ISSN 2395-8618

Business Process Models Clustering Based on Multimodal Search, K-means, and Cumulative and No-Continuous N-Grams

Hugo Ordoñez, Luis Merchán, Armando Ordoñez, and Carlos Cobos

Abstract—**Due to the large volume of process repositories, finding a particular process may become a difficult task. This paper presents a method for indexing, search, and grouping business processes models. The method considers linguistic and behavior information for modeling the business process. Behavior information is described using cumulative and no-continuous n–grams. Grouping method is based on k-means algorithm and suffix arrays to define labels for each group. The clustering approach incorporates mechanisms for avoiding overlapping and improve the homogeneity of the created groups using the K-means algorithm. Obtained results outperform the precision, recall and F-measure of previous approaches.**

Index Terms—**Clustering, business process models, multimodal search, cumulative and no-continuous n-grams.**

I. Introduction

BUSINESS processes (BP) are composed of related and structured activities or tasks that contribute to a business goal. Consequently, BP models allow representing and documenting and sharing companies' internal procedures. These models may be useful also to guide the development of new products and support improvement of processes. Notwithstanding the advantages of BP models, the management of its repositories may become a big challenge. The latter is because commonly these repositories store hundreds or even thousands of BP models, that in turn are made up of tens or hundreds of elements (tasks, roles and so on) [1]. As a result, to find a particular BP matching specific requirements may become a complex task.

Most of the existing research approaches for BP search are based on typical measures such as linguistics, structure, and behavior. However, other techniques from the field of Information Retrieval (IR) may be applied to improve existing results. Among these IR techniques, the multimodal search reports good results among users; this is in part because multimodal search combines different information types to increase the accuracy of the results [2]. Moreover, clustering techniques have been used in BP search to improve the results display. Clustering techniques create groups of the BPs obtained from the query. These groups are created based on the similarity of the BPs.

This paper presents an approach for clustering of BP models based on multimodal search and cumulative and no-continuous n-grams. Cumulative and no-continuous n-grams allow us to analyze more linguistic information that traditional n-grams. These n-grams are built following a tree shaped path based on syntactical information. This method allows reviewing the branches of the tree [3]. This approach unifies linguistic and behavioral information features in one search space while takes advantage of the clustering techniques for improving results display, thus giving users an clear idea of the retrieved BP [4].

Firstly, this approach includes an indexing and search method based on a multimodal mechanism that considers two dimensions of the BP's information. 1) linguistic information which includes names and descriptions of BP elements (e.g. activities, interfaces, messages, gates, and events). And 2) behavior information represented as codebooks (text strings) which include all the structural components representing sequences of the control flow (i.e. the union of two or more components of the control flow simulates the formation of cumulative and no-continuous N-grams) [5], [6]. Secondly, the present approach includes a technique for grouping BP based on affinity. This grouping uses a clustering technique based on the both dimensions of the BP aforementioned.

The present approach is based on a multimodal mechanism previously described in [4] and introduces improvements in two areas. By using cumulative no-continuous n-grams, more elements of control flow may be represented and analyzed during the search and indexing process. In addition, the

Manuscript received on August 12, 2016, accepted for publication on October 4, 2016, published on October 30, 2016.

Hugo Ordoñez and Luis Merchán are with the Facultad de Ingeniería, Universidad de San Buenaventura, Cali, Colombia, and the Laboratorio para investigación en desarrollo de la ingeniería de software (LIDIS) of the Universidad de San Buenaventura, Colombia (e-mail: hugoeraso@gmail.com, lmerchan@usbcali.edu.co).

Armando Ordoñez is with the Facultad de Ingeniería, Fundación Universitaria de Popayán, the group Intelligent Management Systems, Popayán, Colombia, (e-mail: armandoordonez@gmail.com).

Carlos Cobos is with the Departamento de Sistemas, Facultad de Ingeniería Electrónica y Telecomunicaciones, Universidad del Cauca, Colombia (e-mail: ccobos@unicauca.edu.co).

https://doi.org/10.17562/PB-54-4

ISSN 2395-8618

clustering mechanism was improved to avoid overlapping (BP results can not belong to many groups simultaneously) and increase the homogeneity of groups. The latter improvements were achieved by implementing k-means algorithm, and by performing more iterations of the algorithm for selecting the best group. Finally clusters are tagged (labeled) based on their functionality using a Suffix Array algorithm.

