Journal of Theoretical and Applied Information Technology
15th December 2021. Vol.99. No 23

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

RECOMMENDATION OF A LIST OF ITEMS OF SEARCH RETRIEVAL FOR USER'S INTENT

SALMA GAOU[1] , MOURAD ELOUALI[2], KHALID AKHLIL[3], HICHAM TRIBAK[4]

Renewable Energies Team Acoustic and Mechanical Microsystems University Ibno ZOHR, Morocco
E-mail:[1]S.Gaou@uiz.ac.ma,[2]M.elouali@uiz.ac.ma, [3]K.Akhlil@uiz.ac.ma, [4]H.Tribak@uiz.ac.ma

ABSTRACT

Information retrieval systems aim to generate Search Engine Results Pages (SERP), which are web pages automatically generated by a search engine according to the keywords entered by the net surfers. The results are presented in a list where the most relevant data from the search engine are at the top. The main challenge about Information retrieval Systems is the gap between the intent of the Internet user and the appropriate keywords in their disposal. The emergence of such systems is motivated by the need of precise information and they may be different from Internet search engines like Google or Yahoo! WikiAnswers, Answers and domain-specific forums like Stack Overflow, on certain specific points. Although the idea of receiving a direct and targeted response to an issue seems very attractive and the quality of the question itself can have a significant effect on the likelihood of obtaining useful responses. Such an information retrieval paradigm is particularly appealing when the problem cannot be answered directly by the search engines due to the unavailability of relevant online content. A good understanding of the underlying purpose of an issue is important to better meet the information needed by the user.
In this paper, we propose a new approach to detect the user's intent. This approach is based on the method of the recommendation of a list of items but without calculation of prediction. The method lies on the co-dissimilarity and the tree covering minimum weight based on the theory of graphs. Our approach improves the ranking of a website in organic search results to increase visibility and quality.

Keywords: *Search engine optimisation (SEO), intent User, Information search, Ranking of search results, search retrieval*

1. INTRODUCTION

With the bifurcation of the web from a predominantly vertical distribution system (a broadcaster for a multitude of consumer) to a mostly a horizontal communication system (each consumer is also a broadcaster), many sites whose sharing communication between user's have appeared. These sites, as social networks (e.g., facebook, LinkedIn), social bookmarking sites (e.g., Connotea1), microblogging sites (e.g., Twitter), constitute the so-called, by some enthusiasts of the world, the "web 2.0 revolution", the "social web" or, more modestly, the "participative web" [30]. From this new wave of sites whose content is generated by their contributors, the Community Question-Answering (CQA) sites were born. These sites allow users to create questions, answer questions from other users, comment on various questions and answers, and judge the relevance of other users' answers using scoring devices (score of 5, positive vote / Negative, etc.).

Information retrieval systems (IRS) have seen the emergence of new types of tools called Community Question-Answering systems. It is the taking into account of the need for precise information of the user that motivated the emergence of such systems. Moreover, the need of the user's information should be enumerated, explored and a single query may involve different needed user's information[1] [2] [3]. For the query "swine flu", doctors may be interested in the pathogenesis of treatment solutions, while patients can be concerned by transmission and preventive measures. Recently, the understanding of the intent behind user queries attracted much attention in the search for information retrieval (IR) [4] [5] [6]. Practically, whenone type a search query, the search engine will try to match one's words with the best and most relevant web pages. In seconds, one will see thousands of keywords containing the word appears in a list. There are several things we can do with this list, but before to start dealing with it, we will sort it by the method of the recommendation of a list of items without calculation of prediction based

on the co-dissimilarity and the tree covering minimum weight.

2. PRELIMINARIES

2.1 Related Works

There are several Approaches to classify user intents. Generally, they can be divided into several categories. Jansen et al. in [9] [10] and [11] present a methodology developed to classify the user's intent in terms of content type specified by the query and other expressions of the user, a set of characteristics for each category in the taxonomy. Broder [43] reported three levels of categories for users, navigational, informational and transactional. The last two are obtained from manually classified queries[7]. A class is trying to raise queries with additional data, including search results returned for a query, the information from an existing corpus or an intermediate taxonomy. The second category of Broder levels of categories for users uses data unmarked to help improve the accuracy of supervised learning. The third category develops the training data by automatically marking some queries to certain click-through data by self-training. The anchor text and the results of search engines and the query text are used to represent an application. In [12], one has a relationship of dependency and characteristic words of the query text detection, bigramme duration and content features to represent a query. Some current research studies focus on identifying characteristics of each type of Queries. Ganti et al. [8] report using tag functionality to query based on the co-occurrence between the different types of tags and query terms.
Wen et al. Cluster similar applications according to their content and user logs. They suggested a similarity function based on the application and the content of search results compare two applications [14]. Dumais and Chen classified search results in predefined hierarchical categories such as Yahoo! Directory or Web LookSmart[13] . Carlos Cobos et al. introduces a new description-centric algorithm for clustering web results, called WDC-KSB, which is based on the meta-search heuristic algorithm cuckoo, k-means algorithm, balanced Bayesian Information Criterion, split and merge methods on clusters, and common phrases to approach cluster labeling [15].Then the documents were assigned to the relevant key phrases to form clusters candidates.
Wang and Zhai learned one aspect of the application data users query logs with a star clustering algorithm. They then categorized and search results organized according to the learned aspects [16]. Beeferman Berger and first built a bipartite graph with the click of data, including user queries and clicked URL. Then, they applied an agglomeration for the graph to the query and the URL [17] relative to the group. [18] First found two interesting phenomena of the user's intention: A search for clarification of sub-theme and keyword. The first means that if a user clicks multiple URLs in a query, and then click the URL tend to represent the same facet. This means that users often add additional keywords to extend the queries to clarify their user's intent to search. Based on these two phenomena, they grouped all clicked URLs and corresponding queries, where each group represents a plan. [19] Ranked first user's intents of the queries into two types according to their variation on the timeline: constant and sporadic.
Then they considered query logs as data flows continuously and divided into variable-length partitions. Finally, they grouped each partition into groups of URLs that represent the user's intents. [20]S ummarized similar queries concepts by combining bipartite click-through queries and URLs recorded from query logs. [21] Random used the approach on bipartite graphical URL queries to discover the attributes facet queries. [22] Found requests of user intentions with query logs. For a given query, they identified the first set of possibly linked queries and then used the March likeness algorithm chance to find clusters of the user's intent. [23] Groups of user queries grouped in the underlying mine of the user's intent. They modelled the behaviour of the users in the form of a Markov diagram combining occurrences of co-occurrence of documents, clicks and session co-occurrences, and then they made several random hikes on the graph to get clusters. Clustering / Classifier Terms related to the query. The existing work belonging to this management considers the intention of a request as a set of sub-user applications, namely the terms related to the request [24]. These candidate candidates can come from many sources such as search engine query suggestions, search queries related to log user queries, and so on. Currently, this area is one of the hot topics in the mining sector for application [25]. [26] Proposed an algorithm called dual C-Means to group search results in double representation spaces with the query logs. [27] First found similar applications as candidates for a given query to query logs. Then they used a bipartite graph click to narrow these similar applications.
They provided a taxonomy of keywords of intent from the rigorous manual analysis queries. Recently, this problem has been highlighted by many researchers the task consists of two phases:

the operation of the user's intent of the ranking. Participants of the job offer numerous methods [28].

