

## GENERATING CORPUS FOR EVALUATING PERFORMANCE OF PROCESS MATCHING TECHNIQUES

AYESHA ASMAT\*<sup>1</sup>, AFNAN IFTIKHAR

<sup>1</sup>Department of Computer Science, University of Management and Technology, Lahore, Pakistan

<sup>2</sup>Department of Informatics and Systems, University of Management and Technology, Lahore, Pakistan

Email: ayesha.asmat@umt.edu.pk

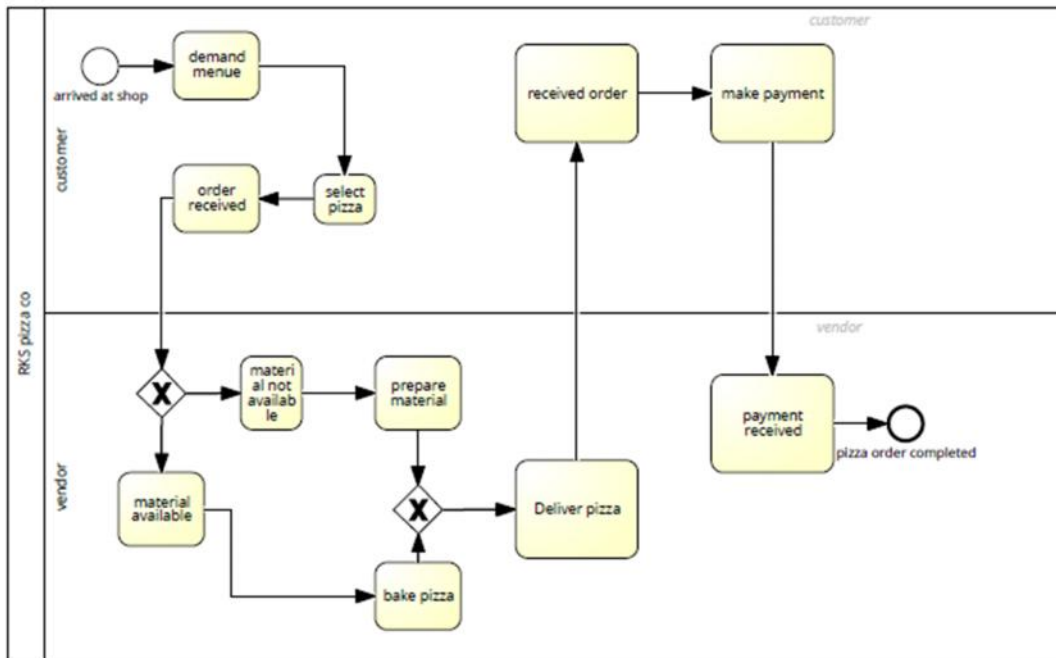
**ABSTRACT.** *Business process management (BPM) plays a vital role in organizations management. A central piece to that is the collection of business process models. Depending upon the size of the organization, the collection may have many process models in the process repository. A key feature to such a repository is searching of process models which requires computing similarity between a pair of process models. For a given pair of process models, similarity refers to finding whether the two process models that form the pair are similar or not. To compute the similarity between process models, several techniques have been established however a rigorous evaluation of these techniques has either not been conducted on numerous occasions or the evaluation has not been sufficiently rigorous. A key reason to that is the absence of benchmark set of queries and their relevant process models, as judge by human experts. In this study, we argue, the fewer queries used for evaluation may not have the necessary diversity to challenge the abilities of the matching techniques. This work is usually not done due to large number of manual comparisons. It is thus required pool of queries. A related challenge is to identify a pool of process models that are declared as relevant to the query models. To address these challenges, we have suggested a technique.*

**Keywords:** Business Process Model; Similarity between Process Models; Manual Annotations.

**Introduction.** Business Process Management (BPM) is an emerging field that deals with the effective and efficient management of business processes [1]. It includes methods, concepts, and techniques to manage design and administrate business processes [2], where a business process contain interrelated set of activities to achieve a certain goal[1,2]. Process models are integral part of the organization as they are used to organize their processes. A bus process is a group of activities executed in a sequence to achieve some specific task, service or end product. The specification of a business process can be represented in two ways: graphical model and in textual form. To represent process model in graphical way different process modeling languages are used. These languages are graphical where elements of the language are graphical signs. The popular languages include, BPMN [3], Petri nets [4] and YAWL [5] etc. Figure-1 shows an example process model representing pizza order process model. In Figure-1 a process model is drawn in BPMN-the de facto standard process modeling language. The process model contains one start event, eleven activities, two XOR gateways, and one end event.

