A Textual Description Based Approach to Process Matching

Maria Rana, Khurram Shahzad, Rao Nawab, Henrik Leopold, Umair Babar

A Textual Description based Approach to Process Matching

Maria Rana, Khurram Shahzad, Rao Muhammad Adeel Nawab,
Henrik Leopold, Umair Babar

University of the Punjab, Pakistan
Vrije Universiteit Amsterdam, The Netherlands
COMSATS Institute of Information Technology, Pakistan
{maria.rana, khurram, umair.babar}@pucit.edu.pk, adeelnawab@ciitlahore.edu.pk,
h.leopold@vu.nl

Abstract. The increasing number of process models in an organization has led to the development of process model repositories, which allow to efficiently and effectively manage these large number of models. Searching process models is an inherent feature of such process repositories. However, the effectiveness of searching depends upon the accuracy of the underlying matching technique that is used to compute the degree of similarity between query-source process model pairs. Most of the existing matching techniques rely on the use of labels, structure or execution behavior of process models. The effectiveness of these techniques is, however, quiet low and far from being usable in practice. In this paper, we address this problem and propose the use of a combination of textual descriptions of process models and text matching techniques for process matching. The proposed approach is evaluated using the established metrics, precision, recall and F_1 score. The results show that the use of textual descriptions is slightly more effective than activity labels.

1 Introduction

Business process models (hereafter process models) are widely used to formally document the business operations of an enterprise. That is because process models are proven to be an effective means for visualizing and improving their complex operations [1]. Due to the increasing number of models, enterprises have to maintain process model repositories which may contain up to hundreds or thousands of process models [2, 3]. The effective use of these collections requires searching relevant source process models against a given query process model [4]. This makes searching an integral feature of process model repositories [5, 6, 7]. The effectiveness of searching depends upon the efficiency of the underlying matching techniques that determines the degree of similarity between a pair of process models [8]. Existing matching techniques [9, 10, 11, 12, 13] take into account the three established feature classes of process models: label features, structural features, and behavioral features. However, the effectiveness of these techniques, is not sufficient [9] and far from being usable in practice [13]. Therefore, several efforts are being made to develop new techniques or

to combine existing techniques for process matching. Another limitation is that most of the techniques require a process *model* as input, which limits the number of users who can search process models.

As a contribution towards addressing these problems, in this paper, we propose to exploit the presence of textual descriptions of process models in a process repository and the availability of established text matching techniques for process matching. Specifically, we investigate, whether the use of textual descriptions performs better than using label features of process models. The reason for the choice of label features over structural and behavioral features is rooted in the fact that label features serve as a primary source for generating textual descriptions, whereas the other two features mainly contribute to the flow of the text. We contend, once the superiority of the use of textual descriptions over label features is established, it can be used in combination with structural and behavioral features for process matching.

In this paper, we first generate textual descriptions of 669 process models using a well-established textual description generation technique [18]. Second, we parse the same set of models to extract their activity labels. Subsequently, we apply four established text matching techniques, n-gram overlap [14], edit distance similarity [15], Longest Common Subsequence (LCS) [16], and Vector Space Model (VSM) [17] to evaluate the effectiveness of textual descriptions over activity labels.

The rest of the paper is organized as follows: Section 2 provides the background on process models and related work. Section 3 provides an overview of the proposed approach for process matching. Section 4 describes the corpus used for our experiments. Section 5 describes the experimental setup (similarity estimation models), and the analysis of the results. Finally, Section 6 concludes the paper.

2 Background

This section introduces the background to this work by first providing an example process model and its equivalent textual description. Then, we reflect on the related work with the help of the example process model.

2.1 Motivating Example

In order to illustrate the correspondences between a process model and its textual description, consider the example of a university's admission process shown in Figure 1. The example process model is depicted using the Business Process Modeling and Notation (BPMN) – the de facto standard process modeling language. The model contains one start event, seven activities, two XOR gateways, and one end event. The start event is represented by circle, activities are represented by rectangles with round edges, XOR gateways are represented by a diamond shape containing a cross, and the end event is represented by a solid circle.

