

Mining Business Policy Texts for Discovering Process Models: A Framework and Some Initial Results

Jiexun Li Harry J. Wang Zhu Zhang J. Leon Zhao
Drexel University University of Delaware University of Arizona University of Arizona
jiexun.li@ischool.drexel.edu hjwang@lerner.udel.edu zhuzhang@u.arizona.edu jlzhao@u.arizona.edu

Abstract

Many organizations use digitized policy manuals to help govern their business operations. However, many business processes are often not synchronized in a timely manner with the business policies because of the high costs of redesigning business process models in the face of frequent policy changes. As such, there is a great need for more efficient ways to maintaining the alignment between business policies and processes. In this paper, we propose a novel approach that will help process designers extract process models out of policy manuals using information retrieval techniques; we refer to our approach as Policy Based Process Mining (PBPM). PBPM can help organizations better understand their business processes by analyzing and visualizing their existing policies. In this paper, we present the PBPM framework and validate the first step of the framework by identifying process-related business policies automatically. Our initial experiments using bag-of-words and tree kernel techniques produced very promising results.

Keywords: Process Mining, Business Policy, Business Process Management, Text Mining

1. Introduction

Business policies enable the efficient management of an organization by defining the standard procedures and rules for its daily business operations, e.g. eleven organization-wide policies are identified in (Peltier 2004), such as Employee Standards of Conduct, Workplace Security, Information Security, Business Continuity Planning, to name a few. Among various types of business policies, many of them are used to define or constraint some aspects of the business processes, such as order fulfillment, product development, travel reimbursement, and cash handling, which we refer to as *process policies*. For instance, a travel reimbursement policy may define that *a reasonable exception request form (RERF) must be submitted if a travel reimbursement form (TRF) is submitted later than 60 days upon completing the travel*. This policy specifies the condition under which RERF submission task must be executed.

In order to describe a business process, a great amount of process policies are often needed to be established. For example, we studied a major US public university's business policy manual in our previous research (Wang et al. 2006), which has 19 sections with average 10 subsections in each section. In particular, the subsection on travel regulation by itself includes 14 topics with more than seven thousand words. Recently, in order to achieve various regulatory compliances such as Sarbanes-Oxley, organizations are investing a great amount to revamp their business policies. The process owners, executives, internal and external auditors often have to study business policy manuals that have hundreds or thousands of pages to understand the whole business processes, which is a time-consuming and overwhelming task. In addition, it is also a challenge for employees to find the information about processes to do their job correctly and efficiently due to the huge amount of policies. We refer to this problem as *Process Policy*

Information Overload. As such, there is a great demand for solutions that can assist process users and owners to analyze process policies in order to discover process information more efficiently and effectively.

In this paper, we propose a Policy Based Process Mining (PBPM) framework to enable automatic process model extraction from business policies using text mining and information extraction techniques. PBPM aims to reduce process policy information overload aforementioned by automatically parsing business policies to determine what policies are process-related and to identify process components such as tasks, data items, roles, and constraints, which can be used to construct a graphical process model. We argue the resulting process model visualizes the related narrative process policies, which can greatly reduce the cognitive load of process owners and users. The contribution of this paper is threefold. First, we propose a new way to discover process information based on business policies, named Policy Based Process Mining (PBPM). Second, we demonstrate the feasibility of PBPM by applying different text mining techniques and discussing the results. Third, PBPM provides insights for developing advanced business policy management and process mining tools. PBPM bridges a critical gap between commercial needs and existing process mining research and therefore is of interest for both industry practitioners and academic researchers.

The rest of this paper proceeds as follows. In the next section, we briefly review the relevant literature. Then, we present the foundation of Policy Based Process Mining (PBPM) extending a previous study on process mapping, which leads to the PBPM framework. In Section 4, we apply a word feature method and a tree kernel method on two policy corpora to identify process-related sentences. We present our preliminary results and discuss their implications. Finally, we summarize our contributions and discuss our future research.