Presented a method that gets the best performance in the Chinese data. Specifically, they ranked the first user's intents into two types: the explicit role the subject and the subject of the implicit role. For subjects of explicit role, they built a graphic modifier based on all the co-kernel-object chains. Then the graph of change has been divided into clusters with high intra-cluster interaction and relatively low inter-cluster interaction.

In summary, there is an increase in research on user intent queries recently, there are still some problems to be solved. First, the existing approaches consider the intent of the mining request and the classification from a static point of view. They ignore the issue of user intent derived from new intentions the user might emerge and the intention of the former user might become unpopular. In addition, issues of diversity and redundancy are not carefully considered in the classification of keywords on the coverage of the user's intent. Google Translate for Business: Google Translator KitGadget TranslationTools.

Second, most current measures of similarity for the keyword is usually constructed from a single point of view, either from queries or only from collections of documents. In addition, the combination of different similarity functions from multiple resources is usually defined heuristically.

This cannot accurately estimate the similarity between the keyword because of their short text characteristics.

In this paper, we propose a new approach to detect the user's intent by the method of the recommendation of a list of items without calculation of prediction based on the co-dissimilarity and the tree covering minimum weight based on the theory of graphs. To improve the ranking of a website in organic search results to increase visibility and quality.

3. PROPOSED WORK

Example: Suppose that user asks about chocolate for example as we can see in figure 1:

When someone tape chocolate into the query box on a search engine page (such as Google), we have about 200.000.000 result and we can found easy what we want.

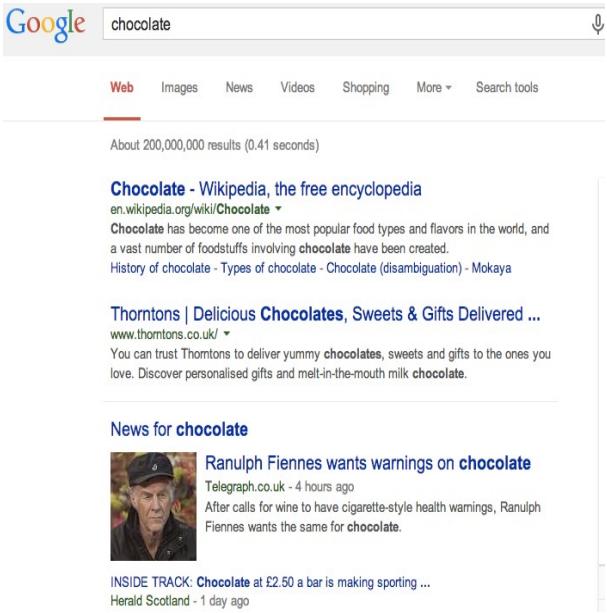

Figure 1: Example result box on a search engine page

"Chocolate" In this case, user used only one word. Actually, into IR system the answer on this question is all information's existing in database about "chocolate".

But our approach is what you see when you go to a search engine - it is the end of what everyone thinks like a search engine.

So when you type in your search terms and hit search, and the search engine will try to match your words with best pages most relevant web it can find a "web search.

You can just enter a keyword unipart (as we have said, if you're researching chocolate).

In seconds, you'll see up to 20,000 keywords containing the word "chocolate" appear in your list. There are several things you can do with this list, but let's start by talking we make any list, sorted by popularity, or search volume, competition, the pages have been directly optimized for each keyword and KEI - effectiveness index keywords.

To find the number of competitors on a keyword, it is very simple and you probably do already know, indeed just search the keyword on google.fr and the number of results for keyword sought appears but this solution just we give as many result and we need just the interested result.

Journal of Theoretical and Applied Information Technology
15th December 2021. Vol.99. No 23

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

4. OUR PROPOSITION

4.1 The recommendation of a list of relevant result for keyword:

It's easy to filter out keywords with high competition (high competition is bad) or low search volume (bad again, a low search volume means the keyword gets little search traffic).

To recommend a list of items or keywords to an active user, we propose a recommendation method based on the theory of graphs. The latter predicts the user's preferences without calculating the quality index prediction is an indicator of the relevance of the keyword. To extract a list of suggestions, the algorithms use the notion of similarity, the latter aims to give a value to the resemblance between two objects. Based on the usage matrix, we have two similarity matrices, between the items and the users respectively. Then, in order to classify the users or the items in the form of an overlapping tree of minimum weight, for each of the two matrices represented by a weighted connected graph, we are based on the kruskal algorithm [30]. Finally, the proposed method relies on these two trees to recommend a list of the best items. Indeed, using the matrix of votes, we calculate the matrix of similarity between the users as well as between the items, we obtain two square matrices of order N and M respectively. These two matrices have importance in the determination of families of items and similar users. To do this, we first compute Cosine similarity between two users or between two items using the formula described in equation 2 of the second chapter. Then, a transformation into dissimilarity is carried out as follows:

$$\mathbf{dis(ci, cj) = 1 - cosine(ci, cj)} \quad (1)$$

The two dissimilarity matrices are based on the formula (1). We realize a graphical representation for the two matrices by taking as weights of the edges the values of the dissimilarity coefficients. Moreover, our approach is based on the dissimilarity between the key words thus between the resources, which implies the use of the notion of co-dissimilarity. We will use these graphs to extract two trees of minimum weight corresponding to keywords and items. From the graphs corresponding to these two keyword and item dissimilarity matrices, we construct two trees covering minimal weights using the Kruskal algorithm [29] described in Table 1.