The evaluation of the proposed approach was done using a BP repository created collaboratively by experts [7]. The results obtained using the present approach, were compared with the results of other state-of-the-art algorithms. This comparison was performed using measures from information retrieval domain.

The rest of this paper is organized as follows: Section 2 presents the related work, Section 3 describes the proposed approach, Section 4 presents the evaluation, and finally Section 5 describes conclusions and future works.

II. Related Works

The proposed approach is focused on two strategies: searching and clustering of BP models. This section presents main research works on both strategies.

Regarding searching, most of the existing approaches for BP search are based on measures such as linguistics, structure, and behavior. Linguistic-based approaches use, for example, the name or description of activities or events. Later during the search process, some techniques are used, such as space-vector representation with a frequency of terms (TF), and cosine similarity to generate the rankings of results [8]. Approaches based on association rules analyze previous executions of business processes using log files. During the search, activity patterns and phrases related to business process activities are identified using domain ontologies. Besides, in order to create a list of results, a heuristic component that determines the frequency of detected patterns is employed [9]. Approaches based on genetic algorithms use formal representations (graphs or state machines) of BPs and include data such as the number of inputs and outputs per node, edge labels, nodes name or description. Although this method may achieve precise results, execution time may be very high [10]. Most of the works in this area merely match inputs and/or outputs, using textual information of BP elements.

Regarding clustering, approaches may be classified into hierarchical and partitional clustering. Hierarchical clustering builds a hierarchy of groups based on structural and behavioral similarity of BP. These proposals allow users to review the hierarchy and choose a set with greater similarity according to their criteria [11], [12], [13]. Partitional clustering uses log files containing previous executions of BP. In this case the clustering algorithm groups BP with similar behavior based on the control-flow and data-flow found in their log files [14], [15], [4].

Unlike these approaches, the present approach uses a multimodal representation of BP models. Equally, clustering techniques are used to improve the results display. This clustering is based on grouping similar BP in the same group, which facilitates the display of results obtained from in a query.

III. Searching and Clustering of BP models

The main tasks of the present approach are i) Indexing (to include a new BP into the repository) and ii) search and clustering (to search BPs similar to the user query). Next, both tasks are described.

A. *Indexing task*

This task uses business rules to manage and pre-process BP models before indexing and storing in the repository. This task includes textual processing and generation of a search index. Next, the modules responsible for implementing these rules are described (Fig. 1), namely: 1) Parser and 2) Indexing and Weighting.

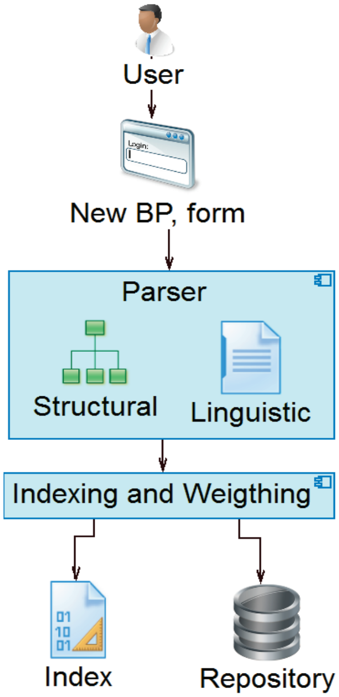

Fig 1. Indexing task: Include a new BP in the repository

Parser: The parser implements an algorithm that takes a BP described in BPMN notation and builds linguistic and structural components (codebook), this component also generates a search index consisting of two arrays by each BP model: an array MC of textual features and another *MCd* of structural components. The algorithm is described below.

