A New Graph-Based Approach for Document Similarity Using Concepts of Non-Rigid Shapes

Lorena Castillo Galdos[1], Grimaldo Dávila Guillén[1], Cristian López Del Alamo[1,2]
[1]Universidad Nacional de San Agustín, [2]Universidad La Salle
Computer Science
Arequipa, Perú
e-mail:lcastillo@unsa.edu.pe, gdavila@unsa.edu.pe, clopez@ulasalle.edu.pe

***Abstract*—Most methods used to compare text documents are based on the space vector model; however, this model does not capture the relations between words, which is considered necessary to make better comparisons. In this research, we propose a method based on the creation of graphs to get semantic relations between words and we adapt algorithms of the theory of non-rigid 3D model analysis.**

Keywords-document similarity; keypoints; keywords; graph based comparison; non rigid shapes

I. INTRODUCTION

The development of technology and, specifically, of the Internet and storage devices, has grown exponentially in the last years, providing a great quantity of textual information, but also generating new challenges. For instance, some of these challenges include document analysis based on the document structure grammar, plagiarism detection, text content search, and others. These problems are converted into areas of interest in the community of Information Retrieval.

As a consequence, in the last years, investigations to generate algorithms for information retrieval by content have been developed. One of the most common methods is the vector-space model [4]; however, this approach does not capture the semantic relations between documents.

On the other hand, in the last years, many algorithms applied to similitude search in non-rigid three-dimensional models have been developed. These algorithms have the advantage of retrieving similar topology three-dimensional models; i.e. they are invariant to non-rigid transformations, like isometric changes and noise presence, among others.

Furthermore, there are many areas in computer science that can provide some ideas and concepts that can be applied to information retrieval. For instance, three-dimensional models and documents can be treated like graphs; then, graph theory based algorithms may be used to analyze the existence of isomorphism patterns and semantic similitudes between the objects.

There are three main contributions of this paper. First, we apply concepts of three-dimensional invariant models such as key-points and K-rings, which are adapted to generate an algorithm for semantic document comparison. Second, we introduce a new form of creating document keypoints-based graphs, and finally, we propose a new approach for key-point comparison in text graph representation.

The rest of this paper is organized as follows. In Section 2, the related work is presented. In section 3, we show the concepts of keypoint and document analysis adapted key components. Section 4 describes the proposal and methodology applied in this research. Section 5 shows the experiments and evaluations and, finally, Section 6 presents the conclusions.

II. RELATED WORK

There exist works in the literature in which graphs are used to compare and classify documents [7][8][11]. The creation and use of text graphs may vary according to their application. These can be term graphs, document graphs, and category graphs, among others.

When we make a semantic comparison between documents, the entered data or the documents itself may change; so, the output data and the techniques must also change. For this reason, Pilehvar et al. [8] proposed a graph based unified approach for measuring the semantic similarity and a multiple-level item comparison. Namely, they proposed sense, words and text levels.

Similarly, in [7] a document is represented as a compact concept graph. Here, the nodes represent extracted concepts from the document and the edges represent the semantic and structural relations between them, which are used to measure the semantic similarity between documents.

To measure the similarity between documents in a category, Wang et al. [11] propose the generation of a term graph. The objective is to represent the document content and the relation between words in order to define new functions of graph-based similarity. This allows combining the advantages of the vector-space model and o-occurring terms (*frequent itemset mining method* [6]).

