Document Classification Method based on Graphs and Concepts of Non-rigid 3D Models Approach

Lorena Castillo Galdos[1], Cristian Lopez Del Alamo[2], Grimaldo Dávila Guillén[3]
Escuela Profesional de Ciencia de la Computación
Universidad Nacional de
San Agustín de Arequipa
Arequipa, Peru

***Abstract*—Text document classification is an important research topic in the field of information retrieval, and so it is how we represent the information extracted from the documents to be classified. There exists document classification methods and techniques based on the vector space model, which doesn't capture the relation between words, which is considered of importance to make a better comparison and therefore classification. For this reason, two significant contributions were made, the first one is the way to create the feature vector for document comparison, which uses adapted concepts of non-rigid 3D models comparison and graphs as a data structure to represent such documents. The second contribution is the classification method itself, which uses the representative feature vectors of each category to classify new documents.**

***Keywords*—Document classification; graphs; non-rigid 3D models; Universidad Nacional de San Agustín de Arequipa (UNSA)**

I. Introduction

Nowadays with the increase of the use of technology, a great amount of textual information is generated as well as the need of innovative methods and techniques for its analysis, comparison, and classification, being the latter defined as the assignment of a category to an unclassified document finding similarities between this and the documents of the different known categories.

There is a wide variety of document classification algorithms, plenty of them are based on similarity comparison techniques [1], whether they are based on the vector space model [2] which treats words independently and does not capture the semantic relations between documents; or methods that do consider them important, which create graphs from the relation between words inside a document [3], [4], [5], [6].

The efficiency of these methods depends mainly of the representation of the documents to be classified, so in this paper it was decided to follow the path of [5] in the utilization of graphs as structure to represent said documents.

Graphs are data structures that are used to represent complex non-structured information about entities and the interaction between them. On the other hand, documents can also be represented as graphs using account concepts of frequencies and relationship between words. Finally, this graph can be used to apply techniques similar to that of three-dimensional meshes for classification.

Also, other areas in computer science can provide some applicable ideas and concepts to the information retrieval field, for example the approach in which this method is basing, uses adapted notions of the area of computer graphics to do a better document similarity comparison, resembling the definition of isomorphism with document semantic similarity. This method takes into account both the individuality and the relations between words, which is used for the document classification since the documents belonging to one category have a very high similitude between one another because when talking about the same topic, exists a very high quantity of words that appear in many documents inside this category, just like consecutive words, which will be detailed in Section IV.

In this paper we propose two significant contributions, the first one is the modification of the work of [5] to obtain feature vectors and the second is the classification method itself, which is based on the obtaining of representative feature vectors per category.

The general objective of this work is to develop a new method of document classification, based on rigid models analysis concepts in geometry processing. The steps to follow are these:

- Select documents to create the training and testing sets.
- Adapt the document comparison approach proposed by [5] to obtain a feature vector representing each category.
- Analyze the new document to obtain its feature vector.
- Apply the proposed classification method to the feature vector of the new document using the feature vectors of all the categories.
- Identify the category the new document belongs to.
- Experiment with the testing set.

This rest of the paper is organized as follows. Section II presents previous concepts. Section III provides an overview of the state of the art. Section IV describes the methodology. Section V evaluates experimental results and we present conclusions on Section VI.

II. Previous Concepts

For a better understanding of the problem and the proposed solution, we define the following concepts.

- **Keypoint**: In 3D models, a *keypoint* is a point which is distinctive in its locality and it is present at all different instances of the object [7].
- **Keyword**: The *keywords* of a document are defined as the words which bring most information about a set of words

inside a neighborhood. Such that, its frequency and the grade in which it is related to its neighbor words are high [5].

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

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

Where $C(v', v)$ is the shortest path from vertex v' to v and $\mid C(v', v) \mid$ is the size of the path $C(v', v)$. It is important to mention that the size of an edge is always 1 [8].
Then we adapted the concept of *k-ring* so that in documents it is called *neighborhood*.

- **Document graph**:
According to the work of [5], a document graph $G(N, A, W)$ is a representation in which the vertexes N are the terms of a document, the outgoing edges A of each node represent the existing relations between them, while W are the weights of the edges which indicate the importance of a relation. Fig. 1 shows an example of a document graph.