Formation of linguistic component (Linguistic): Then the algorithm takes each *BPi*, extracts its textual characteristics Ct (activity name, activity type, and description) and forms a vector $Vtc_i = \{Ct_{i,1}, Ct_{i,2}, ..., Ct_{i,l}, ..., Ct_{i,L}\}$, where L is the number of textual characteristics found in BP_i. at this point, traditional pre-processing task area applied to textual components, namely, tokenize, lower case filtering, stop words removal, and stemming. For each vector Vtc_i, which represents a BP_i, a row of matrix MC_{il} is constructed. This row contains the linguistic component of all *BPs* stored in the repository. In this matrix, array *i* represents each *BP* and *l* a textual characteristic for each of them.

Formation of codebook component (Structural): A codebook *Cd* is a set of *N* structural components describing one or more nodes of the *BP* in the form of text strings. The set of codebooks formed from the whole repository is called the codebook component matrix. This matrix is formed by taking each tree that represent each *BP* in the repository. For example, Fig 2, shows a fragment of a BP_i with its activities. Each activity is represented with a text string defining the node type (StartEvent, TaskUser, TaskService). The node type refers to the functionality of each activity within the *BP*.

Codebooks are formed simulating the technique of traditional n-grams. These codebooks are sequences of textual elements: words, lexical item, grammatical labels, etc. arranged according to the order of appearance in the analyzed information. This method differs from previous works where traditional n-grams are formed with two components (N = 2, bigrams) [4]. Aditionally, in the present approach, the representation includes cumulative and no-continuous n-grams with N = 1 (unigrams), N = 2 (bigrams), N = 3 (trigrams) and so on until a maximum value of N = M. N-grams had shown to be convenient for the tree based representation of business processes. Next, a sample of BP is shown in Fig 2, and then in Table 1 the correspondence between activities of the BP in Fig. 2 and their node types are presented. Next, all codebooks for the BP are shown.

TABLE 1.
EXAMPLE OF THE ACTIVITIES OF THE BP IN FIGURE 2 AND THEIR TYPES

Activity	Type
Start	StartEvent
Evaluate clients payment	TaskUser
Route	RouteParallel
Client status report	TaskService
Recalculating debt	TaskScript

A codebook of n-grams representing the process described in Fig 2 is composed of n-grams which vary in size from 1 to 4 (M=4). n-grams of size 1 are: {*StartEvent, TaskUser, RouteParallel, TaskService, TaskService*}, on the other hand, n-grams of size 2, 3 and 4 are formed as described in Fig. 3, 4, and 5.

In Fig. 3, {StartEvent_TaskUser1, TaskUser_Route Parallel2, RouteParallel_TaskService3, RouteParallel_TaskScrip4}, where StartEvent_TaskUser1 corresponds to the concatenation of Star Event with the Evaluate clients payment activity, similarly to the other components.

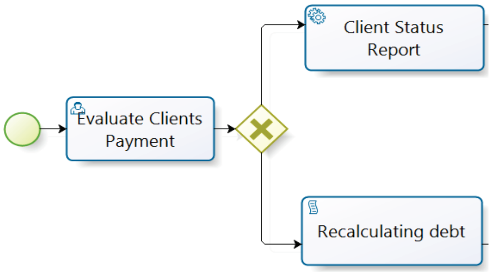

Fig 2. Types of component in Business process

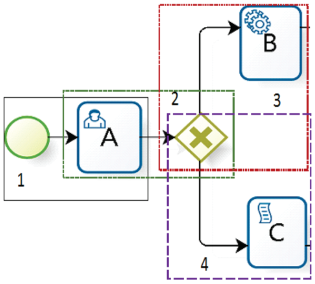

Fig. 3. Size-2 codebook

In Fig 4. {StartEvent_TaskUser_RouteParallel1, Task User_RouteParallel_TaskService2, TaskUser_RouteParallel_TaskScrip3}.

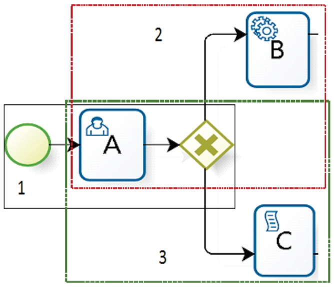

Fig 4. Size-3 codebook

As can be seen, as n-gram size grows a bigger part of the sequential path is covered (by concatenating its components), for example, in logical gates there exist bifurcation. As shown in Fig 4, in the codebook 2 the bifurcation goes from activity A to activity B, consequently, according to the property of cumulative and non continuos n-grams [3], it is possible to form the codebook 3 from activity A to Activity B.