To generate textual description of process model Natural Language Generation System<sup>1</sup> (NLGS) tool is used. NLGS [6] accept input in JSON file and generate textual description of process model. Figure-2 represents textual description generated by NLGS.

<sup>1</sup>Available for download at: <http://www.henrikleopold.com/downloads/>.



**Figure-1:** Pizza order process model

**Figure-2:** Textual Description of Pizza order process model

The process begins when the customer demands a menu. Then, it selects the pizza. Afterwards, the customer orders the received. Subsequently, is.

- The vendor conducts the material available. Then, the vendor bakes the pizza.
- The vendor conducts the material not available. Afterwards, the vendor prepares the material.

Once was the vendor delivers the pizza. Subsequently, it received the order. Then, the customer makes the payment. Afterwards, the vendor conducts the payment received. Subsequently, the process is finished.

Process repository of any organizations contains a number of processes which may lead to thousands of process models [7] and several stakeholders would like to use, update and retrieval process model from a process repository. This may require an abstract representation of process models, as research shows such abstractions for other entities for instance webs services [8]. This makes searching, similarity computation, and possibly ranking between processes models a key feature of such as repository [9]. Various techniques/metrics [10, 11, 12] have been developed to compute similarity between process models, however, a rigors evaluation of these approaches has not been conducted.

The main objectives of this study are to select an appropriate set of queries and identify the process models that are relevant to the query process models, from the collection of process models, as judged by humans. In this article we proposed a technique to develop queries from larger dataset and to find their relevant process models. In this research, we randomly choose queries and, to find out the relevant process model a dataset of 150 process models were pre-processed which were then manually compared against queries. The rest of the paper is organized as follows: Section 2 introduces the background, Section 3 presents the methodology for developing corpus, Section 4 presents the results of the study, and Section 5 presents the Conclusion.

**2. Background.** The number of process matching techniques [10, 11, 12] have been developed to compute the similarity between process models. However, the evaluation of the effectiveness of these approaches generally involves few queries from a smaller dataset. The key reason to the use of a small number of queries in numerous studies is, it requires generating human judgments which is a time consuming and resource intensive task. For instance, if we have, 10 queries and 100 process models, it would require 1000 ( $100 * 10$ ) manual comparisons and several hundred activity pairs. For instance, if we want to compare two process models we will have to compare their activities, flow between activities, process output and the activities of the process. Furthermore, if a single person is involved in generating human judgments, there are chances of individual biasness as the person interprets the process differently and made different decisions. Precision, recall and f-measure were used for measuring effectiveness of any process matching technique. While calculating precision, number of process models retrieved were considered relevant if they were related to the ones mention relevant by human annotators. The relevant processes are determined by human beings who manually declare process models as relevant or irrelevant. Due to that reason, these are also called manual annotations or human annotations.

There are several issues that need to be considered while generating human annotations. Some of them are, a) generating relevant criteria b) relevance procedure c) human annotators. To declare candidate and query process model similar it is required to create relevance criteria that is used by annotators, relevance criteria is necessary for finding similarity between process models. To have consistency and correctness of results, criteria should be defined sharply using which humans can declare process models similar or dissimilar. It is because in the absence of such criteria may lead to the results that are not repeatable.

**3. Methodology.** The dataset for this research was generated with the help of 3 researchers which resulted in 150 different process models. The models were drawn in Signavio<sup>2</sup> – a process model tool and their JSON files are also available. These JSON files were given as input to Natural Language Generation System (NLGS) [6] to create the textual description of these models. These process models were from different categories such as admission, quality assurance processes, reservation/booking processes, website sign up processes, feedback process, SAT test etc. From this dataset 10 queries were randomly selected.

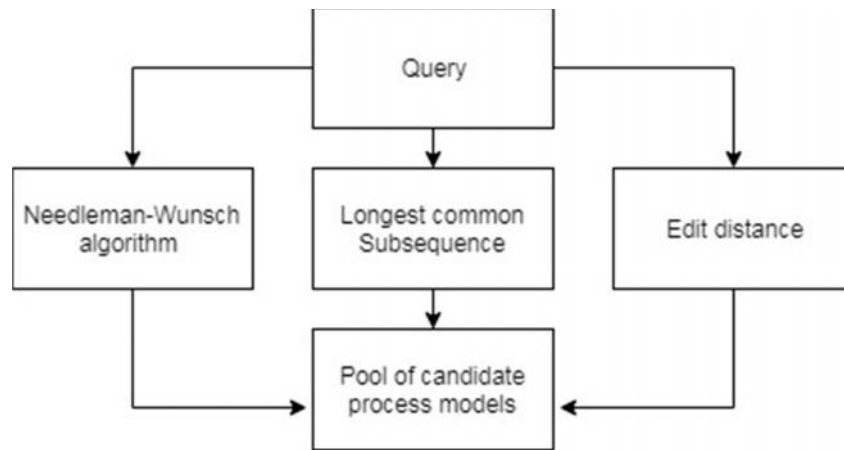
In this research, our focus was to develop a technique to retrieve the process models relevant to the queries from the dataset. So that this technique can be used on larger dataset. The main challenge while retrieving relevant process models against selected 10 queries there was a need to compare each process model against 150 models. It resulted in 1,500 comparisons that is a human intensive task.