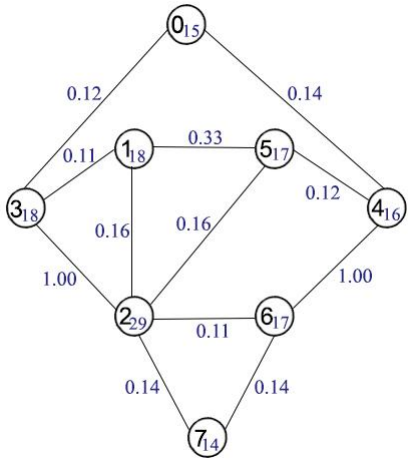


Fig. 1. Example of a Document Graph. [5]

III. State-of-the-Art

In the past years, the document classification task has been widely studied, including approaches of *machine learning* like Bayesian classifiers, Decision trees, K-nearest neighbor (KNN), Support Vector Machines (SVMs), Neural Networks, [9], [10], [11], [12], [13], [14], among others.

This paper focuses on supervised classification since it requires a learning or training process by the classifier. The main idea of supervised classification techniques or algorithms is to build a pattern from each class or category to then find the similitude between this and the new document to be classified.

To perform a better text document classification these can be represented in plenty of ways, this is done to reduce their complexity and to make them easier to handle. The more commonly used representation is the *vector space model* [2], in which the documents are represented as vectors of words. This model does not capture the relationships among words, or the semantic relations between them, for this reason, there exist methods of term weighting a matrix as it is shown in Fig. 2 [15]. A big problem of this representation is that because each entry represents a word of the document, and not all the words appear in every document to be classified, this becomes highly dimensional resulting in a very large disperse matrix [15].

$$\begin{pmatrix} T_1 & T_2 & \dots & T_{at}c_i \\ D_1 & w_{11}w21 & \dots & w_{t1}c_1 \\ D_2 & w_{12}w22 & \dots & w_{t2}c_2 \\ \vdots & \vdots & & \vdots \\ D_n & w_{1n}w2n & \dots & w_{tn}c_n \end{pmatrix} \tag{2}$$

Likewise, documents can be represented using structures like graphs, which demonstrate to better capture the relations among words or terms according to the edges between its vertexes. There are several related works that use this representation [5], [1], [3], [16], [17], [18], [4].

A. Subgraphs and Term Graphs

In the work of [17], they state that a document D_i is represented as a vector of terms $D_i =< w_{1i}, \dots, w_{|T|i} >$ where T is the ordered set of terms that appear at least once in a document inside a collection of documents. Each weight w_{ij} represents how much a term t_j contributes to the semantic of the document. The weight of each term inside a collection of documents is found by building a term graph. The relations between terms are captured using the frequent itemset mining method[1].

In the work of [3], they also use a graph-based approach to classify documents. Their algorithm W-gSpan (weighted subgraph mining algorithm) is applied to identify the subgraphs with frequent weights of the documents, these subgraphs are then used to generate a set of binary feature vectors (one per document), which then serve as entry to the TFPC classifiers (a mining classification association rule), Naive Bayes and decision tree classifier C4.5 showing as a result, a percentage greater than 84% of classification precision using two methods described as follows.

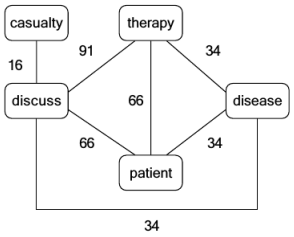
The first classification method consists in treating each term of a graph as a web page to find a PageRank score, which is a method that consists in the idea that if a web page is pointed by several other web pages, then its ranking will be high, or if pages with a high score point to it. Then a rankings vector representing the document is created, and the category whose ranking co-relation coefficients (found with the Spearman algorithm) are higher with this test document is assigned. This vector is used with SVM, obtaining an average of 92% of precision.

The second method is based on the term distance matrix and the distance-weight similarity function. Given a distance matrix set $\{T_1, T_2, \dots, T_n\}$ representing the categories $\{C_1, C_2, \dots, C_n\}$ and a test document D, the document will be classified into the category C_i if and only if the distance-weight similarity of C_i and D is the longest among all the

[1]These algorithms can be used to find subsets of items that surpass a threshold inside a collection.