Fig 5, {StartEvent_TaskUser_RouteParalle_Task Service1, StartEvent_TaskUser_RouteParalle_TaskScrip2}.

As can be observed, the cumulative and non- continuous n - grams can cover a significant part of the tree representing the

ISSN 2395-8618

semantic behavior of BP. The latter demonstrates that the control flow of the BP can be fully analyzed.

Unlike traditional n-grams, codebooks formed by cumulative and no-continuous n-grams provide a better and higher representation of processes control flow and behavior semantics. These codebooks allow better representation of the BPs as they are formed by joining control flow sequences. Behavior semantics of business processes describes the activities and its execution order [12]. It is important to note that as codebooks increase in size, they represent better the behavior semantics of processes.

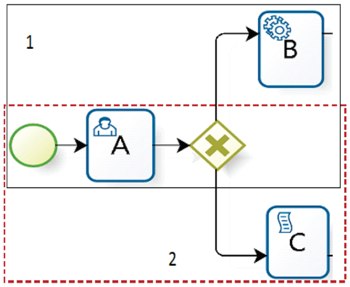

Fig 5. Size-4 codebook

Finally, the codebooks vector for sample BP of Fig. 2 is Vcd_i = {StartEvent, TaskUser, RouteParallel, TaskService, TaskService, StartEvent_TaskUser, TaskUser_RouteParallel, RouteParallel_TaskService, RouteParallel_TaskScrip, Start Event_TaskUser_RouteParalle, TaskUser_RouteParallel_Task Service, TaskUser_RouteParallel_TaskScrip, StartEvent_Task User_RouteParalle_TaskService, StartEvent_TaskUser_Route Paralle_TaskScrip}.

The cumulative and no-continuous n-grams concept can be used for terms (linguistic features) presented in BPs, but in this proposal, they just were used for the behavioral features (control flow).

Indexing and weighting: In this component, the linguistic and codebook components are weighted to create a multimodal search index *MI* composed of the matrix of the linguistic component (*MC*) and the codebook component matrix (MCd) i.e. $MI = \{MC_d \cup MC\}$. The index also saves the reference to the physical file of each of the models stored in the repository.

Weighting: Next, this component built the term by document matrix applies a weighting scheme of terms similar to that proposed in the vector space model for document representation, this approach is described elsewhere [16][17]. This weighting scheme is based on the original proposal of Salton [18].

B. Search task

This task is responsible for allowing users to conduct BP searches using three query options: linguistic, codebook, and multimodal (see Fig. 6). Each query is represented using a terms vector $q = \{t1, t2, t3, ..., t_j, ..., t_J\}$. The same pre-processing mechanism applied in the indexing task (parser) is applied to the BP query, thus obtaining the terms of the query vector reduced to their lexical root and the cumulative and continuous n-grams of the query.

Query forms: In this component, the user has forms that correspond to the graphical user interface (GUI). These forms allow selecting the search options and displaying the lists of the results and the created groups.

Conceptual ratings: This component sorts and filters the BP retrieval from the search. The ordering is done using one adaptation of the equation of conceptual score (used by Lucene library) [19].

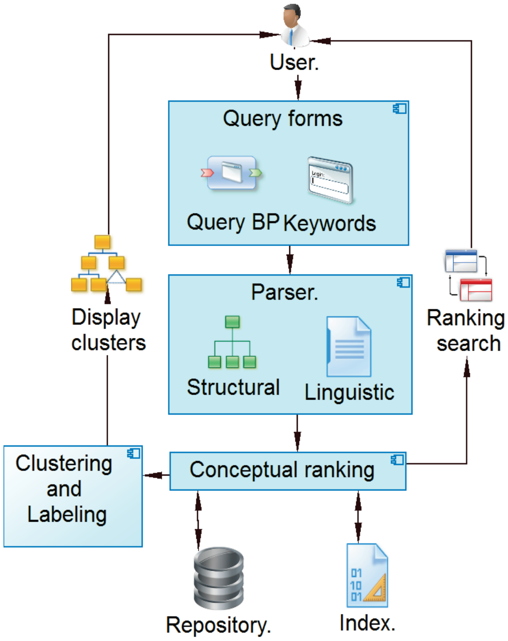

Fig 6. Searching and clustering task

List of results (Ranking): this component shows the results of the search to the user, in order to be analyzed.