To reduce the comparisons, only top files from a text matching techniques are selected in the pool. The files that are not in these top files are supposed to be irrelevant. *Figure - 3* represents our complete methodology. Text matching algorithms such are Needleman-Wunsch algorithm (global alignment) [13], Longest Common Subsequence (LCS) [14] and Edit Distance [15] were implemented and executed. We first computed the similarity score for 10 queries against 150 process models and then sort the similarity result of queries in descending order based on similarity scores. After that, top files were selected and top 20 results were chosen for each query from 3 matching algorithms.

Each Query is compared with the selected top files. These top files are manually compared by the annotators who had a complete knowledge of business process models. In order to avoid the biasness in relevant model selection two annotators performed the comparison task. For the manual comparison models were abstractly compared, i.e. is the query and the candidate process model from the same subject and domain? For example: if one model is admission related and other relates to health care so they both are irrelevant. This level further reduced the models and resulted in approximately relevant domain. After this the process model activities were compared to find there relevance. If the activates matches 50%, they were declared as similar.

---

<sup>2</sup> Available at: <https://www.signavio.com/>



**Figure - 3:** Process for selecting candidate process models

**Results.** The discussed methodology is applied on source-query pairs. For the comparison of each pair (query-source process pair), the criteria was to be judge from top to bottom i.e. first process level and then element level. If a pair fulfills process level criteria, the element level criteria should be screened otherwise it should be declared as irrelevant, without assessing the element level criteria. Table 1 represents the sample sheet for source-query comparison for query P114.

**Table-1:** Sample sheet for manual comparison

Query Process ID	Candidate Process ID	C1	Similar Activities	Similar Output	Similarity Result
P114	P6	N			N
P114	P12	N			N
P114	P13	N			N
P114	P16	N			N
P114	P17	N			N
P114	P37	Y	Y	Y	Y
P114	P68	Y	Y	Y	Y
P114	P75	Y	N	Y	N
P114	P147	N			N

In table 1 column C1 represents process level criteria. For element level criteria, we screened both points of element level criteria, if both points of element level criteria fulfilled then both processes are declared the same. Two annotators compare 600 pairs of the process model and the conflicts are resolved by the 3<sup>rd</sup> annotator. 3<sup>rd</sup> annotator result is declared final. After manual comparison by 3 annotators, final manual comparisons against each query is generated. As the judgment was performed by two annotators independently. Both annotators discussed the criteria and perform human judgments independently. Kappa Static for 600 query-source pairs was calculated. Kappa static represents inter-rater agreement between annotators. The Table 2 below shows the kappa statistics score 0.96.

**Table-2:** Kappa Static

		Annotator 2			
		0	1		
Annotator 1	0	558	12	570	95.00%
	1	5	25	30	5.00%
		563	37	600	
		93.83%	6.17%		

Kappa Static 0.96

After applying the discussed methodology, the final queries and their corresponding manual annotation are represented in table 3.

**Table-3:** Annotations for Queries

Query	Manual Annotations
P1	P1, P74, P16
P14	P14
P37	P37, P114
P66	P12, P66, P72
P67	P40, P67, P111
P71	P71, P126
P99	P83, P99
P114	P37, P68, P114
P121	P94, P121, P133
P148	P140, P144, P148

Above table display the 10 selected queries and manual annotations of these queries. These manual annotations can be used to test any technique. We have two contributions in this research, we generate manual annotations for the dataset and develop a technique for generating manual annotations for any dataset.

**5. Conclusion.** Business process are integral part many organizations and organizations have growing number of process models in their repositories. There may be 100s or 1000s of process models in a process repository. The management of repositories requires effective search techniques. Matching processes is required for several reasons for example if we want to identify common or similar processes between consolidated companies. Various techniques/metrics has been developed to find similarity between process models. However, rigours evaluation of these approaches has not been conducted. In this work we generate queries and manual annotations for the dataset of 150 process models. Benefit of this research is that we also developed a technique to generate manual annotations for any dataset. These queries and manual annotations can be used for testing any technique.