Itemset	Support
$\{therapy, discuss\}$	91
$\{therapy, discuss, patient\}$	66
$\{therapy, discuss, patient, disease\}$	34
$\{casualty, discuss\}$	16

(a) Frequent Itemsets



(b) The Corresponding Graph

Fig. 2. Example of a Term Graph, in a) the Frequent Itemsets are shown, and in b) its Corresponding Graph [17].

categories. This method obtained an average precision of more than 60%.

Also, there exists other methods that combine this subgraph and term graph approaches to perform the classification task [19], [20], [21].

B. Graphs and Graph-Kernels

In the work of [18], they consider the text classification task as a graph classification problem, model text documents as a graph-of-words, which correspond to a graph in which vertexes represents unique terms of the document and the edges represent co-occurrences between the terms inside a fixed size window. An example of this graph is shown in Fig. 3.

Then, they used linear SVMs to perform the classification because the objective was discovering and exploring new characteristics. To perform the characteristics extraction the used gSpan (graph-based Substructure pattern) to get frequent subgraphs, the minimum quantity of these depends of a parameter known as support, the optimal value of this parameter can be learned trough cross-validation to maximize the prediction precision of the classifier, turning this whole process in a supervised process. When reducing the graphs, it is necessary to keep the more dense parts for which they extract its main cores. This method obtained results of up to more of 90% of precision.

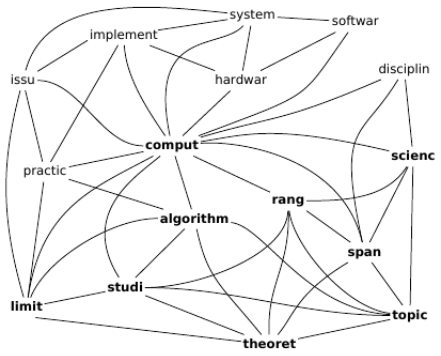


Fig. 3. Graph of Words, Bold Words Represent the Words of the *Main Core* [18].

Author in [4] present a similarity measure based in the definition of a graph-kernel[2] between pairs of documents, using the terms contained in the documents and the relations among them, representing them as a graph-of-words. Specifically they capitalize on the kernel and modify it to compare the graph representations of a pair of documents.

The method takes as entry a pair of documents and automatically computes how similar are one of another based only on their content. This method was tested by doing text categorization for which a SVM classifier was used taking as entry the kernel matrix of the training set, showing results of up to 77% of precision in one database and more than 91% in the other three.

IV. Methodology

Due to the wide variety of techniques that have been developed to solve the document classification problem, is that this paper adopts the innovative approach of Lorena *et al.* [5] of document similarity comparison using graphs and concepts of 3D models and applies it towards document classification.

At the time of obtaining feature vectors from the document graphs, what we look for is to capture better the relation among words inside a document, extracting this way a semantic representation of it, to then be able to use them with the classification method.

A general diagram of the document classification process for a new unclassified document is shown in Fig. 4.

As it was mentioned previously, this paper modifies a previous work approach. Then, its general functioning is explained as well as the modification to obtain the feature vectors and the classification method. The steps performed are enumerated according to Fig. 4.

a) **Preprocessing and graph construction**:
For the preprocessing phase, first we do the cleaning step, which consists in the elimination of stop words[3]. Then the Porter algorithm is applied for the stemming step, which preserves only the roots of the words to avoid the different time, gender and number variations; and because there will be repeated roots we proceed to the ID Assignment step, which assigns numeric IDs to each root, to later be inserted on the list L.

[2]Graph-kernels can be intuitively understood as functions that measure the similarity of pairs of graphs.

[3]They can be pronouns, articles, etc.