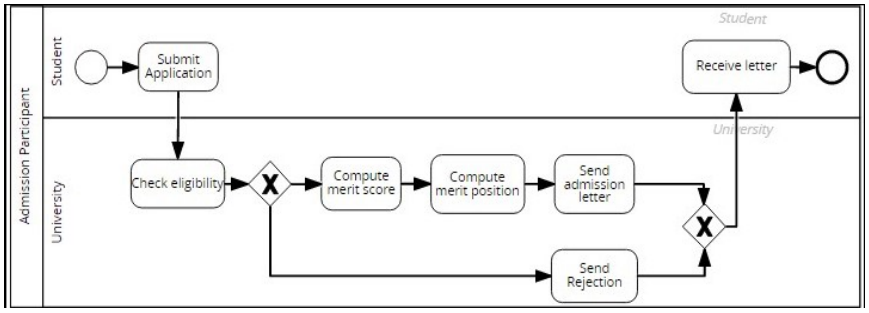

Fig. 1. University admission process model

The corresponding textual description of the example process model generated by using the Natural Language Generated System (NLGS) from [18] is shown in Figure 2. A careful look at the two specifications reveals the correspondences between the nodes (events, activities, and gateways) of the process model and the sentences of the textual description. For instance, both the model and the textual description specify that the process starts when a candidate submits an admission application. Also, it is clear from both specifications that after submitting the application, the eligibility of the candidate is checked.

```
The process begins when the student submits an application. Then,
the university checks the eligibility. Afterwards, one of the
following branches is executed:
        - The University computes the merit score. Subsequently,
       the University computes the merit position. Then, the
       University sends the admission letter.
       - The University sends the rejection.
Once one of the following branches was executed the student
receives the letter.
Afterwards, the process is finished.
```

Fig. 2. Textual description of the example process model

2.2 Related Work

In line with the three feature classes (label, structural, and behavioral) of process models, the related work to this research is classified into three main categories: label based approaches, structure based approaches, and behavior based approaches.

Label based approaches extract the activity labels of process models and apply matching approaches to evaluate the similarity between query-source process model pairs. The underlying techniques include the edit distance [15], the bag of words model [10] or contextual similarity [9]. Given two labels, approaches based on the edit distance compute the minimum number of atomic string operations (insertion, deletion, substituion of words) required to transform the sequence of query labels into the sequence of source labels and divides it by the maximum length of the two labels. Approaches based on the bag of words model divide labels into individual words and

compute the ratio of the the number of common words by the number of words in one or both labels [10]. In contrast to latter two techniques, context similarity takes into account the preceding and succeding label of activities to detect the equivalance of activites [9]. However, a key limitation is that these approaches consider two process models as similar by comparing the labels only. Thus, differences in the structure are not taken into account. For the process model from Figure 1 this means that any model with identicial activity labels is considered as similar, even if the gateways, actors, or the control flow between the activities are entirely different.

Structure based approaches generally disregard the labels of process models and rely on the topology of models to evaluate the similarity between query-source process model pairs. Among others, such approaches [9, 19] rely on the use of the graph-edit distance to compute similarity between models. Given two models, these approaches compute the number of graph edit operations (insertion, deletion or substituion of process elements) required to transform one model into another one. A typical limitation of these approaches is that they assume that semantically identical activites have identical or similar labels. [20] combines label matching and graph edit distance based approaches to compare the models. For the example model given in Figure 1, these approaches would focus structural aspects such as the decision after the second activity and disregard the specific meaning of the activities.

Behavior based approaches rely on the use of depdency graphs or causal footprints to evaluate the similary between query-source process model pairs. However, these approaches, such as [12], typically do not distinguish between certain connector types. For the example model from in Figure 1, such approaches may determine a query-source process model pair as equavalent even if contains OR gateways instead of XOR gateways.

The most relevant work to this paper is a recent contribution from [29], which promotes the use of textual descriptions on top of the process model. Matching the example model from Figure 1 with a query model requires the consideration of a document for checking the eligibility of student (as additional textual description for the activity *check eligibility*) and the document that explains the process of computing the merit score or the merit position (as additional textual descriptions for the activities *compute merit score* and *compute merit position*). In contrast to that approach, the approach we propose in this paper relies on the use of textual descriptions as an alternative to combining process models with additional textual descriptions of its activities.

3 The Proposed Approach

A brief overview of our proposed approach to process matching is presented in Figure 3. From the figure it can be seen that our process matching approach consists of a collection of source process models and their corresponding textual descriptions. While keeping textual descriptions alongside process models increases the comprehension of business processes among users, we propose to use the textual description for process matching.

Our approach relies on the use of an automatic approach to generate textual descriptions of a process model if needed. As far as we are aware, the Natural Language Generation System[1] (NLGS) is the only available tool that can automatically generate textual descriptions of a process model. It uses a well-established technique[2] that takes a process model in the JSON format as input and generates its textual description. For that reason, this system is used in our approach to automatically generate textual descriptions of all process models in the repository.