2. Brief Literature Review

Understanding existing organizational processes, i.e., as-is process models, is the foundation for any further process changes and improvements (Kettinger et al. 1997). Process mapping is a set of methodologies and tools that help organizations identify, understand, and improve their as-is processes (Hunt 1996). Traditional process mapping approaches are participative, where interviews, meetings, and workshops are used as the major instruments to collect process information (Cobb 2004; Madison 2005; Scheer 2000a). Different from a participative approach, an analytical process mapping approach aims to derive process model by using formal theory and techniques. Various formal methods such as linear programming (Aldowaisan and Gaafar 1999), process cost optimization (van der Aalst 2000), computational experiments (Hofacker and Vetschera 2001), and probability theory (Datta 1998) have been applied to analytical process design and redesign. Workflow design based on data dependencies has also been proposed (Reijers et al. 2003).

Although business policies have been used in both traditional and analytical process mapping approaches aforementioned, no existing method has focused on deriving process models from business policies in a systematic manner. In our previous research, we proposed a policy-driven process mapping methodology, which leverages business policies and extract process models from policies by following a set of algorithms and rules (Wang et al. 2006). Although that approach proves the possibility and provides insights for process model extraction based on business policies, the procedure of analyzing business policies for process information

is still manual, which does not provide direct solution for the process policy information overload problem discussed in the previous section. This paper is built on top of our previous work, which aims to automate the process model extraction from business policies using information retrieval and text mining techniques.

In recent years, there has been extensive research on process mining (PM) and many tools and techniques have been developed (van der Aalst and Weijters 2004). PM aims at the automatic discovering of process, control, data, organizational, and social constructs based on the event logs produced by contemporary information systems, such as ERP, CRM, and Workflow Management Systems (van der Aalst et al. 2007). The most well-known process mining algorithms are integrated in an open source toolset named ProM (www.processmining.org). ProM can import various types of event logs into a standard XML format and then construct process models from the logs, which are represented as Petri nets. Different from process mining, there also has been a increasing interest in monitoring business processes, which is often referred to as BAM (Business Activity Monitoring) and BPI (Business Process Intelligence) (Grigori et al. 2004). BAM and BPI applied classical data mining techniques to the event logs to discover knowledge on various performance indicators, such as flow time, resource utilization, and cost, which can be used to identify bottlenecks and modeling problems in the business processes. There have been many commercial BAM and BPI tools, such as Business Objects, Cognos BI, ARIS Process Performance Monitor, and Hyperion.

Our approach is related to both PM and BPI, because we leverage data mining techniques and aim to construct process models. However, unlike PM and BPI, our approach is unique by leveraging unstructured business policy documents rather than structured system event logs as inputs. In addition, the main goal of our approach is to reduce process policy information overload, which is not the purpose of PM and BPI. Next, we first show the foundation of policy based process mining (PBPM) using the results from our previous study, and then present PBPM framework in detail.

3. A Framework for Policy Based Process Mining (PBPM)

In our previous research (Wang et al. 2006), we conducted a case study of a business policy manual and found that rich process information can be found in those policies to construct graphical process models via a systematic procedure. That research serves as the foundation for our Policy Based Process Mining in this paper, which is discussed next.

3.1 Foundation for Policy Based Process Mining

Figure 1 is a screenshot of the business policy manual for a major US Midwest public university. This manual is published online and accessible to the public. In particular, we chose the section on travel approval and reimbursement for detailed analysis, which has been widely implemented in most organizations. The result of our study shows that business policies can be classified into a business policy taxonomy based on their relationship with processes and process components as shown in Figure 2. Business policies can be first divided into two categories, i.e. process policies and non-process policies, according to whether the policy is related to processes or not. For example, the policy statement “*The University's business policies are aligned with the Internal Revenue Service (IRS) rules and regulations*” is a non-process policy, because it does not contain information that related to any business process. In contrast, the policy statement “*All*

travel must be authorized and approved by the appropriate administrative officer within the unit” is a process policy, because it defines a task, i.e. travel authorization and approval, which is part of the travel approval business process. Given our goal is to construct process models, non-process related policies should be identified and excluded for further analysis.