Table 1. The Kruskal Algorithm [29]

The Kruskal algorithm makes it possible to find a minimal value a tree of Recovery of a graph G = (X, U) Step1: Sort the edges of G by increasing value, place Step2: For each edge (x, y) and by increasing value do: If T U {(x, y)} is without cycle then Add (x, y) to T: T = T U (x, y)

The shaft shown will be obtained by eliminating the edges having a high weight. The keyword or item tree keeps the edges with the minimum values of the dissimilarity coefficients.

A. Keyword tree

The tree $Ar_{keywords}$ representing the dissimilarity matrix of the users is the following:

Let G (X, T) be the tree of the users, where X represents the set of vertices of users and T the set of edges whose weight is the dissimilarity coefficient between the users.

This tree represents the set of users classified according to minimal dissimilarity.

Let C be the set of n users $\{C_1, C_2, \ldots, C_n\}$, through $Ar_{keyword}$, for each keyword of C. We look for S_{Ci}: the set of neighboring keywords that are similar to keyword C_i. For each keyword C_j in the set S_{Cj}, we determine a set of items E_{Cj} with high quality indexes. Finally, we will have a set containing E_{Sci} all the keyword items of the set described in the following formula:

$$E_{SCi} = \coprod_{j=1}^{n} E_{Cj} \quad (2)$$

B. Item Tree

By the same principle for the tree of items $An_{keyword}$, we will pass from a simple graph where the vertices are the items and the edges represent the values of dissimilarity coefficient to a covering tree of minimum weight.

Let P be the set of m items $\{P_1, P_2, \ldots, P_m\}$ of our recommendation system, for each keyword C_i, we look E'_{Ci} for the set of items of P that are similar to the elements E_{Ci} of the set from the tree of items $An_{keyword}$. c

Result of recommendation

The intersection of the sets E'_{Ci} and E_{Sci} gives a new set of items similar to those of the set E_{Ci}, these are recommended by the keywords of S_{Ci}, which are considered the most similar to C_i.

Journal of Theoretical and Applied Information Technology
15th December 2021. Vol.99. No 23

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Taking an explanatory example of applying this recommendation method to the keyword.
Once we have the two trees covering the minimum weights shown in Figure 2, we will proceed to the recommendation phase of a list of items according to our proposed approach.

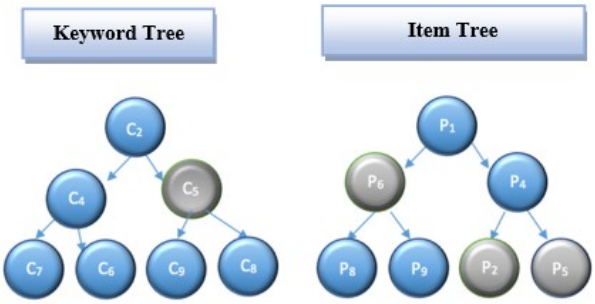

Figure 2: The Trees Of Items And Keyword

Using P = $\{P_1, P_2, P_4, P_5, P_6, P_8, P_9\}$ and C = $\{C_2, C_4, C_5, C_6, C_7, C_8, C_9\}$
Figure 9 illustrates this approach, the keyword C_5gave a high quality index to items $\{P_2, P_6\}$
Which builds the wholeE_{C5}. S_{C5}= $\{C_2, C_8, C_9\}$ The set of neighboring keywords that are similar to a given keywordC_5.
We determineE_{Sc5}= $\{E_{C2}, E_{C8}, E_{C9}\}$ = $\{P_1, P_2, P_6, P_8, P_9\}$, the set of items measured by S_{C5} the set and that is determined through the keyword tree $Ar_{keyword}$.
And we also determine E'_{C5}= $\{P_1, P_4, P_8, P_8\}$ the set of items belonging to P and which are similar to the elements of E_{C5}, to be determined from the tree of items Ar_{items}. as illustrated in Figure 3.

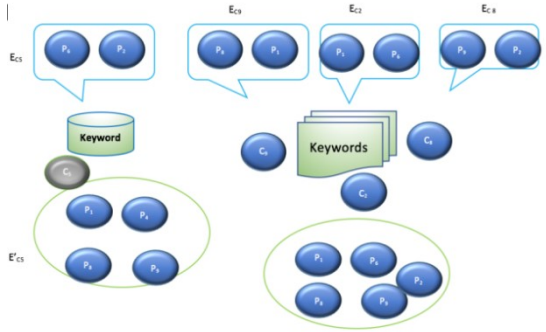

Figure 3: The Explanatory Diagram Of The Procedure For Recommending A List Of Items

The intersection of the sets $\acute{E}_{c5}$and E_{sc5}gives the set { P_1, P_8, P_9} of the items that are similar to those of the set E_{c5} (the set of items with high quality indexes by C_5) They are already recommended by the keywords that are considered closest and most similar to C_5, as illustrated in Figure 4

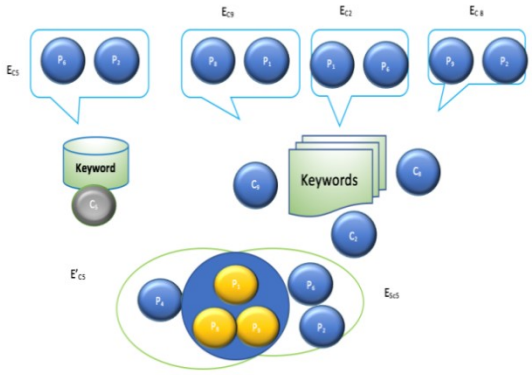

Figure 4: The Explanatory Diagram Of The Procedure For Recommending An Item List

In the case of an empty intersection, we will classify the items of the union of sets and based on the average value of the item quality indices, and we recommend those with the high averages. In summary, the steps of our procedure are presented in the following table:

Table 2. Steps In The Procedure For Recommending An Item List

1. Construct two minimal weight overlap trees of minimal dissimilarity of keywords and items by the Kruskal algorithm
2. For each keyword C_i, determine the set E_{Sci} described by the formula$E'_{SCi} = \coprod_{j=1}^{n} E_{Cj}$, from the keyword tree $Ar_{keyword}$
3. For each keyword C_i, we look for $E'_{Ci}$the set of items of P that are similar to the elements $E_{Ci}$of the set from the tree of items $Ar_{keyword}$
4. The intersection of the setsE_{Sci} and E'_{ci}constitutes the list to recommend.

We presented collaborative filtering for this type of systems to have best result, which is based on graphs and which aims to recommend a list of items without prediction computation.