ISBN: 978-1-61208-566-1

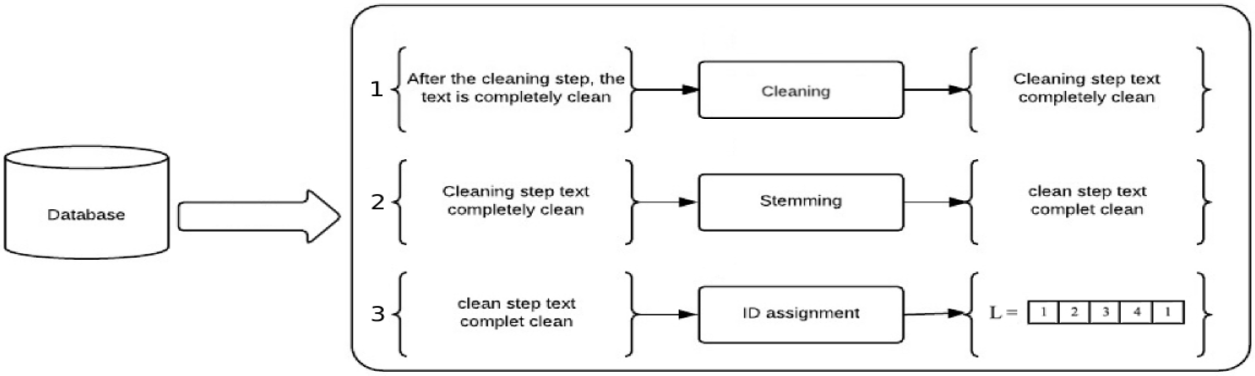

Database
1
After the cleaning step, the text is completely clean
Cleaning
Cleaning step text completely clean
2
Cleaning step text completely clean
Stemming
clean step text complet clean
3
clean step text complet clean
ID assignment
L = 1 2 3 4 1

Figure. 1. Preprocessing

In [1], neighboring document graphs are generated in one hyperlink space, in which the nodes are the Web pages, and the edges are the hyperlinks between them, which helps with the task of classifying and labeling each category.

Unlike previous research papers that focus on creating maximum co-occurrence graphs and word frequency based graphs, we concentrate on determining keywords. These words are the ones that not only have high frequency but also have a strong relation inside a word neighborhood. This concept is taken from the term keypoints [5] used in non-rigid three-dimensional models, aiming to determine high curvature localized vertices. With this, a 3D model with 10 000 vertices is reduced to a small subset if these vertices represent high semantic sense zones of the three-dimensional model. Similarly, in this paper, we propose to determine a set of keywords that represent high semantic content zones of the document.

In the next section, we specify the concepts of keypoints and k-rings and their respective adaptations to the information retrieval field.

III. PRELIMINARY CONCEPTS

A. Keypoints and Keywords

In 3D models, a keypoint is a point that holds some distinctive characteristics inside its neighborhood and it is present in different object instances [5].

On the other hand, in a similar way, the keywords (kw) of a document are defined as the words that bring more semantic information about a set of neighbor words. So, its frequency is high, and at the same time, the degree in which this keyword is related to its neighbors is seen many times in the document.

B. K-rings and Neighborhood

In 3D models a k-ring $R_k(v)$of k profundity level with center on the vertex v is defined by:

$$R_k(v) = \{v' \in V', | C(v',v) | = k \} \tag{1}$$

where $C(v',v)$ is the shortest path from vertex v' to v and $|C(v',v)|$, is the size of the path $C(v',v)$. It is important to mention that the size of an edge is always 1 [3].

The adapted concept of k-ring for documents is called neighborhood. The neighborhood of a node *n* iscomposed of all the nodes inside a radius ρ having the center on the node *n* which has to exist in both graphs to be compared.

IV. METHODOLOGY

In this section, we describe the necessary stages to obtain the most relevant characteristics of a document using adapted techniques of similarity search in non-rigid three-dimensional models. These stages are divided into three phases. The first stage, named preprocessing stage, is summarized in Fig. 1 and described in the next subsection. The second stage is called graph construction and finally, the third stage is graph comparison.