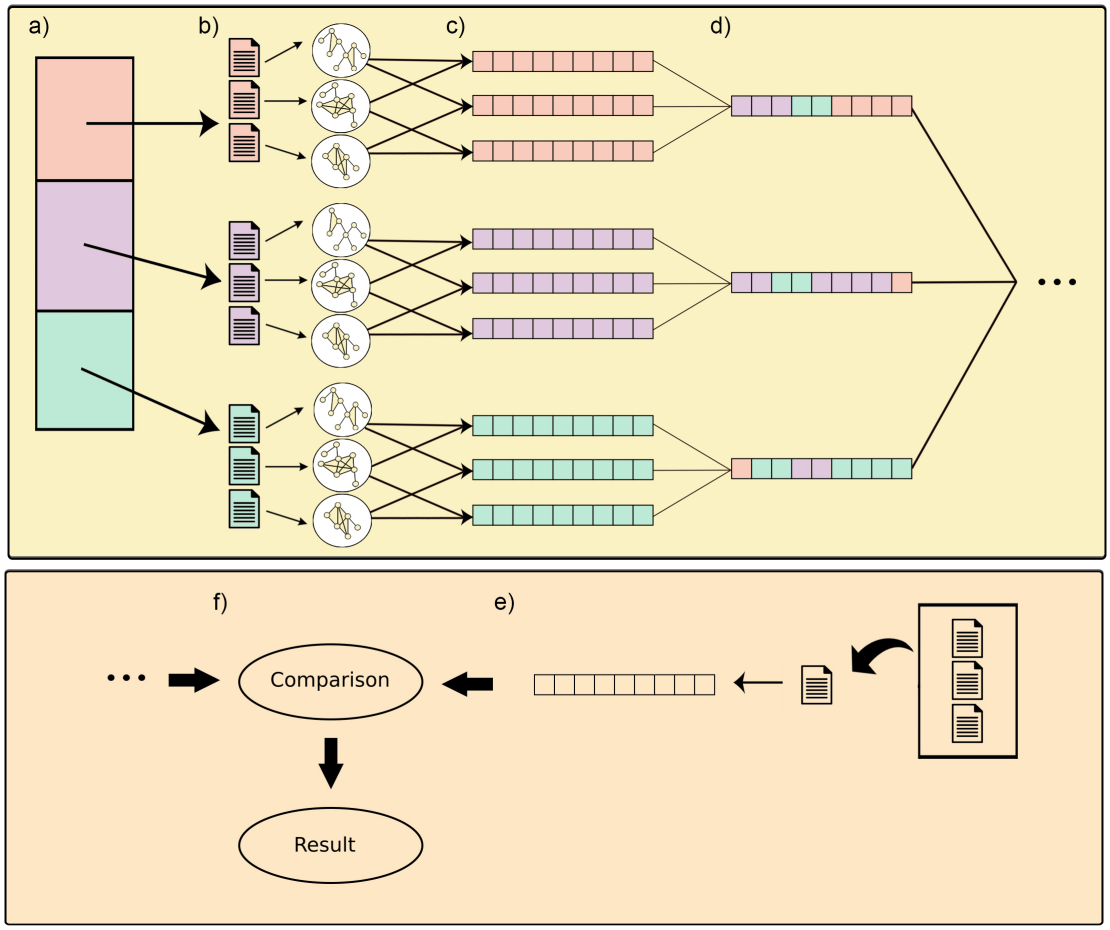


Fig. 4. Pipeline of the Proposed Model.

b) **Graph construction** After the preprocessing step, we proceed to build the graph $G(N, A, W)$ where N are the nodes of the graph, which represent the elements of the list L, A indicates the edges which are the existing relations between the elements of the list L, and W are the weights of the edges. The protruding edges of the nodes represent the grade in which these are related with their neighbors, as it is shown in Fig. 1.

c) **Comparison**:
Following the approach of [5], to perform the comparison between two document graphs G_1 and G_2, first we obtain a list of keywords (L_{kw}) of each graph, which are the μ nodes with greater weights. Then we found a list with the intersection of both lists, which will contain the common keywords between both graphs as it is shown in Equation 3.

$$KW(G_1, G_2) = \max_{\mu}(G_1) \cap \max_{\mu}(G_2) \tag{3}$$

Where $\max_{\mu}$ represents 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. Given that w is the number of times that a relation between two words (a, b) appears on the text, to find the distance between the nodes that represent these words the Equation 4 is applied.

$$D_{a,b} = \{\frac{1}{w_{a,b}}\} \tag{4}$$

Then we use the Equation 5 to find the neighborhood.

$$R = \{F_{\rho}(L_{kw_1}) \amalg \cdots \amalg F_{\rho}(L_{kw_{|L_{kw}|}})\} \tag{5}$$

Where $F_{\rho}(L_{kw_j}) = \{n \in G_1, G_2 : D(n, L_{kw_j}) \leq \rho\}$, D denotes the shortest distance between the node n and L_{kw_j} applying the Dijkstra algorithm, n are all the nodes which distance D is shorter than a radio ρ.