Comparison of similarity indices

Our comparison will be based on the dendrogram structures of similarity indices obtained from their resemblance matrices and therefore the result will be a dendrogram grouping these indices into families. The goal is to know the closest similarity indices.
Different approaches that have been proposed, compare clusters, based on several performance indicators. On the other hand, to compare two classifications, our procedure is based on a distance

between structure dendrograms. It is a question of finding the dendrograms considered as hypergraphs by the hierarchical classification, and of comparing them from the distance Marczewski-Steinhaus [31].

Much of the work has been devoted to the presentation and the definition of the different indices which seem important to us in this regard, comparing directly resulting classes. When we have two partitions made on the same data, we must know if they agree or if they differ significantly. One way to approach this problem is to compute a concordance index between partitions and to define a critical value from which it will be concluded that the two partitions are or are not concordant. Most indices are presented in relational formulations using the passage formulas proposed by Kendall [33] and Marcotorchino [34]. At the well-known Rand index and the one corrected by Hubert [35], an asymmetric version of Rand [36] has been proposed and used for the comparison of nested partitions, with different numbers of classes. Two other indices inspired by Mac Nemar's test and Jaccard's index. The vector correlation index introduced by P. Robert and Y. Escoufier [37] which proves to be identical to the coefficient of S. Janson and J. Vegelius [38], the kappa coefficient of Cohen [39], the index of redundancy proposed (i) Stewart and Love [40], as well as the Popping index [41].

(ii)

The proposal of a comparison which is based on dendrograms and not on the comparison of the resulting classes directly, aims at providing a rational and efficient tool for the grouping of the different classification methods in order to locate them relative to each other. to others. This proposal will be a general approach including the choice of the closest methods and that of replacement families.

Our procedure for comparing classification methods consists in making a dendrogram resulting from the matrix of distances between all the trees of these classification methods. To do this, we use a hierarchical ascending classification algorithm taking into account the average link criterion. Then, from the classification tree of these methods and making cuts in the tree, we get a number of families. The choice of the cutting threshold depends on the criteria chosen (number of replacement classification methods, the partitions closest to a given partition, Ward's index) which condition our intended objectives. This method has the advantage of providing a global view of consistent families of methods.

In what follows, we will present the theoretical framework of the distance between trees [31] before the presentation of an explanatory example.

4.2 Theoretical frame :

Hypergraphs generated by trees

According to [31], Treated trees are a particular case of tree-generated hypergraphs whose family of nodes has special properties.

X= $\{x_1, x_2, x_3, \ldots\ldots, x_n\}$ the set of terminal vertices of a tree.

d-(xi) =1, d+(xi)=0 for any element of X or d-(xi), d+(xi) represent the inner and outer half-degrees of the knot xi respectively. Is A a class of all trees with X all the terminal vertices. Is A $\in$ A represented by the hypergraph (X, EA) where the class of arrest EA is defined as follows: each V $\notin$ X ie each non-terminal node in the tree generates d+(V) -1 arrest in EA. Such an arrest consists of those elements of X that are terminal nodes of the subtree generated by V and which is obtained by considering V it as a root, that is, assuming that d-(V) =0.

Our method of constructing the hypergraph H_A leads to the following insertion:

Proposal1:

(i) If H_A= (X,E_A) is the hypergraph generated by a tree A $\in$A as described above, then $|E_A|$ = n-1

(ii) The hypergraph generated by A is not simple if at least one of the nodes. By definition, a hypergraph is simple if all its edges are distinct.

Distances between trees

Let's say $|E_A|$= n ; where $|.|$ is the cardinal of the set X. Let be E the class of all sub-sets of X, and $\mu(E)$ the measurement of E on E . Let's consider$\mu(E) < \infty \forall E \in E$. The distance of Marczewski-Steinhaus[31] between two sets and from east:

$$\sigma_\mu(E_1, E_2) = \begin{cases} \frac{\rho(E_1,E_2)}{\mu(E_1 \cup E_2)} Si E_1 \cup E_2 > 0 \\ 0 \quad Si E_1 \cup E_2 = 0 \end{cases} \quad (3)$$

With $\rho(E_1, E_2)\mu(E_1 \Delta E_2), \Delta$ is the symmetrical difference.

It should be noted that $0 \leq \sigma_\mu(E_1, E_2) \leq 1$,

especially if we consider that $\mu_c(E) = |E|$ and and then pose $e_1 = |E_1|$, and $e_2 = |E_2|$ and $d = |E_1 \cap E_2|$.

$$\sigma_{\mu c}(E_1, E_2) = \frac{e_1 + e_2 - 2d}{e_2 + e_2 - d} \quad (4)$$

We also have: $0 \leq \sigma_{\mu c}(E_1, E_2) \leq 1$

Let us consider A_1andA_2, two elements of A represented by the hypergraphs $H_{A1} = (X, E_{A1})$ and $H_{A2} = (X, E_{A2})$respectively. The distance between

these hypergraphs takes into consideration the specific step of edge construction. The distance between trees is given by the following formula:

$d(A_1, A_2) = \frac{1}{n-1} \min_{p \in P} \sum_{i=1}^{n-1} \sigma_\mu \left(E_{A2}^i, E_{A2}^{pi}\right)$(5)

Where P_i is the $i^{ème}$ element of the permutation p of entire n-1. P is the set of all permutations, $\sigma_\mu(...)$ is given above. $E_{A1}^i \in E_{A1}$ and $E_{A2}^{pi} \in E_{A2}$ i=1, n-1.

The following facts are involved in the above definition:

(A,d) is a metric space.

$d(A_1, A_2) \leq 1, A_1$ and $A_2 \in A$ the distance d(.,.) < 1 if formula 27 is used $\sigma_\mu(...)$ instead $\sigma_{\mu c}(...)$ of formula 26.

Table3 The Steps In Our Procedure For Comparing Similarity Indices Are:

1. Construct the dendrograms corresponding to each similarity index from the similarity matrix obtained by the hierarchical classification (HAC) on the data matrix.
2. Compare the dendrograms 2 to 2 by the Marczewski-Steinhaus distance of formula (28).
3. Construct the matrix of distances between all dendrograms.
4. Obtain the final meta-dendrogram of all similarity indices

Example of distance between trees:

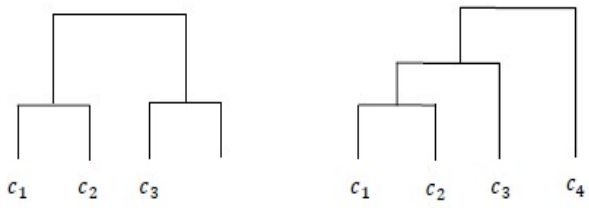

Figure 5: Trees Of Assembly

By taking X= $\{C_1, C_1, C_2, C_3, C_4\}$ all the components of a product, $A_1$and A_2 two possible connection shafts (figure 11). We calculate the proposed distance between the two ranges $A_1$and A_2 of A, based on all the components of X. $|X| = 4$.