A. Preprocessing

1. **Cleaning**: Because not all the words in the document bring relevant information (like *stop words*), it is required to eliminate them, and usually these are the most frequent, for example: pronouns, articles, etc.
2. **Stemming**: One of the problems that occur in natural language is that a word can have different variations of time, gender, and number; these variations affect the computational calculation because a word represented differently can be interpreted in a different manner, namely, as two separate nodes in the graph. To avoid this problem, we use the Porter [9] algorithm, which allows us to make a stemming process, and hence obtain the roots of the words after the cleaning process.
3. **ID Assignment:** We manage to handle the roots in a different manner, assigning a unique numeric identifier to each root (ID), to insert them later in a list L, which will contain all the roots' IDs of the document in the occurrence order in the text.

$$L = \{ id_1, id_2, ..., id_t\} \tag{2}$$

where id_i is the *id* of the word in the position *i*. For example, if the word *friend* has the *id = 5*, then all the occurrences of the word friend in the text will have the *id = 5*. This identifier is assigned with the objective of handling the term graph with integer numbers instead of words, and to accelerate the comparison algorithm between edges or vertices. Finally, t represents the number of words of the text after the cleaning process.

B. Graph construction

After the preprocessing stage, we build the graph $G(N, A, W)$ where N are the nodes of the graph, which represent the elements of the list L, i.e., the representative words of the text. Set A indicates the edges, which represent the relations that exist between the elements of the list L, and set W the weights of each edge; this weight accounts for the degree of importance of that relation.

The protruding edges of a node N_i represent the degree in which this node is related to its neighbor nodes. That is, the degree in which a word is related to adjacent words in the text. This degree is represented by the value $K >= 1$ as it is shown in Fig. 2.

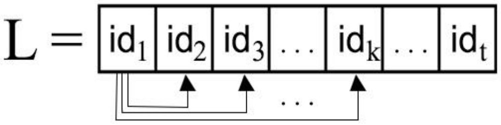


Figure. 2. Degree K in the list L

In equation (3), we formally describe the edges A of the graph $G(N, A, W)$:

$$A = \{l_i l_{i+1}[w_{i,i+1}], \ldots, l_i l_k[w_{i,k}]\} \quad 1 <= i <= s \qquad (3)$$

where w_{ij} is the weight between the edge i and the edge j, $i < j < k$, l_i is an element of the list L and s is the number of edges in the graph.

To each node, we assigned a weight ψ that will be the summation of the weights of the adjacent edges of the node; this can be observed in Fig. 3 b), which will serve to determine the keywords in the text.

C. Comparison

In the comparison stage, we obtain the keywords (kw) of the compared graphs.

Let G_1 and G_2 be two graphs that represent two documents. Then, the keywords of G_1 and G_2 are those μ nodes whose weights are the maximum. In other words, if we order all the vertices of G_1 and G_2 decreasingly and take the first μ nodes of both lists, then we have the keywords of both graphs.

Then, we find the intersection of both lists, so that the nodes with more weight, that are both in G_1 and G_2 will be the set of common keywords between both graphs. This can be represented formally in (4).

$$KW(G_1, G_2) = max_\mu(G_1) \cap max_\mu(G_2)\} \qquad (4)$$

where max_μ represent the μ higher values, G_1 and G_2 are the graphs that represent two different documents, and finally $KW(G_{1,}G_2)$ is the set of common keywords between $G_1$and G_2.

On the other hand, considering that an edge represents the relation between two words *(a,b)* of the text T, and its weight w is the number of times this *(a,b)* relation is repeated in the document, to find the distance or dissimilarity between two graphs, we propose to use the inverse of the weights of the edge *w* so we can get the distance between two graphs. This is shown in (5).

$$D_{a,b} = \{\frac{1}{w_{a,b}}\} \qquad (5)$$

In Fig 3 a), a graph is shown, where the vertices represent the words, and the vertices labels are the *ids* of those words. On the other hand, the edges represent the relations between neighbor words, and their respective labels are the number of times that the relationship appears in the document.

For each node, we calculate the sum of the weights of the protruding edges, as we can observe in Fig. 3 b); so that, the higher the value of one node, the greater the strength of the relations it maintains with its neighbors. So, a node with a higher value is probably a word of high importance in the text.

Finally, as it is shown in Fig. 3 c), the weights of the edges have been inverted according to (5), so we can apply the Dijkstra algorithm and find the neighborhood of a vertex v inside a ρ radius.