Figure 1. Screenshot of an Online Business Policy Manual

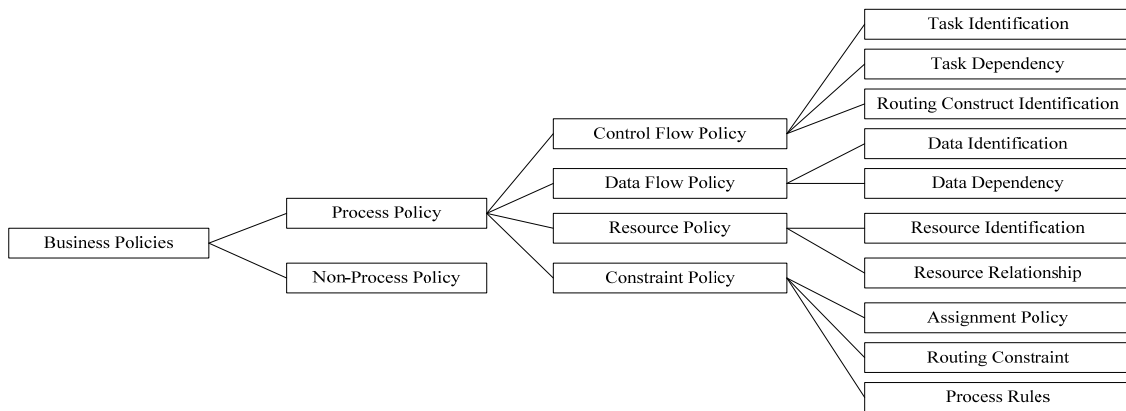


Figure 2. Business Policy Taxonomy

To better classify process policies, four types of process policies are further defined, which correspond to the four major components of a process model, namely, control flow, data flow, organizational model, and constraints (Basu and Kumar 2002; OMG 2005). For example, the second policy in the previous paragraph is a control flow policy, which is a task identification policy more specifically. As another example, the policy statement “Disbursement Services Center (DSC) must review the reimbursement form” is a process constraint policy, which assigns an organization resource, i.e. DSC, to execute the task “Review Reimbursement Form”. Note that a process policy could be of multiple types. For instance, the policy “After the Travel Reimbursement Form (TRF) is approved, a check will be issued” is a control flow policy and also a data flow policy, because it identifies two tasks “Submit TRF” and “Issue Check”, and two corresponding data items TRF and Check. This policy also defines the execution order of the two identified tasks, which can be used to construct the control flow model of the business processes. If process policies are complete, all process components and their relationships can be identified from policies to construct process models. Detailed discussion on different types of process policies and their relationships and how process model can be extracted based on those

information is not the focus of this paper and we refer the interested readers to (Wang and Zhao 2005; Wang et al. 2006) for more information. We summarize the key findings of the case study as follows, which are the foundations for automatic policy based process mining

- Process policies should be distilled from business policies to construct process models, whereas non-process related policies should be excluded;
- Process components including tasks, data items, resources, and constraints can be identified from business policies as the building blocks of process models;
- Relationships among process components, such as ordering relationship between tasks, can be identified from business policies to represent different aspects of process models, such as control flow aspect, data flow aspect, and organizational aspect.

Based on the findings above, we propose a policy based process mining framework as discussed next.

3.1 PBPM Framework

Figure 3 shows the framework for policy based process mining. The inputs of PBPM are the business policies in natural language and the outputs are the process models defined by the policies. PBPM consists of four major steps as discussed below:

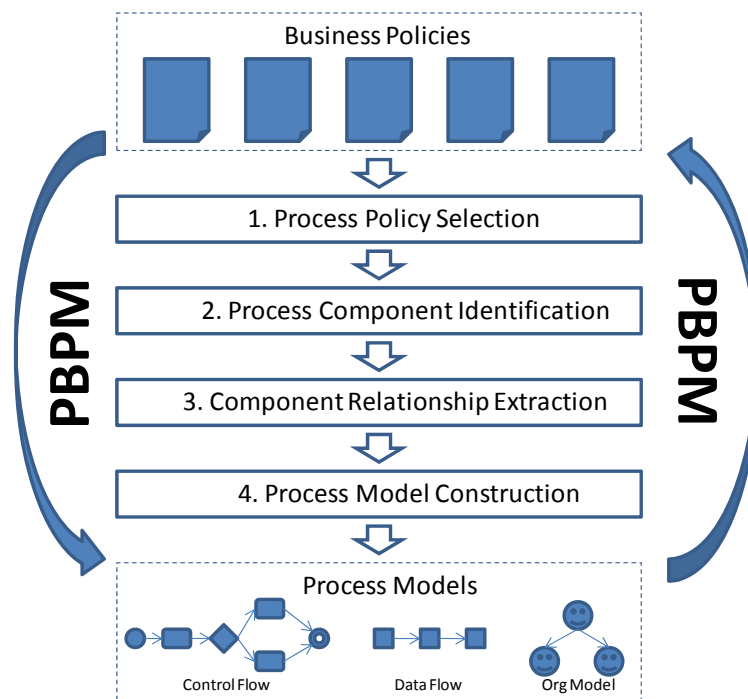


Figure 3. Policy Based Process Mining (PBPM) Framework

- Step 1: process policy selection. In this step, the process policies are separated from non-process policies to reduce the volume of policies needed for process model construction. We view this as a typical text classification problem. More specifically, domain experts are recruited to manually classify the policies as either process related or non-process related, which serve as the training data to the classification algorithms. Then, the trained classifier will be used to automatically identify process policies from business policies.

- Step 2: process component identification. Individual process components are identified from process policies selected by Step 1. In particular, we are interested in three process components: tasks, data items, and organizational resources. From a text mining point of view, we need to solve a named entity recognition (NER) problem, which is a sub-problem in information extraction. Typically, the NER problem is formulated as a sequence labeling problem, and one can approach it with Hidden Markov Models (HMM) or Conditional Random Fields (CRF). Existing general-purpose NER tools such as GATE (<http://gate.ac.uk/>) or LingPipe (<http://www.alias-i.com/lingpipe/>) are not directly applicable to our problem; therefore we need to retrain customized statistical models to extract entity types of our interest. Given the process components identified in Step 2, we can incorporate such information back to Step 1 so as to improve the process selection performance, e.g., the existence (or non-existence) of process components in a text chunk may increase (or decrease) the possibility of it being policy-related.
- Step 3: component relationship extraction. After the process components are identified, their relationships are extracted in this step. Specifically, we are interested in the following relationship: the ordering relationship between tasks, data dependency relationship, and relationship among organizational resources. Task execution order relationship can be directly used to construct control flow model. Data dependency can be used to build data flow model and in addition can be used to infer task sequences as discussed in our previous research (Wang et al. 2006). Resource relationship can be used to study the social network formed by different process participants and also can be used to specify the organization hierarchy. Technically, identifying relationships among entities is another sub-problem in information extraction. Rule-based methods and statistical learning methods are both potentially applicable to extracting intra-sentence relationships between components. Moreover, a more challenging task is to identify relationships among components across different sentences, which still has not been well studied in the information extraction field.
- Step 4: process model construction. Given the process components and their relationships, process analysts can construct the process models. Due to the fact that real business policies are often incomplete and contain ambiguities, more process data may need to be collected to completely specify the process models. In addition, some other issues such as eliminating process syntactical and semantic errors are also handled in this step as discussed in (Wang et al. 2006). Compared with the bulky policy documents, it is much easier and more efficient for the process analysts to understand the identified process components and relationships and use them to derive process models.

The main objective of PBPM is to automatically extract process models from narrative business policies to reduce the cognitive load of process owners and users. In addition, as shown in the Figure 3, PBPM also builds a bi-directional link between business policies and corresponding process models. This link can be used to check whether the business policies represent the actual processes in the field. Also, when policies and processes are changing as response to new regulations, customer demand, and market opportunities, this automatically constructed link can be used to identify the potential inconsistencies between them and therefore greatly facilitate the maintenance of their compatibility. In this section, we present the PBPM framework, which includes four steps. Next, we discuss two experiments we conducted to automatically classify process and non-process policies, i.e. PBPM Step 1, which demonstrates the feasibility of PBPM.