Clustering process: Once the results are ranked, they are grouped using the K-means clustering algorithm [20]. Thus, the results are organized in groups of BP which are correlated according to textual and structural features.

K-means: The algorithm receives as input the number of groups (k-clusters, for performing the experiments in the assessment, we use values of k between 4 and 5 based on the recommendation presented in [13] to be formed). Then, k BPs are randomly selected to represent the starting centroids of each group. Later, each BP in the result list is assigned to the closest cluster centroid according to a distance function (where the most used one is the cosines similarity). For each one of the formed groups, the centroid of all these BP is

ISSN 2395-8618

calculated. Centroids are taken as new centers of their respective groups. The steps of the K-means algorithm are described below:

- *Step 1*: the algorithm selects k BP to be used as initial centroids (k is the number of groups to be formed);
- *Step 2*: each BP is added to the group with the highest similarity or proximity;
- *Step 3*: the algorithm calculates the centroid of each group to be new centroids.
- *Step 4:* if a convergence criterion is not reached return to step 2. For example, if the classification of BP is not changed.

Labeling: most of the clustering algorithms create groups without labels that allow identifying its content. Conversely, the present approach adopted a labeling method based on Suffix Arrays to determine the content of each group created (i.e., related to the purpose or functionality of BP models) and to ease user's interaction with the results. Thus, users may get a better idea of groups to review.

The labeling process starts creating a snippet (*S*) using tasks names. These tasks describe functionalities of BP models that compose the group to be labeled. Subsequently, chain *S* is pre-processed and converted to lowercase. Later, special characters and empty words are removed from *S*. Finally, an array of suffixes *As* is created. This array is ordered lexicographically to find most common phrases in *S* that identify the group content.

In the labeling algorithm, *S* is processed as a character set $S = \{s_1, s_2, s_3, s_n\}$. From this set, a new set $S'[i\ ,j]$ is formed, i.e., an array of sub-strings of S, which runs from index *i* to index *j*. After that, an array of integers *As* is created containing initial positions of suffixes in *S* ordered lexicographically. Then, *As*[*i*] stores the starting position of the *i*-th smallest suffix in S. Afterwards, the array of substrings *S′* is traversed using a binary search that aims to find the most common and with higher length suffix. The search starts with a separator of terms (in the case we have used the character $) and the subsequently found suffix is returned for labeling the BP group.

Display clusters: This component displays the formed groups in organized and structured way. This structure enables users to review and select the group with higher similarity with the query.

IV. Evaluation and Results

Results obtained using the present approach, were compared with the results of the manual evaluation performed on a closed test set, which is presented in [7]. This closed test set was created collaborative by 59 experts in business process management. In addition, the results of the present clustering multimodal (from now on *N-gramClusterBP*) approach were also compared with the results of grouping of the *MultiSearchBP* model [21] (from now on *LingoBP*) and *BPClustering* for grouping [22] (from now on *HC*). *LingoBP* uses two component, firstly a multimodal search based on by-grams (n = 2) and then clustering based on the Lingo algorithm. *HC* uses cosine coefficient to measure the similarity between two process models and implements an agglomerative hierarchical algorithm for clustering.

The evaluation was conducted in two phases: 1) internal assessment and 2) external assessment.

The first phase involves the application of internal metrics for clustering analysis that do not require human intervention. These metrics are used to identify how close or distant BPs are from each other in the formed groups. The used metrics are described below.

Sum of squares Between clusters (*SSB*)*:* this measures the separation between clusters (high values are desired). In Equation 3, *k* is the number of clusters, n_j is the number of elements in the cluster *j*, c_j is the centroid of cluster *j, and x* is the mean of the data set [23]:

$$SSB = \sum_{j=1}^{k} n_j \; dist\ (c_j \; - \; \bar{x})^2. \tag{1}$$

Sum-of-squares within cluster (SSW): this measures the variance (low values are desired) within groups, based on each of the existing elements in each group [23]:

$$SSW = \sum_{i=1}^{k} \sum_{x \in c_i} dist\ (m_i, x)^2, \tag{2}$$

where *k* is the number of clusters, *x is* a point in the cluster c_i and m_i is the centroid from cluster c_i.