Subsequently, instead of obtaining a comparison coefficient per each pair of documents as the authors do in [5], we perform the comparison between them following the Equation 6, obtaining the comparison vectors B which are the union of the keywords in common plus the neighborhood of these, keeping like this more information than just a coefficient. This is performed for every document inside each category.

$$B = R \cup L_{kw} \tag{6}$$

TABLE I. TABLE OF CLASSIFICATION PERCENTAGES WITH 4 CATEGORIES.

	Method 1				Method 2			
threshold ϕ>3000								
$kw = 15$, $\rho = 2$, grade $k = 2$								
baby	73	0.75	5.75	20.5	89.25	0.5	1.25	9
dvd	2.75	80	4	13.25	7.25	81.25	4.5	7
software	3.75	4	78.25	14	6.5	2.5	75.25	15.75
toys_&_games	10.25	5	4	80.75	34	2.25	2.25	61.5
$kw = 10$, $\rho = 3$, grade $k = 2$								
baby	73.25	2	2.25	22.25	85	2.25	1.25	11.5
dvd	2	86.5	1.25	10.25	3.75	89.5	1.25	5.5
software	4.75	7	72	16.25	6	6	71.5	16.5
toys_&_games	9.25	6	1.5	83	32.25	5	2	60.75
threshold ϕ>10000								
$kw = 15$, $\rho = 2$, grade $k = 2$								
baby	66	1.5	8.25	24.25	67.75	1.25	8.75	22.25
dvd	2.25	80	3.75	14	2.25	70.25	9.5	18
software	5.25	6	73.75	15	6	1.25	76.5	16.25
toys_&_games	8	4.75	4.75	82.5	6.75	2	5.75	85.5
$kw = 10$, $\rho = 3$, grade $k = 2$								
baby	64.25	3	3.25	29.5	67	3.5	3.25	26.25
dvd	2	84.75	1.25	12	2.25	78.5	4	15.25
software	7	10	65.25	17.75	14.25	6.25	57	22.5
toys_&_games	7	6.5	2.5	84	7.25	3	2.5	87.25

d) **Feature vectors per category**

To obtain the representative vectors $\Gamma_1, \Gamma_2, \dots \Gamma_n$ where n is the number of categories, we considered to apply the intersection of the vectors B; this concept was initially considered to obtain the common IDs of all vectors, but because of the low probability of a word being considered a keyword and also appear in every document inside a category, this idea was dismissed, also because in the experimentation step the results of the intersection came to be 0 or the size of the resulting vector was too small.

Instead of this, we obtain the occurrence frequencies δ of each word of the dictionary of all vectors B. This frequencies vector is then ordered in a decreasing way to obtain the words with higher frequencies according to a threshold ϕ which is passed by parameter.
Finally, the resulting vectors Γ are obtained using the Equation 7, each vector will represent a category and will contain the IDs of their more representative words.

$$\Gamma = B_1\{id_{i=1}, \dots, id_\phi\} \cup \dots \cup B_n\{id_{i=1}, \dots, id_\phi\} \quad (7)$$

Where n is the number of the obtained feature vectors.

e) **Feature vector of a new document**

In order to obtain the feature vector Z of a new document, first we do the preprocessing and graph obtaining steps. Then, each ID will be placed as a position of the vector as it is shown in Equation 8 to then perform the classification method.

$$Z = \{id_1, id_2, \dots, id_t\} \quad (8)$$

Where t is the total number of obtained IDs of the document.

f) **Classification Method**

Once obtained the vector Z of the new document and the representative vectors Γ of each category, we find the intersection of this vector with all the vectors Γ, to get this way the belonging grade X with each category. Then, to obtain X two methods are proposed.

a) **Method 1**: X is the number of elements of the intersection between Z and Γ.

$$X = \sum_{i=0}^{n(Z \cap \Gamma)} 1 \quad (9)$$

b) **Method 2**: X is the sum of the frequencies of the words in Γ that are in the intersection with Z.

$$X = \sum_{i=0}^{n(Z \cap \Gamma)} \delta(Z \cap \Gamma)_i \quad (10)$$

By last, the category to which the new document will belong to, will be the one with which it obtained the higher belonging grade X.