4. Experiments and Preliminary Results

In this preliminary study, we only focus on Step 1 of PBPM framework and formalize it as a sentence-level text classification problem. Each sentence s from a policy document is regarded as a data instance, and to be assigned a label l based on its relatedness to business processes. Specifically, label $\langle P \rangle$ stands for process and $\langle NP \rangle$ stands for non-process. Formally, the goal is to define or approximate a function

$$f : s \rightarrow l \quad l \in \{\langle P \rangle, \langle NP \rangle\}$$

The rationale of our approach is that sentences related to business process may exhibit certain lexical or syntactic patterns. We can use statistical learning techniques to extract these patterns by training a classification model on a corpus of labeled policy sentences. Thus, the trained model can be used to extract process from new policy documents.

4.1 Statistical Learning for PBPM

Feature Methods vs. Kernel Methods: Statistical learning can be categorized into feature methods and kernel methods. For feature methods, each data instance must be represented as a vector of n explicitly defined features to capture the data characteristics, $X = (x_1, x_2, \dots, x_n)$. Text mining tasks often use words or phrases as features, which can lead to high-dimensionality but sparse feature vectors. Furthermore, for sentences represented in complex structures such as a parse tree, features cannot be easily defined to capture the structural information. Kernel methods are an effective alternative to feature methods for machine learning (Cristianini and Shawe-Taylor 2000). They retain the original representation of objects and use the object only via computing a kernel function between a pair of objects. Formally, a kernel function is a mapping $K: X \times X \rightarrow [0, \infty)$ from input space X to a similarity score $K(x,y) = \phi(x) \cdot \phi(y) = \sum_i \phi_i(x)\phi_i(y)$, where $\phi(x)$ is a function that maps X to a higher dimensional space with no need to know its explicit representation. Such a kernel function makes it possible to compute the similarity between objects without enumerating all the features. Given a kernel matrix of pair-wise similarity values, a kernel machine, such as a support vector machine (SVM) (Cristianini and Shawe-Taylor 2000), can train a model for future prediction. Kernel-based methods have been frequently used in the machine learning areas, such as pattern recognition (Zhao et al. 2006), data mining (Zhou and Wang 2005), text mining (Sun et al. 2004), and Web mining (Yu et al. 2004).

Bag-of-words Feature Method: The “bag-of-words” method is a simple yet widely-used technique in text classification studies. It represents a chunk of text (in this case, a sentence) as a vector, in which each element indicates the occurrence of a specific word. Specifically, if a word w occurs in a sentence s , then in the feature vector representation of s , the feature corresponding to w takes a value “1”; if w does not occur in s , the feature takes a value of “0”. As a result, each data instance is represented as a vector of 1’s and 0’s, e.g., (1,0,0,1,0,...,1,0). For instance, given a sentence “Purchasing will take action by telephone and mail a confirming order to the vendor,” it can be represented as a vector in which features corresponding to the words within the sentence (e.g., “purchasing,” “will,” “take,” and so on) equal 1 while others equal 0. In this method, the dependencies among words are ignored, and the total number of features is determined by the total number of words in the text corpus.

Tree Kernel Method: In text mining applications, sentences are usually represented as syntactic parse trees to capture the syntactic structure. A parse tree is made up of nodes connected by branches. Each leaf node corresponds to a word. Each node in the tree is annotated

with a part-of-speech (POS) tag. Figure 4 shows the parse tree representation of the same exemplar sentence. A tree (or a subtree) T is represented as $\{p, [T.c]\}$, where p is the T 's root node with a set of attributes $V = \{v_1, v_2, \dots\}$ and $[T.c]$ denotes p 's children (nodes or subtrees). The node attributes often consist of word, POS, etc. Tree kernels can capture both the node attributes and the structural information of parse trees. They have been applied to information extraction (Culotta and Sorensen 2004; Zelenko et al. 2003). A tree kernel function computes the similarity between two parse trees by comparing their nodes and subtrees in a top-down fashion. A high similarity score of two parse trees indicates that the two sentences share common syntactic patterns and therefore tend to be in the same class. To define a tree kernel, we first need to define two functions over tree nodes: a matching function $m(p_i, p_j) \in \{0, 1\}$ and a similarity function $s(p_i, p_j) \in [0, \infty)$. The matching function determines whether two nodes are matchable (e.g., having the same POS tags) by comparing a subset of attributes, $V^m \subseteq V$:

$$m(p_i, p_j) = \begin{cases} 1, & \text{if } v_k^i = v_k^j, \forall v_k \in V^m \\ 0, & \text{otherwise} \end{cases}$$

where v_k^i and v_k^j is the value of attribute v_k of nodes p_i and p_j , respectively.