Table 2 shows the results of the internal evaluation. Regarding *SSB*, *N-gramClusterBP* reached an average value of 0.510. This result evidence the high separation of the created groups since the elements are assigned to the cluster having higher similarity and the intermediate elements between groups are removed. *N-gramClusterBP* outperforms *HC* in 0.09 and outperforms *LingoBP* in 0.14. Regarding *SSW*, elements variation between groups created using *N-gramClusterBP* is low. This good result show that BPs in the same group share similarly textual and structural features.

TABLE 2.
Results of internal assessment of the grouping

Algorithm	SSB	SSW
N-gramClusterBP	**0,510**	**0,048**
LingoBP	0,350	0,070
HC	0,420	0,065

The second phase, external assessment is focused on the quality of clustering by comparing groups created by automatic grouping techniques with groups generated by domain experts.

ISSN 2395-8618

In this phase, metrics such as weighing precision, weighing recall, and weighing F-measure were used. To evaluate weighing precision (Equation 3), weighing recall (Equation 4) and weighing F-measure (Equation 5), the groups' set $\{C_1, C_2, \ldots, C_k\}$ automatically created with evaluated approaches were compared with the ideal groups' collection $\{C_1^i, C_2^i, \ldots, C_h^i\}$ generated collaboratively by experts [24]. During assessment, the following steps were performed: (a) for each group C_n^i in the ideal set, a group C_m was found in the automatically generated set which most closely approximates to the first group. Later, the following metrics are calculated: $P(C, C^i)$, $R(\mathrm{C}, \mathrm{C}^{\mathrm{i}})$ and $F(C, C^i)$ as defined in Equations 6, 7 and 8; (b) to calculate the weighting precision, weighting recall and weighting F-measure based on Equation 8.

$$P(C, C^i) = \frac{|C \cap C^i|}{|C|} \tag{3}$$

$$R(C, C^i) = \frac{|C \cap C^i|}{|C^i|} \tag{4}$$

$$F(C, C^i) = \frac{2P(C, C^i)R(C, C^i)}{P(C, C^i) + R(C, C^i)} \tag{5}$$

$$P = \frac{1}{T} \sum_{j=1}^{h} |C_j^i| \, P(C_m, C_j^i) \tag{6}$$

$$R = \frac{1}{T} \sum_{j=1}^{h} |C_j^i| \, R(C_m, C_j^i) \tag{7}$$

$$F = \frac{2PR}{P+R} \; ; \; T = \sum_{j=1}^{h} |C_j^i| \tag{8}$$

In Equation 8, C is a group of BP models, C^i is a group from the ideal set. Fig 7 shows the average values of Precision, Recall, and F-Measure for the assessment of groups created using of *N-gramClusterBP, LingoBP* and *HC*.

Regarding the precision, best results were achieved with *N-gramClusterBP*. This algorithm increases precision by 16% compared with (*LingoBP*) and 12% compared to *HC*. This result is due to the high number of similar elements of the control flow and textual information that can be found both in the groups generated using *N-gramClusterBP* and in the ideal set. Moreover, the combination of structural and textual information used in *N-gramClusterBP* allows creating groups with greater similarity with the groups created by experts. The latter occurs because human experts consider several data types existing in the *BPs*.

On the other hand, groups formed by the (*LingoBP*) contain shared *BPs*, i.e., *BP* that belong to various groups, which increases the number of Falses Negative (NF) (*BPs* placed in groups different to the one that was expected) and consequently the *Pg* was reduced. Regarding *HC*, the values are explained by the fact that only structural information was used during the formation of groups. Besides, this formation of groups is done statically, that is, one *BP* is assigned to one group and cannot be assigned to another group with higher similarity in a posterior iteration.