V. EXPERIMENTS AND RESULTS

For the experimentation phase, we used the amazon database [22], from which we randomly chose 4 categories and 8000 documents, being 2000 per category. The set of documents was then divided in 1600 training documents and 400 testing documents. After this the next steps were performed:

First we get the vectors B from the training set. Then, by analyzing the obtained results we can assign the value of the threshold ϕ, which controls how many IDs will be extracted for the classification method. The results of the category vectors B showed values of δ superior to 3000 and 10000 becoming these the assigned values to the threshold ϕ to then get the representative vectors per category Γ.

Next, in the Tables I and II, the values of the diagonals represent the percentage of correct classified documents as well as the error percentage, this is to say, documents assigned

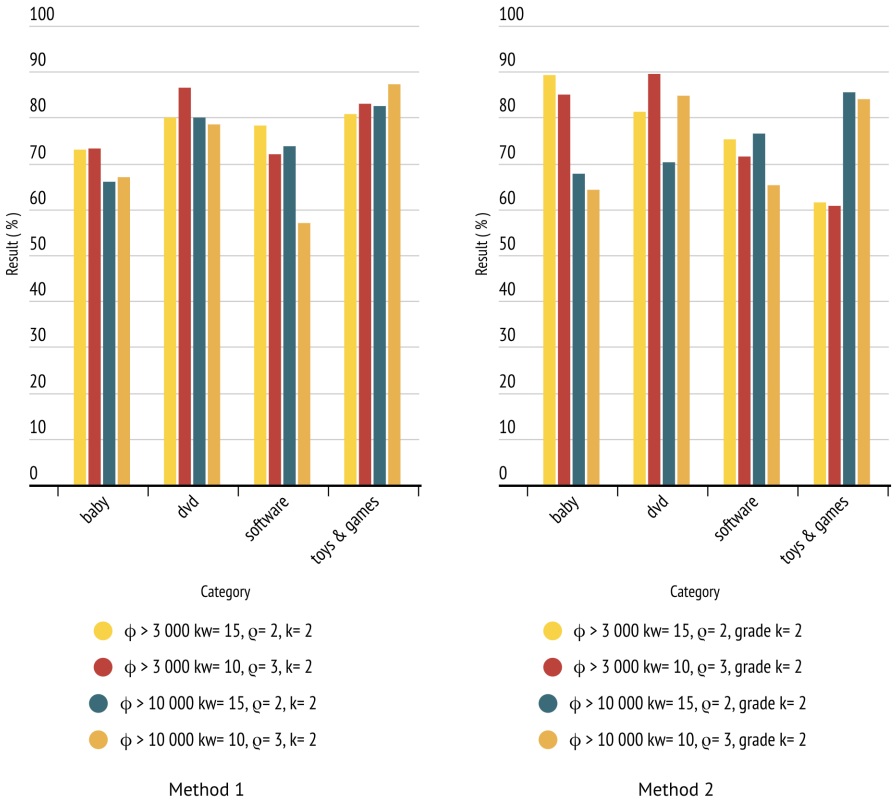


Fig. 5. Bar Chart of the Percentages of the Correctly Classified Documents in Table I, using Method 1 and 2, where each Bar represents a Different Experiment.

TABLE II. TABLE OF CLASSIFICATION PERCENTAGES WITH 3 CATEGORIES.

	Method 1			Method 2		
threshold ϕ>3000						
$kw = 15$, $\rho = 2$, grade $k = 2$						
baby	92	1.5	6.5	97.25	0.75	2
dvd	5.5	89.25	5.25	8.75	86	5.25
software	10	6.25	83.75	10.75	6	83.25
$kw = 10$, $\rho = 3$, grade $k = 2$						
baby	94.5	2.75	2.75	94.25	3	2.75
dvd	3.5	95	1.5	4	94	2
software	10.5	11.75	77.75	9	11.25	79.75
threshold ϕ>10000						
$kw = 15$, $\rho = 2$, grade $k = 2$						
baby	87.25	2	10.75	80	3.25	16.75
dvd	4.5	89.5	6	2.75	80.5	16.75
software	11	7.5	81.5	8.5	5.5	86
$kw = 10$, $\rho = 3$, grade $k = 2$						
baby	90	5.75	4.25	86.25	7	6.75
dvd	3.5	94.25	2.25	2.75	89.25	8
software	12.5	14	73.5	17.5	13.25	69.25

to an incorrect category; for this we assigned different input parameters like $\rho = 2$ and $\rho = 3$, keywords number of $kw = 15$ and $kw = 10$, and grade $k = 2$.