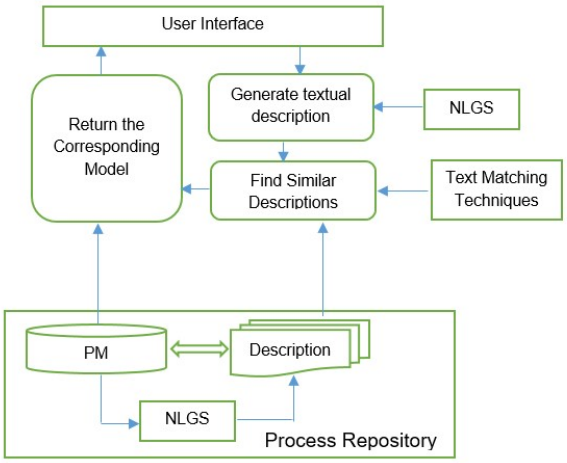

Fig. 3. Overview of the Proposed Approach

The input to our proposed approach is a query process model or its textual description. The task is thus to identify all the process models in the repository which are similar/relevant to the query in two/three major steps: i) generate textual description, ii) find similar process descriptions, and iii) identify the corresponding process models.

In the first step, if the input is a query process model, a textual description of the query process model is generated using the NLGS. Then, the (generated) textual description of the query process is compared to the textual descriptions of all the source process models in the repository. A ranked list of source process models is subsequently generated based on their similarity scores. In the third step, the top K source process models in the ranked list are marked as potential relevant process models against the query and returned to the user.

3.1 The Baseline Approach

As a baseline for comparison, we use the *label-based N-gram overlap* approach. That is because, among the three features classes (label, structural, and behavioral), label

[1] Available for download at http://www.henrikleopold.com/downloads/

[2] Runner-up McKinsey BT Award 2013, and winner TARGION Award 2014

features serves as a primary source for generating textual descriptions, whereas, the other two features mainly guide the structure and the flow of the textual description. Another reason is that the label contain all the important keywords of the process model, which makes a label based approach a logical baseline. Using this approach, the degree of similarity between the query-source process model pair is computed by counting the number of common words (extracted from the activity labels of the process models) between the query-source process models pair. Subsequently, it is divided by the length of one or both textual descriptions to get a normalized score between 0 and 1. The similarity score of 0 indicates that the query and source process models are entirely different and the similarity score of 1 indicates that they are exactly the same.

3.2 The Similarity Estimation Models

The following paragraphs give a brief overview of the estimation models used in this paper.

3.2.1 N-gram Overlap

The similarity between a query-source textual description pair is computed using a simple and well-known similarity estimation model, the n-gram overlap [14]. Note that we propose to use textual descriptions of process models instead of collections of labels that are used in the baseline approach. Using the similarity estimation model, both the query and the source textual descriptions are divided into chunks of length n (or sets of n-grams with length n). The degree of similarity between the query-source textual description pair is calculated by taking the intersection of the sets of n-grams of the query and the source textual descriptions and dividing it by the length of one or both textual descriptions to get a normalized score between 0 and 1. This similarity estimation model has been used in plagiarism detection [21], duplicate/near-duplicate document detection [22], and measuring text reuse in journalism [23]. For this paper, the similarity between the query-source textual description pair is computed using the overlap similarity coefficient. If S(Q, n) and S(S, n) represent the sets of unique n-grams of length n in a query textual description Q and a source textual description S respectively, then the similarity between them using the overlap similarity co-efficient is calculated using the following equation.

$$\mathrm{S}_{\mathrm{overlap}}(Q,S) = \frac{|S(Q,n) \cap S(S,n)|}{\min(|S(Q,n)|,|S(S,n)|)}$$

The range of the similarity score is between 0 to 1, where 1 means the two textual descriptions are exactly same and 0 means they don't have any common n-gram. In this paper, we have computed the similarity between the query-source textual description pairs for n = 1, i.e. unigrams. Before computing the similarity, all punctuation marks and stop words were removed and remaining words were stemmed using Porter's Stemmer.

3.2.2 Edit Distance Similarity

Edit distance is a distance-based model [15]. Using this model, the query-source textual description pair is first represented as a sequence of words or characters. Then, the number of atomic string edit operations (insert, delete, and substitute) required to transform the query textual description into the source textual description are counted. Subsequently, the edit distance is the minimum number of operations needed to transform the query textual description into the source textual description. For instance, if A = "abcd" and B = "abcdef", then the number of operations required to convert A to B is 2 (i.e. 2 insertions + 0 deletions + 0 substitutions). Similarly, the number of operations required to convert B to A is also 2 (i.e. 0 insertions + 2 deletions + 0 substitutions). The minimum number of operations is also 2. Thereafter, the similarity score between the query textual description (Q) and the source textual description (S) is computed using the following Equation.

$$E_s(Q,S) = 1 - \left\{ \frac{ed(Q,S)}{max(|Q|,|S|)} \right\}$$

where ed(Q, S) is the edit distance between query-source textual description pair.