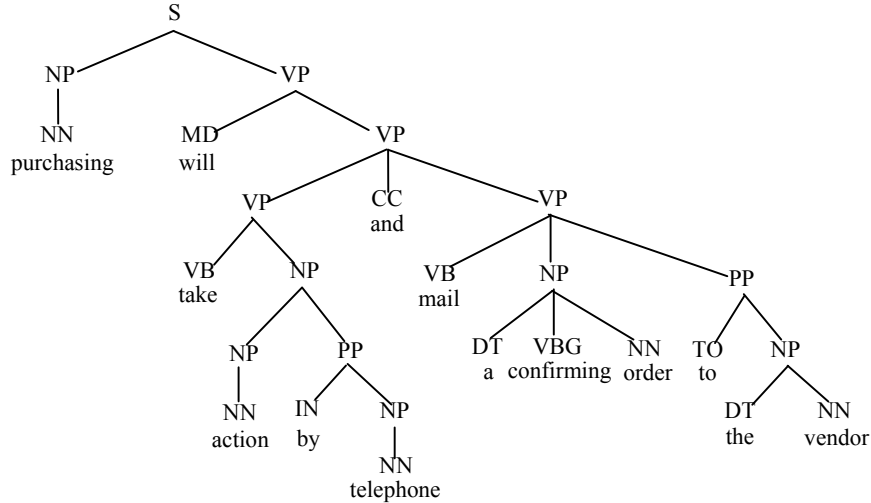


Figure 4. A Parse Tree Representation of a Sentence

If two nodes are matchable, then the similarity function is computed by comparing the other attributes of nodes, $V^s \subseteq V$:

$$s(p_i, p_j) = \sum_{v_k \in V^s} \omega_k C(v_k^i, v_k^j)$$

where $0 < \omega_k \leq 1$ is the weight of attribute k and $C(v_k^i, v_k^j)$ is a function that computes the compatibility between two attribute values (e.g., words):

$$C(v_k^i, v_k^j) = \begin{cases} 1, & \text{if } v_k^i = v_k^j \\ 0, & \text{otherwise} \end{cases}$$

Then $s(p_i, p_j)$ returns the weighted number of attributes values in common between p_i and p_j . For two relation instances T_1 and T_2 , we define the tree kernel $K_T(T_1, T_2)$ that includes the similarity of the parent nodes and the similarity of the children (i.e., nodes or sub-trees).

$$K_T(T_1, T_2) = \begin{cases} 0, & \text{if } m(T_1.p, T_2.p) = 0 \\ s(T_1.p, T_2.p) + K_c(T_1.c, T_2.c), & \text{otherwise} \end{cases}$$

where the similarity function K_c defined over children nodes $T.c$.

Let i be a sequence of indices such that $i_1 \leq i_2 \leq \dots \leq i_n$, and likewise for j . Let $d(i) = i_n - i_1 + 1$ and $l(i)$ be the length of i . For a relation instance T , let $T[i]$ denote a subsequence of children $T.c = \{T[i_1], \dots, T[i_n]\}$. Then we have

$$K_c(T_1.c, T_2.c) = \sum_{i, j, l(i)=l(j)} \lambda^{d(i)} \lambda^{d(j)} K(T_1[i], T_2[j])$$

where constant $0 < \lambda \leq 1$ is a decay factor that decreases the similarity between two sequences that are spread out within children sequences. For a pair of matching instances T_1 and T_2 such that $m(T_1.p, T_2.p) = 1$, the kernel function $K(T_1, T_2)$ needs to recursively compute the matching sequences of their children and accumulate the similarity scores.