To find the keypoints, we must consider the neighborhood inside a ρ radius of a node, and, as we can see in Fig. 4 a), v is the key point from which the k-rings will be taken. The color nodes represent the neighbors of *v*, and each color represents a different neighborhood. In Fig. 4 b), the concept of k-rings is adapted, so that ρ represents the radius and all the nodes inside of it are the neighborhood of the node *n*.

In Fig. 5, we apply the Dijkstra algorithm to the graphs $G_1$and G_2 for comparison. We do this to obtain the minimum distance from the nodes of the list L_{kw} and the rest of the nodes in both graphs, as shown in Fig. 5 a) and 5 b).

Next, based on the idea of [2], (6) formally describes the way of finding the neighborhood, which is the disjoint union ⊔, of the intersections of the adjacent nodes to the keywords inside a ρ radius.

$$R = \{F\rho(L_{kw_1}) \sqcup \ldots \sqcup\ F\rho(L_{kw_{|L_{kw}|}})\ \} \qquad (6)$$

where $F\rho(L_{kw_j}) = \{n \in G_1, G_2 \colon D(n, L_{kw_j}) <= \rho\}$, D denotes the shortest distance between the node n and L_{kw_j}through the Dijkstra algorithm, n are all the nodes whose distance D is less than a radius ρ, as shown in Fig. 5.

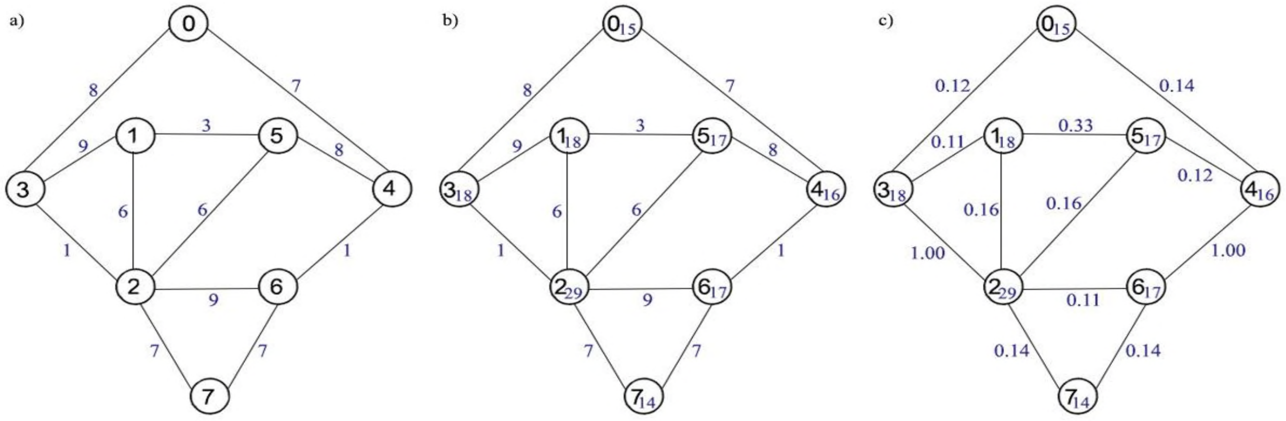
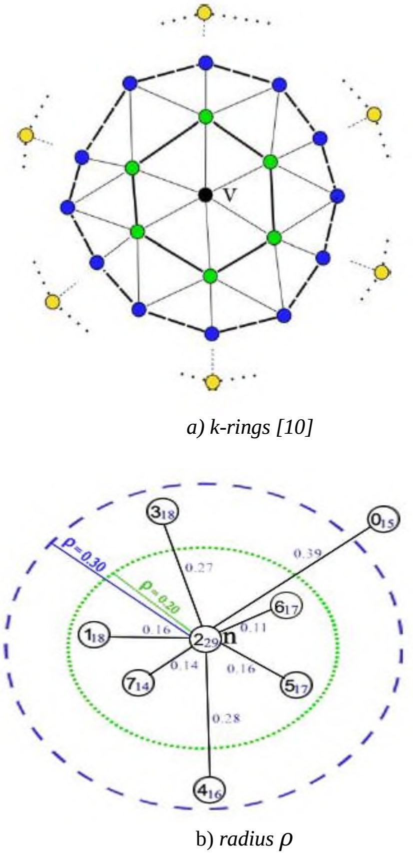