3.2.3 Longest Common Subsequence Approach

The Longest Common Subsequence (LCS) [16] is another similarity estimation model used to compute the similarity between query-source textual description pairs. Using this similarity estimation model, the query-source textual description pair to be compared is represented as a sequence of characters or words. The number of edit operations (deletions and insertions) used to transform the query textual description into the source textual description are thereafter counted to compute the similarity between the textual descriptions. For instance, if A = "abcdef" and B = "abgdef", then abdef is the LCS between A and B.

In this paper, we used LCS to compute a normalized similarity score (called LCSnorm) between the query-source textual descriptions by dividing the length of LCS by the length of the shorter textual description. Since the LCS similarity estimation model is order-preserving, the alterations in the text caused by different edit operations (word substitutions, word re-ordering etc.) are reflected by the length of LCSnorm.

$$LCS_{norm} = \frac{|LCS|}{\min(|Q|,|S|)}$$

where |Q| and |S| are the lengths of the query and the source textual description respectively.

3.2.4 Vector Space Model

The VSM [17] is another similarity estimation model used to compute similarity between query-source textual description pairs. It computes the degree of similarity between manual-automatic description pairs by first representing the texts in a high dimensional vector space. The number of dimensions in the vector space is equal to

the number of unique words (or vocabulary) in the document collection. Then, the degree of similarity between a manual textual description (q) and a system textual description (d) is computed using the cosine similarity measure (see the Equation below).

$$sim(q,d) = \frac{\hat{q}.\hat{d}}{|\hat{q}| \times |\hat{d}|}$$

$$sim(q,d) = \frac{\sum_{i=1}^{n} q_i \times d_i}{\sqrt{\sum_{i=1}^{n}(q_i)^2 \times \sum_{i=1}^{n}(d_i)^2}}$$

4 The Corpus

This section provides details about the process model collection, query models, and the human annotations used in the experiment.

4.1 Source Process Models

We generated a collection of 669 BPMN process models and compared it to the widely used SAP Reference Model consisting of 604 process models. The goal was to illustrate the superiority of our developed collection with respect to the diversity in label and structure-related features [27]. For generating the collection, we employed a systematic protocol in order to handcraft the necessary diversity that we deem necessary for a benchmark collection. According to the protocol, at first 150 process models of different sizes, diameters, densities, network connectivity, sequentiality, separability and token split etc., were collected. For the interested reader we kindly refer to [24] for more details about the metrics. To generate diverse label and structural features we reproduced three other variants of these 150 process models, formally called, Near Copy (NC), Light Revision (LR), and Heavy Revision (HR). The NC variant is generated by 'slightly' changing the formulation of each label of a model in such a way that the semantic meanings of the labels are not changed. For instance, a possible NC of the label 'customer inquiry processing' could be 'client inquiry processing'. The LR variant is generated by 'substantially' changing the formulation of each label in such a way that the meanings of the labels are not changed. A possible LR of 'prepare replacement order' could be 'fulfill alteration request'. The HR variant is generated by making two types of changes to process models: a) changing the formulation of each label without changing the semantic meaning of the labels, and b) changing the structure (control flow) between activities of a process model in such a way that the semantics of the control flow remains intact.

In order to reduce the human bias, a team of three researchers was formed. To develop a common understanding of the variants, five example process models and their three variants were given to the researchers along with ample time to comprehend these models. This was followed by a discussion and an informal question answering session. The session was led by a three member advisory board

with expertise in business process modeling, natural language processing, and corpus generation. Subsequently, the 150 process models were divided into two subsets, 1-75 and 76-150, and each participant was asked to perform two revisions on a subset i.e. one researcher was asked to generate the NC and the HR variant on the first and the second subset respectively. The second participant was asked to generate the LR and the HR variant on the first and the second subset respectively. Similarly, the third participant was asked to generate the NC and the LR on the first and the second subset, respectively.

The smallest model in the collection contains 11 activities and largest model contains 54 activities. In terms of structural features, the average size of our collection of 669 process models is 20.75 with a standard deviation of 7.09, a diameter of 16.78 with a standard deviation of 5.46, a sequentiality ratio of 0.41 with standard a deviation of 0.17, and an average degree of connectors of 2.94 with a standard deviation of 0.52. Another key feature is that the process models in our collection are free of structural errors. For instance, the connector mismatch in our collection is 0. This indicates that there are no process models in our collection with a split connector (AND/OR/XOR) without a corresponding join connector (as requested by prominent process modeling guidelines [24, 25]). It is to be noted that the generation of process models with diverse label features required the participants to perform 24,092 operations (insertion, deletion synonyms replacement, and reordering or words). Similarly, to generate diversity with respect to structural features, 1,764 operations (adding/removing activities, adding/removing/changing gateways, adding/removing/ renaming lanes etc.) were performed by the participants.