4.2 Dataset Description and Evaluation Metrics

The first set of policies is from the purchasing policy manual of a major southwest public university, and the second set of policies is from the travel policy manual of a major midwest public university. Both sets of policies are publicly available html webpages, which are automatically downloaded, converted to text, and segmented into sentences. Some sentences including tables and long lists were regarded as noise and removed from our testbed. In total, the purchasing policy contains 336 sentences, among which 96 were tagged as ‘‘P’’ (i.e., process-related) and 240 as ‘‘NP’’ (i.e., not process-related). The travel policy contains 158 sentences, among which 31 were ‘‘P’’ and 127 as ‘‘NP.’’

We use standard machine learning evaluation metrics, accuracy, precision, recall, and F-measure, to evaluate the performances of process mining. These metrics have been widely used in text mining studies. In particular, accuracy measures the overall correctness. Precision, recall, and F-measure evaluate the correctness for each class. Specifically, precision indicates the correctness of identified relations and recall indicates the completeness of identified relations. F-measure is the harmonic mean of precision and recall.

$$\begin{aligned} \text{accuracy} &= \frac{\# \text{ of all correctly identified instances}}{\text{total \# of instances}} \\ \text{precision}(i) &= \frac{\# \text{ of correctly identified instances for class } i}{\text{total \# of instances identified as class } i} \\ \text{recall}(i) &= \frac{\# \text{ of correctly identified instances for class } i}{\text{total \# of instances in class } i} \\ \text{F-measure}(i) &= \frac{2 \times \text{precision}(i) \times \text{recall}(i)}{\text{precision}(i) + \text{recall}(i)} \end{aligned}$$

4.3 Experimental Results

We applied the bag-of-words and tree kernel methods on the two policy datasets individually. In the tree kernel computation, POS is used in the matching function, while word is used in the similarity function. The decay factors λ was set to 0.5. For both methods, we used a support vector machine (SVM) for learning the classification model due to its excellent performance reported in many applications. In particular, we chose an SVM package, LibSVM, for kernel learning (www.csie.ntu.edu.tw/~cjlin/libsvm), because (1) it has been frequently used

in previous studies, (2) it accepts customized kernels, and (3) it performs parameter selection for better performance.

For each dataset, we performed a 10-fold cross-validation to estimate the performances of process mining. Specifically, in every round, we used 9 folds as training data to learn a classification model and predicted the class label of sentences in the other testing fold. The final performance is averaged over all 10 rounds. Table 1 summarizes the performances of the process mining methods for these two datasets. We only report the precision, recall, and F-measure values for the positive class (i.e., process-related).

Table 1. Performances of PBPM on Two Sets of Business Policies

Policy	Methods	Accuracy	Precision	Recall	F-measure
Purchasing	Bag-of-words	77.33%	62.75%	40.00%	46.48%
	Tree kernel	78.71%	68.67%	36.25%	45.34%
Travel	Bag-of-words	87.69%	50.00%	45.00%	45.33%
	Tree kernel	89.23%	50.00%	30.00%	36.67%

In general, our preliminary results are encouraging but not very satisfactory. Compared with the naïve prediction as a baseline ($240/336 = 71.43\%$ for the purchasing policy and $127/158 = 80.38\%$ for the travel policy), both methods did achieve higher performance. However, there is still a lot of space for improvement of precision, recall, and F-measure. Specifically, on the purchasing policy corpus, the tree kernel method achieved higher overall accuracy (78.71%) and precision (68.67%) than the bag-of-words method, whereas bag-of-words had higher recall (40.00%) and F-measure (46.5%). The situation on the travel policy corpus is similar. Although both classifiers are conservative in assigning positive labels, the tree kernel tends to predict more sentences as negative (i.e., non-process). In distinguishing process-related sentences from others in policy documents, the sentence structural information captured by the tree kernel does not seem to be more powerful than the lexical patterns captured by bag of words.

5. Conclusion and Future Work

In this paper, we proposed a process mining framework we refer to as Policy Based Process Mining (PBPM), which can assist process designers in their efforts of extracting process models from narrative business policies. PBPM is aimed at reducing the cognitive load of process design by preprocessing large volumes of texts on business policies. In this paper, we presented our initial research effort that showed some promising results.