To do this, we look for the assemblies E_{A1} and E_{A2} sub-trees corresponding to the intermediate stages of the product's constitution. These steps are the edges of the hypergraphs $H_{A1} = (X, E_{A1})$ and $H_{A2} = (X, E_{A2})$. According to proposal 1, the number of intermediate steps for the constitution of the product is $|E_{A1}| = |E_{A2}| = |X| - 1 = 3$.

To simplify ratings, we pose $c_i = i$:

$E_{A1} = \{\{1,2\}, \{3,4\}, \{1,2,3,4\}\}$ with $E_{A1}^1 = \{1,2\}$, $E_{A1}^2 = \{3,4\}$ and $E_{A1}^3 = \{1,2,3,4\}$.

$E_{A2} = \{\{1,2\}, \{3,4\}, \{1,2,3,4\}\}$

The distance $d(A_1, A_2)$given by the formula given by ***Marczewski-Steinhaus*** is calculated between the components of E_{A1} and the components of the permutations of E_{A2}. To do this, we look for the set P of permutations p of E_{A2}. P is described as follows:

$\{\{1,2\}, \{1,2,3\}, \{1,2,3,4\}\}$ with $E_{A2}^1 = \{1,2\}$, $E_{A2}^2 = \{1,2,3\}$ and $E_{A2}^3 = \{1,2,3,4\}$.

$\{\{1,2\}, \{1,2,3,4\}, \{1,2,3\}\}$ with $E_{A2}^1 = \{1,2\}$, $E_{A2}^2 = \{1,2,3,4\}$ and $E_{A2}^3 = \{1,2,3\}$.

The same approach is applied to search for E_{A2}^{pi} of the remaining permutations:

$\{\{1,2,3\}, \{1,2\}, \{1,2,3,4\}\}$
$\{\{1,2,3\}, \{1,2,3,4\}, \{1,2\}\}$
$\{\{1,2,3,4\}, \{1,2\}, \{1,2,3\}\}$
$\{\{1,2,3,4\}, \{1,2,3\}, \{1,2\}\}$

We calculate the distance between $E_{A1}$and permutations of E_{A2}. The minimum of the values obtained ensures the distance between the trees A_1andA_2. In the case discussed above, $d(A_1, A_2)$=0.25.

4.3 PRECISION AND RECALL

In the case of constructing a list of items, we are not based on a prediction heuristic, but rather we propose a graphical or structural method based on *results of the comparison of similarity coefficients* sets, which involves the use of evaluation metrics such as Accuracy and reminder.

When attempting to predict whether a keyword is suitable for an item, four possibilities are offered by the confusion matrix.

Table 4. Confusion Matrix For Recommending An Item To A User

Item	Relevant	Not relevant
Recommended	True Positive (tp)	False Positive (fp)
Not recommended	False Negative (fn)	True Negative (tn)

Precision [31] is the percentage or number of suggested items that is truly relevant to the user. For example, if one considers a list of the Top-N results, the precision corresponds to the proportion of items actually consumed, appreciated by the current user. It is calculated using the following expression:

$$\text{Precision}=\frac{tp}{tp+fp} \quad (6)$$

The recall [32] measures the number of relevant recommendations issued against the total number of relevant recommendations. Concretely, one enumerates the number of items whose associated measure is non-zero and is found among the suggested items, it is calculated by the formula below:

$$\text{Recall}=\frac{tp}{tp+fn} \quad (7)$$

If the accuracy is low, the user will be dissatisfied because he will have to waste time reading items that do not interest him. If the reminder is weak, the user will not have access to an item he / she would like to have.

A system that contains perfect results must have a Precision and a reminder close to 1, but these two requirements are often contradictory and a very strong Precision can only be obtained at the price of a weak reminder and vice versa.

5. EXPERIMENTATION

We will apply our comparison method to indices applied in the field of industry but which can be used in recommendation systems.

We used the database of [32] presented in (Table 4).

TABLE 5 The Matrix Of Eight Machines

Machines \ Keyword	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	1	1					1	1		1		1	1		1	1		1	
2			1	1		1	1							1				1		1
3		1						1	1		1		1	1		1	1		1	
4			1	1		1	1			1								1		1
5	1				1	1				1		1			1		1			
6	1				1				1	1		1			1					1
7			1	1		1	1				1	1						1		1
8			1	1		1	1											1		1

We have 8 machines whose groups must be identified in order to create production cells. Each cell will contain a number of machines that process a product family. The 20 similarity indices compared in[32] are used here to classify the machines in the matrix into families. Figure 24 shows all possible CAH classification dendrograms. Our method, based on these structures, calculates the distances between these dendrograms 2 to 2, using the CAH, we obtain a meta-dendrogram classifying the 20 methods.

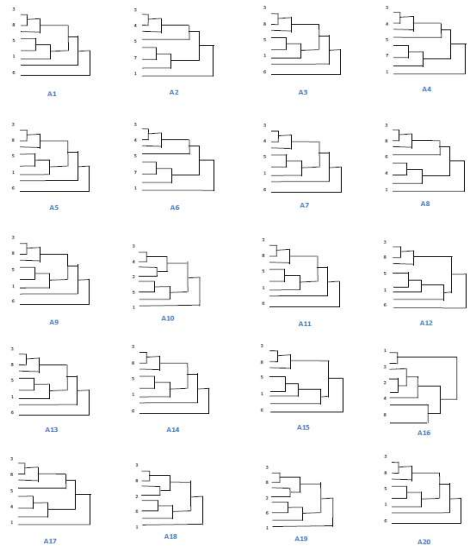

Figure 6 The Classification Of 20 Similarity Indices

Journal of Theoretical and Applied Information Technology
15th December 2021. Vol.99. No 23