Figure 3. Weighted Graph G1



a) k-rings [10]

b) *radius* ρ

Fig. 4. Comparison between a K-ring centered on a vertex v of a 3D model triangular mesh, and a K-ring in a document, with a neighborhood ρ

We defined the coefficient of C as:

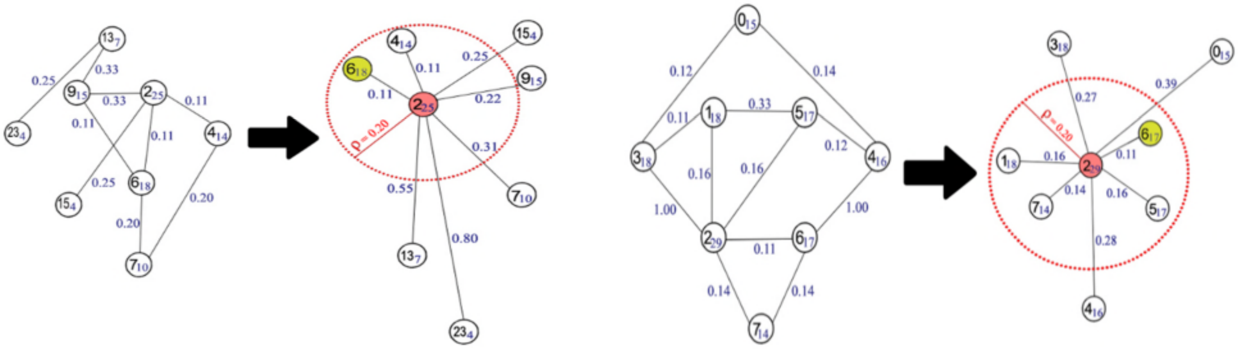
$$C = |R| + |L_{kw}| \tag{7}$$

where $|R|$ represents the importance of the relation between words and $|L_{kw}|$represents the importance of the individuality of these. Finally, the coefficient of similarity S is defined as:

$$S = \begin{cases} \text{if C} = 0\,, \ \text{inf} \\ \text{if C} > 0\,, 1/C \end{cases} \tag{8}$$

V. EXPERIMENTS AND RESULTS

The experiments were conducted using the *Reuters-21578* database, from which we chose the documents of the top 10 categories. The categories and the number of documents per category used in the experiments are listed in Table I. These documents were preprocessed according to the subsection IV-A of Section IV. Then, for each document, the corresponding graph was created, as indicated subsection IV-B. Finally, after applying the graph comparison method proposed in subsection IV-C, we made comparisons between the graphs to obtain a similarity matrix, to which we applied a minimum spanning tree algorithm to detect the groups with similar documents.



a) *Graph* G_1

b) Graph G_2

Figure 5. $F_{0.02}(2) = \{6\}$

TABLE I. TOP 10 REUTERS-21578 CATEGORIES

Category	Number of Documents
acq	2131
corn	209
crude	512
earn	3754
grain	529
interest	391
money-fx	603
ship	277
trade	450
wheat	264

In addition, once we get the minimum spanning tree, we take all pairs of adjacent nodes of the tree, and we value the categories they have in common. If two nodes have a category *Z* in common, then both belong to this category *Z*. Finally, the groups generated with the real categories of the Reuters-21578 database are contrasted.

In Table II, we can observe the results in each experiment. In the table, we can appreciate that the experiment with the less number of keywords kw = *5* and the less number of radius ρ=1, column 3 in the table, obtains the worst results. On the other hand, incrementing the number of kw to 10, and the radius ρ to 2, improves the percentage of documents correctly classified. However, if we perform a high increment in the kw number, for example, 15 and the radius ρ of 2, the percentage of success decreases, as we can see in column 7 of Table II.