Mendling et al. [26] highlighted that model understanding strongly depends upon accurate interpretation of the labels. Their study presented four semantic challenges about labels, including, the use of ambiguous grammar, label terms, compound words, and vocabulary with possibly different semantics. We generated another 69 process models for our collection by explicitly inducing semantic challenges to labels. Note that at least 17 models were added for each of the four semantic challenges. Accordingly, the generated collection has 669 process models. In addition to the 669 models, we generated textual description for each process model. The size of the descriptions ranges from 48 words to 394 words with an average of 13.7 words per label.

4.2 Query Process Models

From the collection of 669 models we selected 56 process models as query models. These numbers should be seen in the context of existing studies, such as [9], which *randomly* selected 10 query models and 100 source models to evaluate the effectiveness of their proposed approaches. In contrast to that, the choice of 56 query models in our case is not arbitrary. We rather employed a systematic procedure to choose the necessary and sufficient set of query models. The necessary and sufficient set is required because the chosen set of queries will afterwards be used to manually determine the relevance of the query process model against the set of 669 source process models. In case, the query models include models that are irrelevant, it will unnecessarily increase the human effort for manually determining the relevance

between models. Similarly, if relevant models are not included, the approach is not sufficiently useful.

Our set of 56 query models includes models with diverse structural and label features. For choosing the necessary and sufficient set of query models with respect to the *structural* features, we first computed the values of 15 widely used structural metrics (M) of 150 original process models. The structural metrics include size, diameter, density, coefficient of connectivity, average degree of connectors, maximum degree of connectors, separability ratio, sequentiality ratio, and token split. Subsequently, the correlation was calculated between all possible combinations of these metrics, i.e. $|m_i|^2 \mid \forall\ m_i \in M$. The pair of structural features (m_1-m_2) with a correlation value of 0.95 indicates that if we choose a process model with a higher score of the structural feature m_1, it is likely that the process model with a higher value m_2 is also chosen and vice versa. This part of the procedure ensures the choice of a sufficient set of query models. For the necessary set of structural features we chose query models with minimum and maximum value of each structural metric $\{m_i \mid \forall\ corr(m_i,m_j) <= 0.95\ \&\ m_i, m_j \in M\}$. Accordingly, 14 query models were chosen from the collection of 150 original models (recall Section 4.1 the collection of 669 models contains 150 original models).

For choosing the necessary and sufficient set of query models with respect to the *label* features, 14 query models from each process model variant (NC and LR) were chosen by using the procedure described in the preceding paragraph. Note that the diversity in the label features comes from the fact that the near copy variant was generated by 'slightly' changing the formulation of each label of the model and the light revision variant was generated by 'substantially' changing the formulation of each label of the model. Thus, the choice of queries from each variant ensures sufficient diversity in query models with respect to the *label* features.

The chosen set of 42 query models (14 query models from each, original, NC and LR) were analyzed once again to identify the necessary set of query models. The analysis revealed that the identified set of query models includes variants of the same query model, i.e. if P1 query model is included for the reason that it has the maximum value of the structural metric m_1, P1NC (its near copy variant) was also included. This is unnecessary because the query model P1 will be matched with all source models, including P1NC, to challenge the ability of the text matching technique to detect the label variant of P1. Nonetheless, the inclusion of P1NC also does not have a different value of the structural metric m_1, i.e. the inclusion of P1NC as query model simply add another model with exactly the same structural feature. To ensure a sufficient set of query models, the duplicate models (P1NC in the example case) were replaced by another near copy variant process model with the next maximum value of m_1. This ensures that another near copy variant of process model with next maximum value of metric m_1 is also included. The process was repeated until a unique set of query models were identified.

These 42 query models do include the structural and label variants. However, they do not include the process models where labels as well as the structure was changed. To overcome this limitation, 14 query models from the heavy revision process models are also chosen by using the same procedure described earlier. Heavy revision variants are generated by making two types of changes to the process models: re-writing labels and changing the structure (control flow) between activities. Thus, the

inclusion of heavy revision variants will challenge the ability of the text matching techniques to detect the process models where label and structure was changed as well. Accordingly, 56 queries models were generated for the experiment.