#### REFERENCE

- [1]. Houy, C., Fettke, P., & Loos, P. (2010). Empirical research in business process management – analysis of an emerging field of research. *Business Process Management Journal*, 16(4), 619–661. <https://doi.org/10.1108/14637151011065946>
- [2]. Weske, M. (2010). *Business Process Management: Concepts, Languages, Architectures* (1st ed.). Springer Publishing Company, Incorporated.
- [3]. White, S. A. (2008). *BPMN Modeling and Reference Guide: Understanding and Using BPMN*. Future Strategies Inc.

- [4]. AjmoneMarsan, M., Balbo, G., Conte, G., Donatelli, S., &Franceschinis, G. (1995). Modelling with generalized stochastic Petri nets.
- [5]. van der Aalst, W. M. P., &terHofstede, A. H. M. (2005). YAWL: Yet Another Workflow Language. *Inf. Syst.*, 30(4), 245–275. <https://doi.org/10.1016/j.is.2004.02.002>
- [6]. Leopold, H., Mendling, J., &Polyvyanyy, A. (2012). Generating Natural Language Texts from Business Process Models. In *Proceedings of the 24th International Conference on Advanced Information Systems Engineering* (pp. 64–79). Berlin, Heidelberg: Springer-Verlag. [https://doi.org/10.1007/978-3-642-31095-9\\_5](https://doi.org/10.1007/978-3-642-31095-9_5)
- [7]. Phios Process Repository. (n.d.). Retrieved March 9, 2018, from <http://process.mit.edu/>
- [8]. Ceri, S., Abid, A., Helou, M. A., Barbieri, D., Bozzon, A., Braga, D., ... Valle, E. D. (2010). Search Computing: Managing complex search queries. *IEEE Internet Computing*, 14(6), 14–22.
- [9]. Antunes, G., Bakhshandeh, M., Borbinha, J., Cardoso, J., Dadashnia, S., Francescomarino, C. D., ...Ghidini, C. (2015). The process model matching contest 2015. *Enterprise Modelling and Information Systems Architectures*.
- [10]. Dijkman, R., Dumas, M., &García-Bañuelos, L. (2009). Graph Matching Algorithms for Business Process Model Similarity Search. In *Proceedings of the 7th International Conference on Business Process Management* (pp. 48–63). Berlin, Heidelberg: Springer-Verlag. [https://doi.org/10.1007/978-3-642-03848-8\\_5](https://doi.org/10.1007/978-3-642-03848-8_5)
- [11]. Weidlich, M., Dijkman, R., &Mendling, J. (2010). The ICoP Framework: Identification of Correspondences Between Process Models. In *Proceedings of the 22Nd International Conference on Advanced Information Systems Engineering* (pp. 483–498). Berlin, Heidelberg: Springer-Verlag. Retrieved from <http://dl.acm.org/citation.cfm?id=1883784.1883832>
- [12]. Dijkman, R., Dumas, M., van Dongen, B., Käärik, R., &Mendling, J. (2011). Similarity of Business Process Models: Metrics and Evaluation. *Inf. Syst.*, 36(2), 498–516. <https://doi.org/10.1016/j.is.2010.09.006>
- [13]. Luu, V.-T., Ripken, M., Forestier, G., Fondement, F., & Muller, P.-A. (2016). Using Glocal Event Alignment for Comparing Sequences of Significantly Different Lengths (Vol. 9729, pp. 58–72). [https://doi.org/10.1007/978-3-319-41920-6\\_5](https://doi.org/10.1007/978-3-319-41920-6_5)
- [14]. Allison, L., & Dix, T. I. (1986). A bit-string longest-common-subsequence algorithm. *Information Processing Letters*, 23(5), 305–310. [https://doi.org/10.1016/0020-0190\(86\)90091-8](https://doi.org/10.1016/0020-0190(86)90091-8)
- [15]. Mullin, R. (1985). Time warps, string edits, and macromolecules: The theory and practice of sequence comparison. edited by D. Sankoff and J. B. Kruskal. Addison-Wesley Publishing Company, Inc., Advanced Book Program, Reading, Mass., Don Mills, Ontario, 1983. 300 pp. U. S. \$31.95. ISBN 0-201-07809-0. *Canadian Journal of Statistics*, 13(2), 167–168. <https://doi.org/10.2307/3314879>