O (n) to store them all. If k is small, this is not a high overhead. Furthermore, the space goes down as items are repeated.

1) Character Level

We can also create k-grams at the character level. The (N = 3)-character grams of D1 ∪ D2 are: {[iam], [ams], [msa], [sam], [ami], [mia]}.
The (N = 4)-character grams of D1∪D2 are: {[iams], [amsa], [msam], [sams], [sami], [amia], [miam]}.
Jaccard With N-Grams
Consider two sets A = {0, 1, 2, 5, 6} and B = {0, 2, 3, 5, 7, 9}. How similar are A and B? The Jaccard similarity is defined

$$JS(A,B) = \frac{|A \cap B|}{|A \cup B|} \qquad (11)$$
$$= \frac{|\{0,2,5\}|}{|\{0,1,2,3,5,6,7,9\}|} = \frac{3}{8} = 0.375$$

More notations, given a set A, the cardinality of A denoted $|A|$ counts how many elements are in A. The intersection between two sets A and B is denoted $A \cap B$and reveals all items which are in both sets. The union between two sets A and B is denoted $A \cup B$and reveals all items which are in either set. Confirm that JS satisfies the properties of a similarity.

To fully generalize set similarities (at least those that are amenable to large scale techniques) we introduce a third set operation. The symmetric difference between two sets A and B is denoted$A \triangle B = (A \cup B) \backslash (A \cap B)$. Note that n is called set minus and A\B is all of the elements in A, except those also in B. Thus, the symmetric difference of A and B describes all elements in A or B, but not in both. Here consider the follow class of similarities. It uses$\overline{A \cup B} = [n] \backslash (A \cup B)$, where [n] is a superset that all sets A and B we consider a subsets from.

$$S_{x,y,z,z'}(A,B) = \frac{x|A \cap B| + y\overline{|A \cup B|} + z|A \Delta B|}{x|A \cap B| + y\overline{|A \cup B|} + z'|A \Delta B|} \qquad (12)$$

It can define several concrete instances.
Jaccard Similarity defined

$$JS(A,B) = S_{1,0,0,1}(A,B) = \frac{|A \cap B|}{|A \cap B| + |A \Delta B|} = \frac{|A \cap B|}{|A \cup B|} \qquad (13)$$

So how do we put this together? Consider the (k = 2)-grams for each D1, D2, D3, and D4:
D1: [I am], [am Sam]
D2: [Sam I], [I am]
D3: [I do], [do not], [not like], [like green], [green eggs], [eggs and], [and ham]
D4: [I do], [do not], [not like], [like them], [them Sam], [Sam I], [I am]
Now the Jaccard similarity is as follows:

JS (D1, D2)	=	1/3	≈	0.333
JS (D1, D3)	=	0	=	0.0
JS (D1, D4)	=	1/8	=	0.125
JS (D2, D3)	=	0	=	0.0
JS (D3, D4)	=	2/7	≈	0.286
JS (D3, D4)	=	3/11	≈	0.273

This is the special abstract structure of sets to compute this distance (approximately) very efficiently and at extremely large scale.

B. Euclidean Distance

Euclidean separation is a standard measurement for geometrical issues. It is the common separation between two and can be effectively estimated with a ruler in a few dimensional space. Euclidean separation is generally utilized in grouping issues, including bunching content. It fulfills all the over four conditions and subsequently is a genuine measurement. It is likewise the default remove measure utilized with the K-implies calculation.

Measuring distance between text documents, given two documents d_a and d_b represented by their term vectors $\overrightarrow{t_a}$ and $\overrightarrow{t_b}$ respectively, the Euclidean distance of the two documents is defined as

$$D_E(\overrightarrow{t_a}, \overrightarrow{t_b}) = \left(\sum_{t=1}^{m} |w_{t,a} - w_{t,b}|^2 \right)^{1/2} \qquad (14)$$

Where the term set is$T = \{t_1, \ldots . t_m\}$. As mentioned previously, it uses the $tfidf$ value as term weights, that is$w_{t,a} = tfidf(d_a, t)$.

C. Cosine Document Similarity

At the point when records are spoken to as term vectors, the similitude of two reports compares to the relationship between's the vectors. This is evaluated as the cosine of the point between vectors, that is, the supposed cosine similitude. Cosine similitude is a standout amongst the most mainstream comparability estimates connected to content records, for example, in various data recovery applications.
Given two documents $\overrightarrow{t_a}$ and$\overrightarrow{t_b}$, their cosine similarity is

$$SIM_C(\overrightarrow{t_a}, \overrightarrow{t_b}) = \frac{\overrightarrow{t_a} \cdot \overrightarrow{t_b}}{|\overrightarrow{t_a}| \times |\overrightarrow{t_b}|} \qquad (15)$$

Where $\overrightarrow{t_a}$ and $\overrightarrow{t_b}$are m-dimensional vectors over the term set$T = \{t_1, \ldots . t_m\}$. Each dimension represents a term with its weight in the document, which is non-negative. As a

result, the cosine similarity is non-negative and bounded between [0, 1].