4.3 Human Annotations

To evaluate the retrieval performance of our proposed automatic methods (see Section 3), we need manual annotations of relevant source process model(s) against each query process model. This would require a comparison of 37,464 query-source model pairs. Given that, declaring the two process models to be equivalent requires comparing all activities, the amount of human effort is even more substantial. This is also the reason why existing studies, such as [9], used a small sample of 10 query models and 100 source models. In contrast to that approach, we created a sharply defined relevance screening criteria. Subsequently, two researchers were asked to independently compare 56 query models with 150 original models only to significantly reduce the human effort. At first glance, one may question the manual benchmark because not all pairs were compared. However, this is far from being true, since the remaining 519 models in the collection are handcrafted variants of these models with the same meaning. Hence, such a comparison is not necessary. We subsequently calculated the inter-rater agreement using Kappa statistics [28], which was 0.906. The inter-rater agreement score is very good which demonstrates that the human judgement was consistent across the researchers, and that the relevance screening criteria was sharp enough to be used in practice.

5 Experimental Setup

The proposed approach presented in Section 3 is implemented as a Java Prototype. For the experiments the complete collection of 669 process models and the 56 query models serve as input to the prototype. For each query the prototype returned a text file which contains the names of the source models and their similarity score with the query model, i.e. each file contains 669 source models and their similarity scores in descending order. The top K process models were subsequently separated. Afterwards, we used the manual annotations for computing average precision, recall, and F measure across different values of K.

Note that the experiments were repeated by using the collection of labels of all elements of the process models and by applying the four similarity estimation models explained in Section 3.2. We call it label-based approach. Similarly, the experiments were repeated after preprocessing, i.e. before applying the similarity estimation models for computing similarity between the query-source pairs. Each textual description/labels was pre-processed by removing stop words and remaining words were stemmed using the Snowball stemmer.

5.1 Evaluation measures

The main goal of this experiment is to measure the effectiveness of the proposed approaches in retrieving relevant source process models (from the repository) against a given query process model. To evaluate the retrieval performance of our proposed approaches we have used the metrics precision, recall, and F_1. The reason for selecting these measures is that they are standard evaluation measures for evaluating the performance of information retrieval approaches. In this context, precision represents the percentage of source process models that are retrieved and are relevant. Its value varies from 0 to 1, where 0 means that all the process models retrieved by the matching technique are irrelevant and 1 means that all the process model(s) retrieved by the matching technique are relevant, i.e. no irrelevant process model is retrieved. The precision score is computed by using the following equation.

$$Precision = \frac{|retrieved \cap relevant|}{|retrieved|}$$

Recall represents the percentage of source process models that are relevant and retrieved. The value of recall also varies from 0 to 1, where 0 means that none of the source process models that are relevant to the query model are retrieved by the process matching technique and 1 means all the source process models that are relevant to the query model are retrieved by the process matching technique, i.e. no relevant process model is missed by the matching technique. The recall score is computed by using the following equation.

$$Recall - \frac{|retrieved \cap relevant|}{|relevant|}$$

There is a trade-off between precision and recall. To give equal weight to both, F_1 measure is used, which is the harmonic mean of precision and recall. Formally, the F_1 score is computed by using the following Equation.

$$F_1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

6 Results and Analysis

Table 6.1 shows macro-averaged precision, recall and F_1 scores for 56 queries for the four similarity estimation approaches n-gram overlap, edit distance, Longest Common Subsequence (LCS), and Vector Space Model (VSM). In the Table, the results are reported for the top 3, 6, 9, 12, 15 source process models returned by a process matching technique against a query process model. The reason for keeping the gap to three is to evaluate whether the three variants (NC, LR, HR) of the process models are matched or not. Note that we also evaluated the effect of stop word removal and stemming on our proposed process matching techniques. The best results were obtained by using stop word removal.

Overall it can be noted that our proposed textual description based approach outperforms the label based approach in all cases and for all values of top K process models. This gives a clear indication that, compared to the label based approach, the process models returned by the textual description based approach is more effective for process retrieval. This is likely to happen because textual descriptions contain additional information about the process models in comparison to the labels of activities and events. It, for instance, includes, the actors associated with each activity as well as the flow between activities.