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

TABLE 6: Matrix Of Distances Between Trees Of Coefficient Indices

	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}	A_{12}	A_{13}	A_{14}	A_{15}	A_{16}	A_{17}	A_{18}	A_{19}	A_{20}
A_1	0	0.3810	0	0.3810	0.0357	0.3095	0.3146	0.3478	0.0357	0.3748	0.0357	0.0816	0.0357	0.0357	0.0816	0.5722	0.1097	0.3587	0.3388	0
A_2		0	0.3810	0	0.3810	0.1071	0.0816	0.5000	0.3810	0.1857	0.3809	0.3963	0.3810	0.3810	0.3963	0.5607	0.3238	0.4976	0.4976	0.3810
A_3				0.3810	0.0357	0.3095	0.3146	0.3478	0.0357	0.3748	0.0357	0.0816	0.0357	0.0357	0.0816	0.5723	0.1097	0.3388	0.3388	0
A_4					0.3810	0.1071	0.0816	0.5000	0.3809	0.1857	0.3809	0.3963	0.3810	0.3810	0.3963	0.5607	0.3238	0.4976	0.4976	0.3810
A_5					0	0.3095	0.3121	0.3197	0	0.3476	0	0.1097	0	0	0.1097	0.5697	0.0816	0.3048	0.3048	0.0357
A_6						0	0.1862	0.5000	0.3095	0.2571	0.3095	0.3172	0.3095	0.3095	0.3172	0.5607	0.2524	0.4677	0.4677	0.3095
A_7							0	0.4578	0.3120	0.1811	0.3120	0.3810	0.3121	0.3121	0.3810	0.5723	0.3044	0.4701	0.4701	0.3146
A_8								0	0.3197	0.4000	0.3197	0.3963	0.3197	0.3197	0.3963	0.5488	0.2932	0.2214	0.2214	0.3478
A_9									0	0.3476	0	0.1097	0	0	0.1097	0.5697	0.0816	0.3048	0.3048	0.0357
A_{10}										0	0.3476	0.4024	0.3476	0.3476	0.4024	0.5440	0.3310	0.4571	0.4571	0.3748
A_{11}											0	0.1097	0	0	0.1097	0.5697	0.0816	0.3048	0.3048	0.0357
A_{12}												0	0.1097	0.1097	0	0.5607	0.1786	0.3833	0.3833	0.0816
A_{13}													0	0	0.1096	0.5697	0.0816	0.3048	0.3048	0.0357
A_{14}														0	0.1096	0.5697	0.0816	0.3048	0.3048	0.0357
A_{15}															0	0.5607	0.1786	0.3833	0.3833	0.0816
A_{16}																0	0.5488	0.5440	0.5440	0.5723
A_{17}																	0	0.3143	0.3143	0.1097
A_{18}																		0	0	0.3388
A_{19}																			0	0.3388
A_{20}																				0

5.1 . Calculation of distances

To be able to compare these similarity indices, we will compare the dendrogram structures obtained above using the Marczewski distance - modified Steinhaus[31].

For example, for the first two dendrograms in Figure 6, the sets of intermediate steps for classification $\mathbf{E}_{A1}$ and $\mathbf{E}_{A2}$are given. They are directly involved in the calculation of the distance "d" between A_1 and A_2as described in the example in the first paragraph.

$$|E_{A1}| = |E_{A2}| = |X| - 1 = 7$$

$$E_{A1} = \begin{Bmatrix} \{3,8\}, \{1,6\}, \{3,8,5\}, \{1,6,2\}, \{3,8,5,1,6,2\}, \\ \{3,8,5,1,6,2,7\}, \{3,8,5,1,6,2,7,4\} \end{Bmatrix}$$

$$E_{A2} = \begin{Bmatrix} \{3,4\}, \{1,6\}, \{3,4,5\}, \{1,6,8\}, \{3,4,5,7\}, \\ \{3,4,5,7,1,6,8\}, \{3,4,5,7,1,6,8,2\} \end{Bmatrix}$$

$d(A_1, A_2)$=0.3810

The distances calculated between the different trees are given in Table 16.

5.2. Exploitation

From the distance matrix, and applying the upward hierarchical classification algorithm, a meta$A$. dendrogram is obtained (Figure 7). From the latter and by cutting according to the objectives set, we acquire a number of index families.

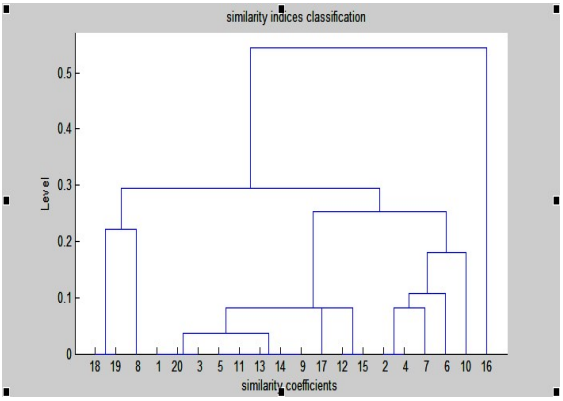

Figure 7: Classification Of Similarity Indices.

To search for index families, it is necessary to cut in the classification tree at an appropriate level. We will choose 0.221 $\alpha \leq$0.252, ensuring the minimum of inertia within classes.

For this level, we obtain the 4 index families shown in Figure 7.

TABLE 7 Coherent Index Families

Famille F_1	F_1= {18,19,8}
Famille F_2	F_2= {1,20,3,5,11,13,14,9,17,12,15}
Famille F_3	F_3= {2,4,7,6,10}
Famille F_4	F_4= {16}

Based on[32], we obtain the same results. We can say that the F2 family contains the best performing indices such as Jaccard, Sorenson, Kulczynski and Sokal and Sneath 2, but the F3 family is the one of the ineffective indices, namely: Hamann, Simple matching, Rogers and Tanimoto.

In this work, we have presented a procedure that aims to provide a rational and effective tool for grouping and comparing different classification methods.

We can also situate the methods in the literature to each other by comparing the dendrograms of these methods. This helps us to keep the best methods for given problems.

If we consider that we have several partitions corresponding to several methods for a given problem, we can use our method to find the consensus partition. In our case, it corresponds to a consensus dendrogram based on all the dendrograms found

5.3. The results of the recommendation of a list of items (RMCS) for a keyword

▪

Table 8 Precision / Recall

Percentage of elimination	Neighborhood	Precision	Recall
25%	1 2 3	0.9604 0.7610 0.5809	0.2501 0.4812 0.6324
50%	1 2 3	0.3416 0.2700 0.2031	0.1435 0.1863 0.2500
75%	1 2 3	0.1781 0.402 0.1373	0.1166 0.0265 0.0085

- **Evaluation parameters of the recommendation algorithm**

To evaluate our algorithm, the calculation of accuracy and recall requires a test basis to measure the performance of our RMCS method. For this, we propose the following solution:

For each keyword in the test set, provide a subset of the measurements. For example, bleach an interval of measurements. Thus, the aim is to predict the quality keyword for the bleached items and the error made by the algorithm is measured during the prediction of the measurement with respect to the true measure previously cleared. In this way, it is possible to detect all the relevant items All relevant and non-recommended data.