In Fig. 6, we can observe the results of applying the minimum spanning tree algorithm to the matrix of document similarity. Different colors represent different categories.

VI. CONCLUSIONS

In this research, we presented an algorithm for document similarity based on concepts from the non-rigid three-dimensional model analysis. The proposed algorithm presented average results of 85% to 90% correctly classified documents. Nevertheless, when considering a major number of keywords and radius, the quality of the documents properly classified decreases; this can be because of the size of the text. In Reuters-21578 database, the size of the text is small and thus, the number of keywords and the neighborhood radius must also be short. On the other hand, as the radius increments, the number of representative key points decreases, which has an adverse effect on the document classification.

In future works, the comparisons will be made with other techniques using large text databases, because it is expected that with larger amount of text, the detection of keywords will be improved. Finally, it is possible to join different nodes from the document graph where each node represents equal words (synonyms), in order to improve the process of comparison.

REFERENCES

[1] R. Angelova and G. Weikum, “Graph-based text classification: learn from your neighbors”. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval* , pp. 485-492. ACM, 2006.

[2] E. Boyer, A. M. Bronstein, M. M. Bronstein, B.Bustos, T. Darom, R. Horaud, I. Hotz, Y. Keller, J. Keustermans, A. Kovnatsky, et al. Shrec 2011: robust feature detection and description benchmark. *arXiv preprint arXiv:1102.4258*, 2011.

[3] C. J. L. Del Alamo, L. A. R. Calla, and L. J. F. Perez, “Efficient approach for interest points detection in non-rigid shapes,” in *Computing Conference (CLEI), 2015 Latin American*, pp. 1-8. IEEE, 2015.

[4] S. Dominich. *Mathematical foundations of information retrieval*, vol. 12. Springer Science & Business Media, 2012.

[5] H. Dutagaci, C. P. Cheung, and A. Godil. “Evaluation of 3d interest point detection techniques via human-generated ground truth”. *The Visual Computer*, 28(9):901-917, 2012.

[6] B. Liu, C. W. Chin, and H. T. Ng. “Mining topic-specific concepts and definitions on the web”. In *Proceedings of the 12th international conference on World Wide Web*, pp. 251-260. ACM, 2003.

[7] Y. Ni, Q. K. Xu, F. Cao, Y. Mass, D. Sheinwald, H. J. Zhu,

and S. S. Cao. “Semantic documents relatedness using concept graph representation”. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, pp.635-644. ACM, 2016.

[8] M. T. Pilehvar and R. Navigli. “From senses to texts: An all- in-one graph-based approach for measuring semantic similarity”. *Artificial Intelligence*, 228:95-128, 2015.

[9] M. F. Porter. “An algorithm for suffix stripping”. *Program*, 14(3):130-137, 1980.

[10] I. Sipiran and B. Bustos. “Harris 3d: a robust extension of the harris operator for interest point detection on 3d meshes”. *The Visual Computer*, 27(11):963-976, 2011.

[11] W. Wang, D. B. Do, and X. Lin. “Term graph model for text classification”. In *Advanced Data Mining and Applications*, pp.19-30. Springer, 2005.

TABLE II. EXPERIMENT RESULTS WITH ROUNDED PERCENTAGE, USING DIFFERENT NUMBER OF KEYWORDS, DEGREE OF ADJACENCY AND RADIUS

Category	**Total**	**kw = 5** ρ **= 1**	**kw = 5** ρ **= 3**	**kw = 10** ρ **= 1**	**kw = 10** ρ **= 2**	**kw = 15** ρ **= 4**
acq	2131	87%	91%	91%	92%	91%
corn	209	74%	79%	78%	78%	80%
crude	512	88%	90%	88%	90%	93%
earn	3754	97%	98%	98%	98%	98%
grain	529	88%	92%	91%	91%	92%
interest	391	84%	86%	85%	87%	86%
money-fx	603	90%	92%	90%	92%	90%
ship	277	78%	81%	78%	81%	87%
trade	450	87%	90%	88%	90%	90%
wheat	264	82%	83%	84%	81%	83%