Regarding Recall, *N-gramClusterBP* increases Recall by 5% in comparison to *LingoBP* and 3% in comparison to *HC* (*N-gramClusterBP* 47%, *LingoBP* with 44% and *HC* with 45%). The Recall value reached shows that more elements in the groups created with *N-gramClusterBP* were placed in the same groups that the manual (ideal) grouping. The latter can be explained by the absence of overlapping (*BPs* existing in many groups simultaneously) in *N-gramClusterBP*. As a result, *N-gramClusterBP* approach reduces the false negatives (FN) as the groups are assigned to the groups with higher similarity. Conversely, in *LingoBP* the number of elements per group decreases the value of true positives (TP) (*BPs* in the same group which was created by the manual grouping). Regarding F-measure, *N-gramClusterBP* (57%) achieves 12% more than *LingoBP* and 9% more that *HC*. The latter allows inferring that the created groups are more relevant and similar to groups created manually by the experts.

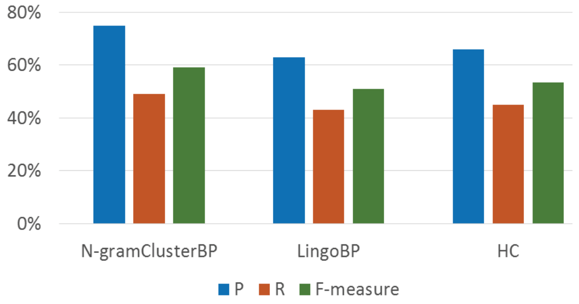

Fig 7. Results in the external clustering evaluation

V. Conclusions and Future Work

This paper presents an approach for improving the recovery, and clustering of business processes (BP) presented in [4]. The presented approach uses a multimodal search method based on cumulative and no-continuous n-grams of behavioral (structural) features. The use of textual information and structural information in multimodal index offers greater flexibility and precision in queries.

Results of the internal assessment show that using textual and structural information offers more compact BP groups because elements in the same group share diverse features. Moreover, by eliminating the overlapping (BP Models that may exist in several groups at the same time), N-gramClusterBP creates groups with more similar elements and also provides greater separation between the created groups.

The grouping process using N-gramClusterBP showed a high similarity (75% of precision) with the grouping performed by experts. This similarity is higher that the similarity achieved by LingoBP (63%) and HC (66%). This result can be explained by the absence of overlapping (elements in many groups simultaneously) and high refinement of groups (by performing iterations for assigning the most similar group) in N-gramClusterBP.

Future work includes adding specific domain ontologies to the proposed model; this will make possible including semantics to the search process, achieving more precise results. Equally, future work will be focused on the

ISSN 2395-8618

assessment of labeling method to determine if created labels help users to identify more easily information and functionality in the created groups. Finally, a hierarchical clustering method will be incorporated to create categories and subcategories of existing BP models in the repository.

REFERENCES

[1] M. La Rosa, “Detecting approximate clones in business process model repositories,” *Information Systems*, vol. 49, pp. 102–125, 2015.

[2] J. C. Caicedo, J. Ben Abdallah, F. A. González, and O. Nasraoui, “Multimodal representation, indexing, automated annotation and retrieval of image collections via non-negative matrix factorization,” *Neurocomputing*, vol. 76, pp. 50–60, Aug. 2012.

[3] G. Sidorov, “N-gramas sintacticos no-continuos,” *Polibits*, no. 48, pp. 69–78, 2013.

[4] H. Ordoñez, J. C. Corrales, and C. Cobos, “Business Processes Retrieval Based on Multimodal Search and Lingo Clustering Algorithm,” *IEEE Latin America Transactions*, vol. 13, no. 3, pp 769–776, 2015.

[5] G. Sidorov, *Construcción no lineal de n-gramas en la lingüística computacional*, Mexico DF: Sociedad Mexicana de Inteligencia Artificial, 2013.

[6] G. Sidorov, F. Velasquez, E. Stamatatos, A. Gelbukh, and L. Chanona-Hernández, “Syntactic N-grams as machine learning features for natural language processing,” *Expert Systems with Applications*, vol. 41, no. 3, pp. 853–860, 2014.