			Top K Process Models				
			3	**6**	**9**	**12**	**15**
Unigram	Label based (baseline)	P	0.51	0.38	0.29	0.25	0.21
		R	0.27	0.37	0.42	0.47	0.49
		F_1	0.35	0.37	0.34	0.33	0.30
	Text based	P	0.66	0.48	0.36	0.30	0.26
		R	0.37	0.50	0.54	0.58	0.61
		F_1	0.47	0.49	0.43	0.39	0.36
Edit Distance	Label based	P	0.57	0.40	0.30	0.24	0.21
		R	0.30	0.39	0.42	0.47	0.50
		F_1	0.39	0.40	0.35	0.32	0.29
	Text based	P	0.71	0.46	0.35	0.28	0.24
		R	0.42	0.52	0.57	0.59	0.63
		F_1	0.53	0.49	0.43	0.38	0.35
LCS	Label based	P	0.52	0.38	0.31	0.26	0.23
		R	0.28	0.37	0.44	0.49	0.52
		F_1	0.37	0.38	0.36	0.34	0.32
	Text based	P	0.66	0.47	0.37	0.30	0.26
		R	0.38	0.49	0.55	0.60	0.63
		F_1	0.48	0.48	0.44	0.40	0.36
VSM	Label based	P	0.61	0.46	0.39	0.32	0.27
		R	0.33	0.46	0.57	0.61	0.64
		F_1	0.43	0.46	0.46	0.42	0.38
	Text based	P	0.73	0.53	0.42	0.35	0.30
		R	0.42	0.56	0.63	0.69	0.73
		F_1	0.53	0.54	0.50	0.46	0.42

Table 6.1: Results using baseline and proposed approaches

As expected, the precision score decreases as the value of K increases. These decreasing values indicate that the more models we consider, the more irrelevant models are returned by all techniques. On the other hand, as expected, the recall score increases as the value of K increases. These increasing trends indicate the strength of the proposed method in detecting relevant process models. However, overall the F_1 score decreases as the value of K increases. This indicates that as we increase the value of K, the decrease in precision drops more sharply than the increase in recall, i.e. the proportion of irrelevant models returned are more than the numbers of relevant models returned.

Overall, the highest precision (P = 0.73 for top 3 process models) recall (R = 0.73 for top 15 process models) and F_1 (F_1 = 0.54 for top 6 process models) are obtained using the VSM approach. These scores are significantly higher than the baseline approach (P = 0.21, R = 0.49 and F1 = 0.30). This also reflects that among all the proposed approaches, VSM is the most effective in retrieving relevant process models from the corpus used in this study.

7 Discussion and Conclusions

In this paper, we presented a novel process model matching approach that relies on the use of textual descriptions of processes. The approach exploits the fact that process model repositories often include textual descriptions of processes. For the evaluation we implemented the proposed approach in Java and used it for a set of experiments. The prototype takes textual description or a process model as input and generates its textual description using a NLGS. Subsequently, various similarity estimation models used in text matching are applied to compute the similarity between the query-source process descriptions. We evaluated the proposed approach in terms of precision, recall, and F_1 metrics. The results show that the use of textual descriptions for process matching is slightly more effective for retrieval than the collection of labels.

Note that we are aware that the textual descriptions generated by the NLGS may not perfectly match the textual descriptions produced by a human. However, we content that it represents a separate research problem to investigate the similarity and differences between human and system generated textual descriptions and their use. For instance, what would be the impact of using a text crafted by human for process matching? How to correct the auto-generated textual description? Is the auto-generated and human generated textual description equally useful for process comprehension? All these questions represent promising directions for future work. Also, the comparison of textual descriptions using structure and behavior based approaches needs to be investigated.

References

1. Dumas, M., La Rosa, M., Mendling, J., Reijers, H.A.: Fundamentals of Business Process Management, Springer (2013)
2. Aa, H.V.D., Leopold, H., Reijers, H.A.: Detecting Inconsistencies between Process Models and Textual Descriptions. In Proc. of 13th International Conference on Business Process Management (BPM'15), Springer LNCS, vol. 9253, pp. 90--105. Springer (2015)
3. Zha, H., Wang, J., Wen, L., Wang, C., Sun, J.: A Workflow net Similarity based on Transition Adjacency Relations. Computer in Industry, 61, 463--471(2010)
4. Kunze, M., Weidlich, M., Weske, M..: Querying Process Models by Behavior Inclusion. Software and System Modeling 4, 1105--1125 (2015)
5. La Rosa, M., Reijers, H.A., Aalst, W.M.P., Dijkman, R., Mendling, J., Dumas, M., Banuelos, L.: APROMORE: An Advanced Process Model Repository. Expert Systems and Applications, 38, 7029--7040 (2011)
6. Shahzad, K., Elias, M., Johannesson, P.: Requirements for a Business Process Model Repository: A Stakeholders' Perspective. In Proc. of the 13th International Conference of Business Information Systems, Abramowizc, W.,Tolksdorf, R.(eds.), LNBIP,vol.47, pp.158—170. Springer (2010)
7. Elias, M.: Design of Business Process Model Repositories Requirements, Semantic Annotation Model and Relationship Metamodel. PhD Thesis, Department of Computer and Systems Sciences, Stockholm University, Sweden, 2015.
8. Becker, M., Laue, R.: A Comparative Survey of Business Process Similarity Measures. Computer in Industry. 63, pp. 148 –167 (2012)