- **Data Elimination Percentage**

By varying the bleached interval as shown in Table 7, we observe an effect on the accuracy and recall values. As a result, we find that when 25% of data is eliminated, good results are obtained for accuracy and recall because the remaining 75% of the data contain a lot of information about the user who is helping

Our approach to achieve good result. On the other hand, when 75% of the information is eliminated, unsatisfactory results are obtained.

- **Neighborhood**

Our approach is based on the neighborhood item / item and keyword/keyword to perform the recommendation of a result list. By varying this parameter, we find that: when increasing the neighborhood, a large list of items will be returned. This increases the recall and decreases the accuracy and vice versa. If the keyword does not need a complete list of all the potential elements, but just a list of relevant items, then only the precision may be appropriate. But, if the task is to find all the relevant elements, the recall becomes important, so it depends on our need.

Taking the case of figure. 5 which presents the curve obtained for our approach for the neighborhood of 3 with 25% of data eliminated.

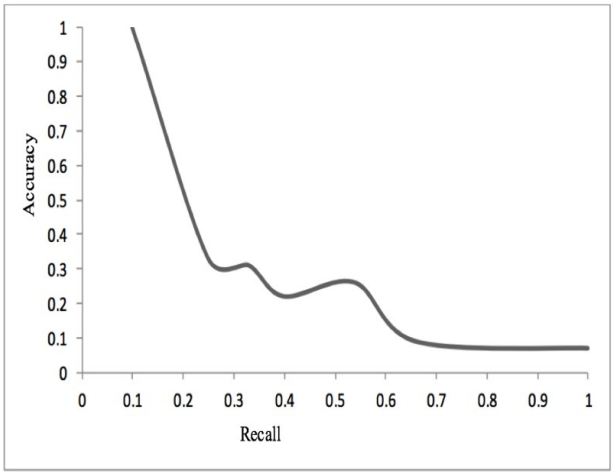

Figure 8: The Precision / Recall Curve

We note that for low values of accuracy, recall becomes important and vice versa. We can select the optimum value for our system it depends on our need.

If the accuracy is low, the user will be dissatisfied because he will have to waste time reading items that do not interest him. If the reminder is weak, the user will not have access to an item he / she would like to have. A perfect recommendation system must have a precision and a reminder close to the value 1, but these two requirements are often contradictory and a very high precision can be obtained only at the price of a weak recall and vice versa.

Users trust the referral system when they recommend items they like. Keyword satisfaction decreases when a significant number of errors are produced by the system. They thus promote precision measurement on that of the recall. They point out that for any trade recommendation system, the most important is to avoid false positives. Thus, the level of user satisfaction can be easily established.

In this sense, we have established this method to have a personalized recommendation. First, this method captures and capitalizes on user preferences to guide them in their choices. Then, the user himself is

Favored by a saving of time and a discovery of items often hidden to which he would not have thought.

In the next section, we compare the similarity indices used in the industrial domain and the recommendation system domain.

6. CONCLUSION

We first propose a formal framework for searching user's intent in search retrieval by dialogue. Second, we propose also SEO application. Our approach is tested using many knowledge bases for have interesting results. With SEO the user can ask system in different domain and with different language. In our future work, SEO will use the more developed communication unit...

REFERENCES:

[1] Hu, Y., Qian, Y., Li, H., Jiang, D., Pei, J., & Zheng, Q. (2012). Mining query subtopics from search log data. In Proceedings of the 35th international ACM SIGIR conference on research and development in information retrieval, ACM (pp. 305–314).

[2] Sakai, T., Dou, Z., Yamamoto, T., Liu, Y., Zhang, M., Kato, M. P., Song, R., & Iwata, M. (2013). Summary of the ntcir-10 intent-2 task: Subtopic mining and search result diversification. In Proceedings of the 36th international ACM SIGIR conference on research and development in information retrieval, ACM (pp. 761–764).

[3] Qian, Y., Sakai, T., Ye, J., Zheng, Q., & Li, C. (2013). Dynamic query intent mining from a search log stream. In Proceedings of the 22nd ACM international conference on conference on information and knowledge management, ACM (pp. 1205–1208)

[4] Dou, Z, Hu, S., Chen, K., Song, R., Wen, J. R. (2011a). Multi-dimensional search result diversification. In Proceedings of the fourth ACM international conference on web search and data mining, ACM, WSDM'11 (pp. 475–484).

[5] Dang, V., & Croft, B. W. (2013). Term level search result diversification. In Proceedings of the 36th international ACM SIGIR conference on research and development in information retrieval, ACM (pp. 603–612).

[6] Dang, V., & Croft, W. B. (2012). Diversity by proportionality: An election-based approach to search result diversification. In Proceedings of the 35th international ACM SIGIR conference on research and development in information retrieval, ACM, New York, NY, USA, SIGIR '12 (pp. 65–74).

[7] Cao Huanhuan, Hao Hu Derek, Shen Dou, Jiang Daxin, Sun Jian-Tao, Chen Enhong, Yang Qiang, Context-Aware Query Classification, The 32nd Annual ACM SIGIR Conference, pp. 3-10, 2009

[8] GantiVenkatesh, König Arnd Chistian, Li Xiao, Precomputing Search Features for Fast an Accurate Query Classification, Proceedings of the third ACM International Conference on Web Search and Data Mining ACM, pp. 61-70, 2010.

[9] Jansen Bernard J., Booth Danielle, Classifying Web Queries by Topic and User Intent, Proceedings of the 28th of the international conference extended abstracts on Human factors in computing systems, pp. 4285-4290, 2010.

[10] Jansen Bernard J., Booth Danielle L., Spink Amanda, Determining the informational, navigational, and transactional intent of Web queries, Journal Information Processing and Management: an International Journal archive V. 44 Issue 3, pp. 1251-1266, 2008.

[11] Jansen Bernard J., Booth Danielle L., Spink Amanda, Determining the User Intent of Web Search Engine Queries, Proceedings of the 16th international conference on World Wide Web ACM, pp. 1149-1150, 2007.

[12] Wu Dayong, Zhang Yu, Zhao Shiqi, Liu Ting, Identification of Web Query Intent Based on Query Text and Web Knowledge, Pervasive Computing Signal Processing and Applications (PCSPA), First International Conference, pp. 128-131, 2010.