[7] H. Ordoñez, J. C. Corrales, C. Cobos, L. K. Wives, and L. Thom, “Collaborative Evaluation to Build Closed Repositories on Business Process Models,” *ICEIS 2014, Proceedings of the 16th International Conference on Enterprise Information Systems*, vol. 3, SciTePress, pp. 311–318, 2014.

[8] A. Koschmider, T. Hornung, and A. Oberweis, “Recommendation-based editor for business process modeling,” *Data Knowl. Eng.*, vol. 70, no. 6, pp. 483–503, 2011.

[9] D. A. Rosso-Pelayo, R. A. Trejo-Ramirez, M. Gonzalez-Mendoza, and N. Hernandez-Gress, “Business Process Mining and Rules Detection for Unstructured Information,” *MICAI 2010*, *Ninth Mex. Int. Conf. Artif. Intell.*, IEEE, pp. 81–85, 2010.

[10] C. J. Turner, A. Tiwari, and J. Mehnen, “A genetic programming approach to business process mining,” *Proc. 10th Annu. Conf. Genet. Evol. Comput. GECCO 2008*, p. 1307, 2008.

[11] C. Diamantini, D. Potena, and E. Storti, “Clustering of Process Schemas by Graph Mining Techniques” (extended abstract), *Methodology*, vol. 4, p. 7, 2011.

[12] J. Melcher, D. Seese, and I. Aifb, “Visualization and Clustering of Business Process Collections Based on Process Metric Values,” *Measurement*, vol. 8, p. 9, 2008.

[13] W. Sheng, X. Liu, and M. Fairhurst, “A Niching Memetic Algorithm for Simultaneous Clustering and Feature Selection,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 7, pp. 868–879, 2008.

[14] D. R. Ferreira, “Applied Sequence Clustering Techniques for Process Mining,” *Handbook of Research on Business Process Modeling*, IGI Global, pp. 481–502, 2009.

[15] D. Ferreira, M. Zacarias, M. Malheiros, and P. Ferreira, “Approaching Process Mining with Sequence Clustering: Experiments and Findings,” *Engineering*, vol. 7, no. 1, pp. 1–15, 2008.

[16] A. Ordonez, H. Ordonez, C. Figueroa, C. Cobos, and J. C. Corrales, “Dynamic reconfiguration of composite convergent services supported by multimodal search,” *Lecture Notes in Business Information Processing*, 2015, vol. 208, pp. 127–139.

[17] C. Figueroa, H. Ordoñez, J.-C. Corrales, C. Cobos, L. K. Wives, and E. Herrera-Viedma, “Improving Business Process Retrieval Using Categorization and Multimodal Search,” *Knowledge-Based Syst.*, vol. 110, pp. 1–17, 2016.

[18] Christopher D. Manning, Raghavan, Prabhakar, Schütze, *Introduction to Information Retrieval,* Cambridge University Press, 2008.

[19] Y.-C. Hu, B.-H. Su, and C.-C. Tsou, “Fast VQ codebook search algorithm for grayscale image coding,” *Image Vis. Comput.*, vol. 26, no. 5, pp. 657–666, May 2008.

[20] T. Handhayani and L. Hiryanto, “Intelligent Kernel K-Means for Clustering Gene Expression,” *Procedia Comput. Sci.*, vol. 59, pp. 171–177, 2015.

[21] H. Ordoñez, J. C. Corrales, and C. Cobos, “Business Processes Retrieval based on Multimodal Search and Lingo Clustering Algorithm,” *IEEE Lat. Am. Trans.*, vol. 13, no. 9, pp. 40–48, 2015.

[22] L. L. Jae-Yoon Jung, Joonsoo Bae, “Hierarchical clustering of business process models,” *Eng. Inf. Syst. Control*, vol. 5, no. 12, pp. 613–616, 2009.

[23] Q. Zhao and P. Fränti, “WB-index: A sum-of-squares based index for cluster validity,” *Data Knowl. Eng.*, vol. 92, pp. 77–89, 2014.

[24] H. Ordonez, J. C. Corrales, C. Cobos, and L. K. Wives, “Collaborative grouping of business process models,” pp. 1–2, 2014.