Fig. 5 and 6 show the bar charts of the percentages of correctly classified documents, which are shown in the diagonals of the Tables I and II. We can observe that in Fig. 5, the results achieved using Method 1 with different input parameters ϕ, ρ, and k, tend to have less variation between them in most categories in comparison with the results obtained with Method 2. We can note that this behavior persists if we vary the number of categories, as showed in Fig. 6.

Also, in Fig. 5 we can see that the results of Method 2 were higher in some experiments in comparison to Method 1, these results vary if we change the input parameters, for example, the results of the category *baby* differ from 67% up to 89.25% as shown in Table I.

VI. CONCLUSIONS

In this paper, we presented a text document classification method based on a similarity comparison approach, which adapts concepts taken from the analysis of non-rigid tridimensional models and uses graphs as the structure to represent

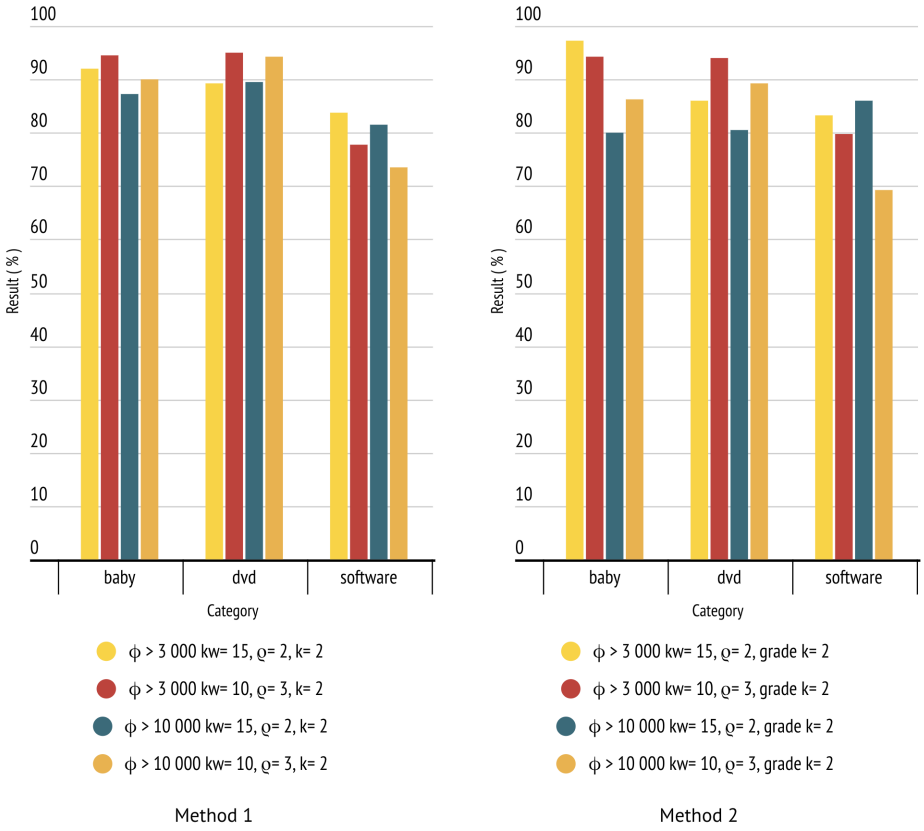


Fig. 6. Bar Chart of the Percentages of the Correctly Classified Documents in Table II, using Method 1 and 2, where each Bar represents a Different Experiment.

such documents. The method proved to have average results from 75.5% up to 78.6% of correctly classified documents with 4 categories and 82.1% up to 89.3% with 3 categories. We can observe that when performing the experiments without the category of toys toys_&_games, which generated a higher error percentage, the percentage of correct classified documents increases.

Furthermore, by the time of getting the comparison vectors B per category, their sizes can be different as well as the size of the representative vector Γ, because unlike the vector space model, in this method it would not be necessary to complete elements inside these vectors to unify their sizes according to the dictionary of words.

Worth noting that the obtained words in this representative vectors Γ are those who keep more information about the category.

ACKNOWLEDGMENT

The authors would like to thank the support and subvention of UNIVERSIDAD NACIONAL DE SAN AGUSTÍN DE AREQUIPA with contract No TP-7-2018-UNSA.