9. Dijkman, R., Dumas, M., Dongen, B., Kaarik, R., Mendling, J.: Similarity of Business Process Models: Metrics and Evaluation. Information Systems. 36, 498—516 (2011)
10. C. Klinkmuller, I. Weber, J. Mendling, H. Leopold, and A. Ludwig: Increasing Recall of Process Model Matching by Improved Activity Label Matching. In Proc. of 11th International Conference on Business Process Management, Zdravkovic, J., Kirikova, M., Johannesson,P. (eds.), LNCS, vol. 8094, pp. 211-- 218. Springer (2013)
11. Grigori, D., Corrales, J., Bouzeghoub, M., Gater, A.: Ranking BPEL Processes for Service Discovery. IEEE Trans. on Services Computing, vol. 3, pp. 178--192, (2010)
12. Weidlich, M., Mendling, J., Weske, M.: Efficient Consistency Measurement Based on Behavioral Profiles of Process Models. IEEE Trans. on Soft. Eng. 37, 410—429 (2011)
13. Cervantes, A.A.: Diagnosing Behavioral Differences between Business Process Models. PhD Thesis, University of Tartu (2016)
14. Cedeno, A., Rosso, P., Benedi, J.: Reducing the Plagiarism Detection Search Space on the Basis of the Kullback-Leibler Distance. In: Proc. of 10th International Conference on Computational Linguistics and Intelligent Text Processing(CICLing 2009), LNCS, vol. 5449, pp. 523-534.Springer (2009)
15. Sankoff, D., Kruskal, J.: Time Warps, String Edits and Macromolecules: The Theory and Practice of Sequence Comparison. Addison-Wesley Publications(1983)
16. Nawab, R.: Monolingual Paraphrased Text Reuse and Plagiarism Detection. PhD Thesis, University of Sheffield, UK (2012)
17. Salton, G., Wong, A.,Yang,C.: A Vector Space Model for Automatic Indexing. Communications of the ACM. 18, 613--620, (1975)
18. Leopold, H., Mendling, J., Polyvyanyy,a.: Supporting Process Model Validation through Natural Language Generation. In: IEEE Trans. on Soft. Eng. 40, 818-840, 2014.
19. Dijkman, R., Dumas, M., Bañuelos, L.: Graph Matching Algorithms for Business Process Model Similarity Search. In Proc. of 7th of International conference on Business Process Management (BPM'09), LNCS 5701, pp. 48--63. Springer, Heidelberg (2009)
20. La Rosa, M., Dumas, M., Uba, R., Dijkman, R.: Business Process Model Merging: An Approach to Business Process Consolidation. ACM Transaction on Software Engineering Methodology. 22, 1—46, (2013)
21. Lane, C., Lyon, C., Malcom, J.: Demonstration of the Ferret Plagiarism Detector. In Proc. of 2nd International Plagiarism Conference (2006), UK.
22. Shivakumar,N., Molina,H.: SCAM: A Copy Detection Mechanism for Digital Documents. In Proc. of 2nd Int. Conf. on the Theory and Practice of Digital Libraries, USA, (1995).
23. Clough, P., Gaizauskas, R., Piao, S., Wilks, Y.: Measuring Text Reuse. In Proc. of 40th Annual Meeting on Assoc. for Comp. Linguistics, pp. 152—159, Stroudsburg (2002)
24. Mendling J.: Metrics for process models: Empirical Foundations of Verification, Error Prediction, and Guidelines for Correctness. Springer (2008)
25. Mendling, J. Reijers, H.A., Aalst,W.M.P.: Seven Process Modeling Guidelines (7PMG). Information and Software Technology. 52, 127--136 (2010)
26. Mendling, J., Leopold, H., Pittke, F.: 25 Challenges of Semantic Process Modeling. International Journal of Info. Sys. and Soft. Eng. for Big Companies. 1, 78--94, (2015)
27. Shahzad. K., Shareef., K., Ali, R.F., Nawab, R.M.A., Abid, A.: Generating Process Model Collection with Diverse Label and Structural Features. In Proc. 6th International Conference on Innovative Computing Technology (IEEE-INTECH'16). (to appear)
28. Fink, A.: Conducting Research Literature Reviews: From the Internet to Paper (3rd edition). London, Sage (2010)
29. Leopold, H., van der Aa, H., Pittke, F., Raffel, M., Mendling, J., Reijers, H.A.: Integrating Textual and Model-based Process Descriptions for Comprehensive Process Search. In Proc. of the BPMDS'16 Working Conference, Ljubljana, Slovenia (2016) (to appear)