An important property of the cosine similarity is its independence of document length. For example, combining two identical copies of a document d to get a new pseudo document d0, the cosine similarity between d and $d'$is 1, which means that these two documents are regarded to be identical. Meanwhile, given another document l, d and d' will have the same similarity value to l, that is,$sim(\overrightarrow{t_d}, \overrightarrow{t_l}) = sim(\overrightarrow{t_{d'}}, \overrightarrow{d_l})$. In other words, documents with the same composition but different totals will be treated identically. Strictly speaking, this does not satisfy the second condition of a metric, because after all the combination of two copies is a different object from the original document. However, in practice, when the term vectors are normalized to a unit length such as 1, and in this case the representation of d and d0 is the same.

IV. Conclusion

The measures have noteworthy impact on Partitional bunching of content reports aside from the Euclidean separation measurer. Pearson relationship coefficient is somewhat better as the subsequent grouping arrangements are progressively adjusted and is closer to the physically made classifications. The semantic closeness frames a focal segment in numerous NLP frameworks, from lexical semantics, to grammatical form labelling, to internet based life investigation. Late years have seen a recharged enthusiasm for growing new similitude procedures. The expanded intrigue has prompted many strategies for estimating semantic similitude. The propose framework utilize distinctive sort of record similitude check utilizing Cosine, Jaccard and Euclidean separation.

References

[1] J. A. Aslam and M. Frost, "An information-theoretic measure for document similarity," in Proc. 26th Annu. Int. ACM SIGIR Conf. Res. Dev. Inf. Retrieval, 2003, pp. 449–450.
[2] D. Cai, X. He, and J. Han, "Document clustering using locality preserving indexing," IEEE Trans. Knowl. Data Eng., vol. 17, no. 12, pp. 1624–1637, Dec. 2005.
[3] J. D'hondt, J. Vertommen, P.-A. Verhaegen, D. Cattrysse, and J. R. Duflou, "Pairwise-adaptive dissimilarity measure for document clustering," Inf. Sci., vol. 180, no. 12, pp. 2341–2358, 2010.
[4] C. G. Gonz´alez,W. Bonventi, Jr., and A. L. Vieira Rodrigues, "Density of closed balls in real-valued and autometrized boolean spaces for clustering applications," in Proc. Brazilian Symp. Artif. Intell., 2008, pp. 8–22.
[5] M. Li, X. Chen, X. Li, B. Ma, and P. M. B. Vit´anyi, "The similarity metric,"IEEE Trans. Inf. Theory, vol. 50, no. 12, pp. 3250–3264, Dec. 2004.
[6] Yung-Shen Lin, Jung-Yi Jiang, and Shie-Jue Lee, "A similarity measure for text classification and clustering,"IEEE Trans. Knowl. Data Eng., vol. 26, no. 7, pp. 1575–1590, Jul. 2014.
[7] L. Mazuel and N. Sabouret, "Semantic relatedness measure using object properties in an ontology," inProc. Int. Semantic Web Conf., 2008, pp. 681–694.
[8] T. Pedersen, S. V. S. Pakhomov, S. Patwardhan, and C. G. Chute, "Measures of semantic similarity and relatedness in the biomedical domain," J. Biomed. Inform., vol. 40, no. 3, pp. 288–299, 2007.
[9] S. C. Sahinalp, M. Tasan, J. Macker, and Z. M. Ozsoyoglu, "Distance based indexing for string proximity search," inProc. 19th Int. Conf. Data Eng., 2003, pp. 125–136.
[10] S. Tata and J. M. Patel, "Estimating the selectivity of tf-idf based cosine similarity predicates,"ACM Sigmod Rec., vol. 36, no. 2, pp. 7–12, 2007.
[11] J. Z. Wang, Z. Du, R. Payattakool, S. Y. Philip, and C.-F. Chen, "A new method to measure the semantic similarity of GO terms," Bioinformatics, vol. 23, no. 10, pp. 1274–1281, 2007.
[12] M.-L. Zhang and Z.-H. Zhou, "ML-KNN: A lazy learning approach to multi-label learning," Pattern Recognit., vol. 40, no. 7, pp. 2038–2048, 2007.
[13] Y. Zhao and G. Karypis, "Comparison of agglomerative and partitional document clustering algorithms," Defense Tech. Inf. Center Document, Fort Belvoir, VA, CA, Tech. Rep. TR-02-014, 2002.
[14] K. Shrivastava, N. D. Londhe, R. S. Sonawane, and J. S. Suri, "First review on psoriasis severity risk stratification: An engineering perspective," Comput. Biol. Med., pp. 52–63, vol. 63, 2015.
[15] Cardoso-Cachopo, "Improving methods for single-label text categorization," Ph.D. dissertation, Instituto Superior Tecnico, Universidade Tecnica de Lisboa, Lisboa, Portugal, 2007.
[16] R. Malarvizhi, Dr. Antony Selvadoss Thanamani, "K-Nearest Neighbor in Missing Data Imputation", International Journal of Engineering Research and Development, Volume 5 Issue 1-November-2012.
[17] Cluster Based Mean Imputation International Journal of Research and Reviews in Applicable Mathematics & Computer Science. Vol 2.No.1, 2012, Ms .R. Malarvizhi and Dr. Antony Selvadoss Thanamani
[18] K-NN Classifier Performs Better Than K-Means Clustering in Missing Value Imputation, International Journal for Research in Science & Advanced Technologies, Vol 1.Issue-2, 2013, Ms. R. Malarvizhi and Dr. Antony Selvadoss Thanamani.