REFERENCES

[1] Z. Wu, H. Zhu, G. Li, Z. Cui, H. Huang, J. Li, E. Chen, and G. Xu, "An efficient wikipedia semantic matching approach to text document classification," *Information Sciences*, vol. 393, pp. 15–28, 2017.

[2] G. Salton, A. Wong, and C.-S. Yang, "A vector space model for automatic indexing," *Communications of the ACM*, vol. 18, no. 11, pp. 613–620, 1975.

[3] C. Jiang, F. Coenen, R. Sanderson, and M. Zito, "Text classification using graph mining-based feature extraction," *Knowledge-Based Systems*, vol. 23, no. 4, pp. 302–308, 2010.

[4] G. Nikolentzos, P. Meladianos, F. Rousseau, Y. Stavrakas, and M. Vazirgiannis, "Shortest-path graph kernels for document similarity," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 1890–1900.

[5] L. Castillo G., G. Dávila G., and C. López Del Alamo, "A new graph-based approach for document similarity using concepts of non-rigid shapes," pp. 41–46, june 2017.

[6] C. Paul, A. Rettinger, A. Mogadala, C. A. Knoblock, and P. Szekely, "Efficient graph-based document similarity," in *International Semantic Web Conference*. Springer, 2016, pp. 334–349.

[7] H. Dutagaci, C. P. Cheung, and A. Godil, "Evaluation of 3d interest point detection techniques via human-generated ground truth," *The Visual Computer*, vol. 28, no. 9, pp. 901–917, 2012.

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

[9] G. N. Chandrika and E. S. Reddy, "An efficient filtered classifier for classification of unseen test data in text documents," in *2017 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*. IEEE, 2017, pp. 1–4.

[10] L. Ge and T.-S. Moh, "Improving text classification with word embedding," in *Big Data (Big Data), 2017 IEEE International Conference on*. IEEE, 2017, pp. 1796–1805.

[11] A. Jain and J. Mandowara, "Text classification by combining text classifiers to improve the efficiency of classification," *International Journal of Computer Application (2250-1797)*, vol. 6, no. 2, 2016.

[12] R. Jindal and S. Taneja, "Ranking in multi label classification of text documents using quantifiers," in *Control System, Computing and Engineering (ICCSCE), 2015 IEEE International Conference on*. IEEE, 2015, pp. 162–166.

[13] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," *arXiv preprint arXiv:1607.01759*, 2016.

[14] Y.-S. Lin, J.-Y. Jiang, and S.-J. Lee, "A similarity measure for text classification and clustering," *IEEE transactions on knowledge and data engineering*, vol. 26, no. 7, pp. 1575–1590, 2014.

[15] V. Korde and C. N. Mahender, "Text classification and classifiers: A survey," *International Journal of Artificial Intelligence & Applications*, vol. 3, no. 2, p. 85, 2012.

[16] K. R. Gee and D. J. Cook, "Text classification using graph-encoded linguistic elements." in *FLAIRS Conference*, 2005, pp. 487–492.

[17] W. Wang, D. B. Do, and X. Lin, "Term graph model for text classification," in *International Conference on Advanced Data Mining and Applications*. Springer, 2005, pp. 19–30.

[18] F. Rousseau, E. Kiagias, and M. Vazirgiannis, "Text categorization as a graph classification problem," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, vol. 1, 2015, pp. 1702–1712.

[19] K.-A. Sohn, T.-S. Chung *et al.*, "A graph model based author attribution technique for single-class e-mail classification," in *Computer and Information Science (ICIS), 2015 IEEE/ACIS 14th International Conference on*. IEEE, 2015, pp. 191–196.

[20] F. D. Malliaros and K. Skianis, "Graph-based term weighting for text categorization," in *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015*. ACM, 2015, pp. 1473–1479.

[21] B. Li, Q. Yan, Z. Xu, and G. Wang, "Weighted document frequency for feature selection in text classification," in *Asian Language Processing (IALP), 2015 International Conference on*. IEEE, 2015, pp. 132–135.

[22] J. Blitzer, M. Dredze, and F. Pereira, "Biographies, bollywood, boomboxes and blenders: Domain adaptation for sentiment classification," in *Proceedings of the 45th annual meeting of the association of computational linguistics*, 2007, pp. 440–447.