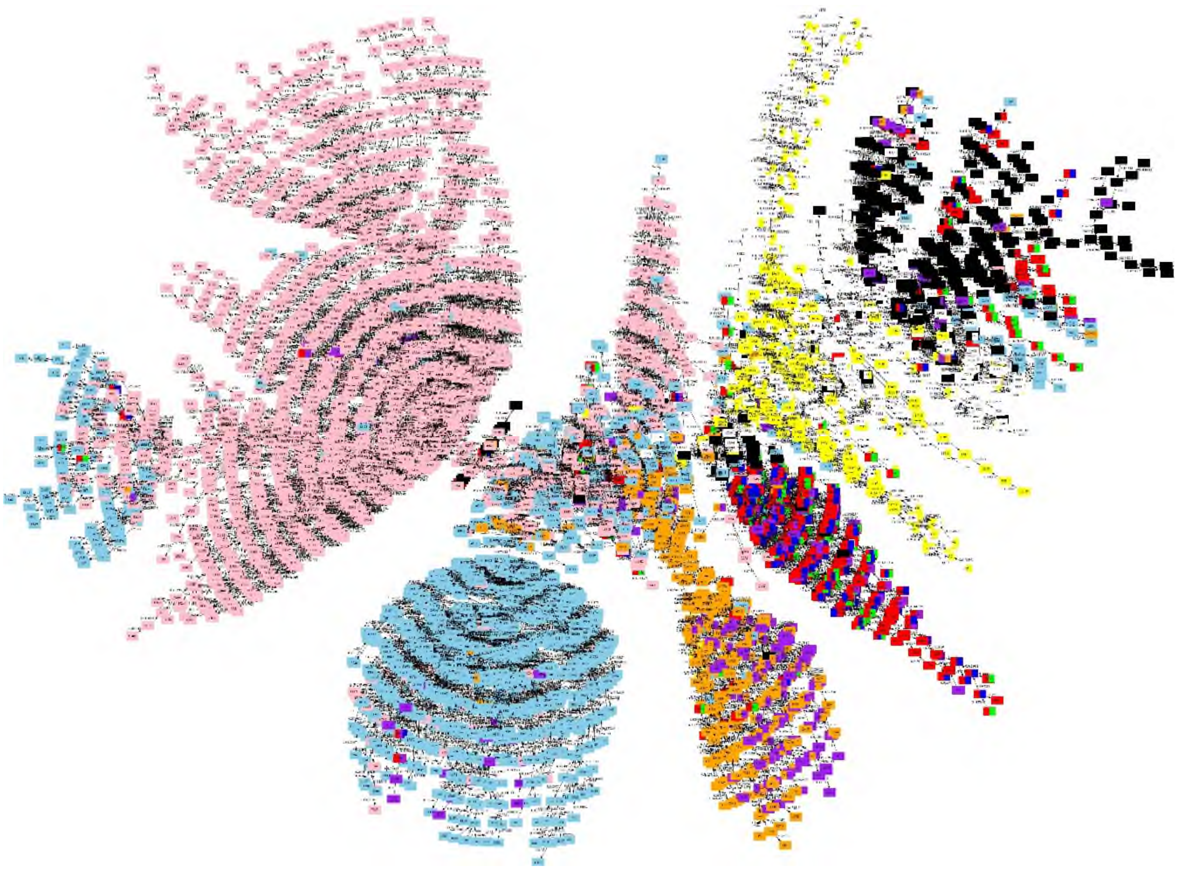


Figure. 6. Minimum spanning tree of results after applying the Kruskall algorithm in the matrix of results with test of *kw* = 5, $\rho = 3$ and adjacency *k* = 1, each color represents one of the 10 categories.