[13] Chen, H., &Dumais, S. (2000). Bringing order to the web: Automatically categorizing search results. In Proceedings of the SIGCHI conference on human factors in computing systems, ACM, SIGCHI'00 (pp. 145–33).

[14] Wen, J., Nie, J., & Zhang, H. (2001). Clustering user queries of a search engine. In Proceedings of the 10th international conference on world wide web, ACM, WWW'01 (pp. 162–168).

[15] Cobos, C., Mun˜oz-Collazos, H., Urbano-Mun˜oz, R., Mendoza, M., Leo´n, E., & Herrera-Viedma, E. (2014). Clustering of web search results based on the cuckoo search algorithm and balanced Bayesian information criterion. Information Sciences, 281, 248–264.

[16] Wang, X., &Zhai, C. (2007). Learn from web search logs to organize search results. In Proceeding s of the 30th annual international ACM SIGIR conference on research and

development in information retrieval, ACM (pp. 87–94).
[17] Beeferman, D., & Berger, A. (2000). Agglomerative clustering of a search engine query log. In Proceedings of the sixth ACM SIGKDD international conference on knowledge discovery and data mining, ACM (pp. 407–416).
[18] Hu, Y., Qian, Y., Li, H., Jiang, D., Pei, J., & Zheng, Q. (2012). Mining query subtopics from search log data. In Proceedings of the 35th international ACM SIGIR conference on research and development in information retrieval, ACM (pp. 305–314).
[19] Qian, Y., Sakai, T., Ye, J., Zheng, Q., & Li, C. (2013). Dynamic query intent mining from a search log stream. In Proceedings of the 22nd ACM international conference on conference on information and knowledge management, ACM (pp. 1205–1208).
[20] Cao, H., Jiang, D., Pei, J., He, Q., Liao, Z., Chen, E., & Li, H. (2008). Context-aware query suggestion by mining click-through and session data. In Proceedings of the 14th ACM SIGKDD international conference on knowledge discovery and data mining, ACM (pp. 875–883).
[21] Fujita, S., Machinaga, K., &Dupret, G. (2010). Click-graph modeling for facet attribute estimation of web search queries. In Adaptivity, personalization and fusion of heterogeneous information, RIAO'10 (pp. 190–197).
[22] Radlinski, F., Szummer, M., &Craswell, N. (2010). Inferring query intent from reformulations and clicks. In Proceedings of the 19th international conference on world wide web, ACM, WWW'10 (pp. 1171–1172).
[23] Sadikov, E., Madhavan, J., Wang, L., & Halevy, A. (2010). Clustering query refinements by user intent. In Proceedings of the 19th international conference on world wide web, ACM, WWW'10 (pp. 841–850).
[24] Xue, Y., Chen, F., Zhu, T., Wang, C., Li, Z., Liu, Y., Zhang, M., Jin, Y., & Ma, S. (2011). Thuir at ntcir-9 intent task. In NTCIR-9 workshop meeting (pp. 123–128).
[25] Aiello, L. M., Donato, D., Ozertem, U., &Menczer, F. (2011). Behavior-driven clustering of queries into topics. In Proceedings of the 20th ACM international conference on information and knowledge management, ACM, CIKM'11 (pp. 1373–1382).
[26] Moreno, J. G., Dias, G., &Cleuziou, G. (2014). Query log driven web search results clustering. In Proceedings of the 37th international ACM SIGIR conference on research and development in information retrieval, ACM (pp. 777–786).
[27] Dang, V., Xue, X., &Croft, W. B. (2011). Inferring query aspects from reformulations using clustering. In Proceedings of the 20th ACM international conference on information and knowledge management, ACM, CIKM'11 (pp. 2117–2120).
[28] Sakai, T., Dou, Z., Yamamoto, T., Liu, Y., Zhang, M., Kato, M. P., Song, R., & Iwata, M. (2013). Summary of the ntcir-10 intent-2 task: Subtopic mining and search result diversification. In Proceedings of the 36th international ACM SIGIR conference on research and development in information retrieval, ACM (pp. 761–764).
[29] Maunendra, S. Sudeshna Sarkar PabitraMitraDesarkar, Aggregating Preference Graphs for Collaborative Rating Prediction ISBN: 978-1-60558-906-0. 2010.
[30] O'Reilly, Tim, 2007. "What Is Web 2.0: Design Patterns and Business Models for the Next Generation of Software," MPRA Paper 4578, University Library of Munich, Germany.
[31] M. Karonski and Z. Palka . On MarZeweski-Steinhaus Distance between Hypergraphs, Expositione Mathematicae, 1977
[32] Yasuda, Y. Yin et K., Similarity coefficient methods applied to the cell formation problem: a comparative investigation. Computers & Industrial Engineering , 2005
[33] Kendall M.G, Stuart A. The Advanced Theory of Statistics, Griffin, Londre, 1961
[34] F. Marcotorchino. Utilisation des Comparaisons par Paires en Statistique des Contingences. Etude du Centre Scientifique IBM France, 1984
[35] L. Hubert, P. Arabie. Comparing Partitions. Journal of Classification, Vol. 2, p.193-198, 1985
[36] Chavent, M., et al., Critère de Rand Asymétrique, in Proceedings SFC, 8èmes rencontres de la Société Francophone de Classification, Pointe à Pitre, 2001
[37] Robert P., Escoufier, Y. A Unifying Tool for Linear Multivariate Statistical Methods, the RV-coefficient. Appl. Statist., Vol. 25, p.257-265, 1976
[38] S.Janson , J.Vegelius. The J-index as a Measure of Association for Nominal Scale Response Agreement. Applied psychological measurement, Vol. 16, p. 243-250, 1982
[39] J.Cohen . A Coefficient of Agreement for Nominal Scales. Educ. Psychol. Meas., Vol.20, p.27-46, 1960

[40] D.Stewart , W.Love , A General Canonical Correlation index, Psychological Bulletin, Vol.70, p. 41- 163, 1968
[41] Popping, R. Traces of agreement. On the Dot-Product As a Coefficient of Agreement. Quality and Quantity, Vol.17, N1, p.1-18, 1983
[42] Muhammad Ismail, Irfan Jamil, Rehan Jamil "Using SEO techniques Google Panda to Improve the Website Ranking" International Journal of Engineering Works, Vol.1, Issue.1, PP. 6-5, Sept. 2014.
[43] Broder Andrei, A taxonomy of web search, ACM SIGIR, V. 36, Issue 2, pp. 3-10, 2002.