In particular, we applied text mining techniques to Step 1 of the PBPM framework, i.e., automatically separating process-related policies from non-process-related policies. We conducted experiments on two sets of business policies, whose results are encouraging and demonstrate the feasibility of PBPM. PBPM advocates a new way of discovering process knowledge based on business policies that widely exist in business organizations; our research initiates a new approach that can have impact on both academic research and industrial practice.

Since policy-based process mining is a brand new area, much research is needed. Specifically, we are planning to extend our work in the following directions:

- Collect more data to improve and evaluate the performance of the sentence classifiers. The two policy corpora used in current experiments have only about 500 sentences in total. Given

the complexity of our task, it is necessary to collect more data for model training and evaluation purposes.

- Conduct experiments on policies from different business domains. In order to validate the portability of our approach, we plan to collect data from additional business domains besides the higher education domain used in this paper and compare the experimental results derived from different business domains.
- Design experiments for the later steps in PBPM, i.e., extracting process components and inter-component relationships. Eventually, full process models should be built with the assistance of PBPM tools.

All in all, a framework like PBPM involves computational understanding of relatively deep semantics in human language, which is a very challenging task. Our research in text-based process discovery could spur new advances in related disciplines such as natural language processing and machine learning.

References

- Aldowaisan, T.A., and Gaafar, L.K. "Business process reengineering: an approach for process mapping," *Omega* (27:5) 1999, pp 515-524.
- Basu, A., and Kumar, A. "Research commentary: Workflow management issues in e-Business," *Information Systems Research* (13:1) 2002, pp 1-14.
- Cobb, C.G. *Enterprise Process Mapping: Integrating Systems for Compliance and Business Excellence* ASQ Quality Press, 2004, p. 128.
- Cristianini, N., and Shawe-Taylor, J. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*, Cambridge University Press, 2000.
- Culotta, A., and Sorensen, J. "Dependency tree kernels for relation extraction," 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04), Barcelona, Spain, 2004, pp. 423-429.
- Datta, A. "Automating the discovery of AS-IS business process models: Probabilistic and algorithmic approaches," *Information Systems Research* (9:3) 1998, pp 275-301.
- Grigori, D., Casati, F., Castellanos, M., Dayal, U., Sayal, M., and Shan, M.-C. "Business Process Intelligence," *Computers in Industry* (53) 2004, pp 321-343.
- Hofacker, I., and Vetschera, R. "Algorithmical approaches to business process design," *Computers & Operations Research* (28:13) 2001, pp 1253-1275.
- Hunt, V.D. *Process Mapping : How to Reengineer Your Business Processes*, Wiley, 1996, p. 288.
- Kettinger, W.J., Teng, J.T.C., and Guha, S. "Business process change: A study of methodologies, techniques, and tools," *MIS Quarterly* (21:1) 1997, pp 55-80.
- Madison, D. *Process Mapping, Process Improvement and Process Management*, Paton Press, 2005, p. 320.
- OMG "UML Superstructure Specification, v2.0," 2005.
- Peltier, T.R. *Information Security Policies and Procedures: A Practitioner's Reference*, (2 ed.), Auerbach Publication, 2004.
- Reijers, H.A., Limam, S., and van der Aalst, W.M.P. "Product-based workflow design," *Journal of Management Information Systems* (20:1), Sum 2003, pp 229-262.

- Scheer, A.-W. *ARIS - Business Process Modeling*, (Third ed.), Springer-Verlag, 2000a.
- Sun, A., Lim, E.-P., Ng, W.-K., and Srivastava, J. "Blocking reduction strategies in hierarchical text classification," *IEEE Transactions on Knowledge and Data Engineering* (16:10) 2004, pp 1305-1308.
- van der Aalst, W.M.P. "Reengineering knock-out processes," *Decision Support Systems* (30:4) 2000, pp 451-468.
- van der Aalst, W.M.P., Reijers, H.A., Weijters, A., van Dongen, B.F., de Medeiros, A.K.A., Song, M., and Verbeek, H.M.W. "Business Process Mining: An Industrial Application," *Information Systems* (32:1) 2007, pp 713-732.
- van der Aalst, W.M.P., and Weijters, A. "Process Mining," *Computers in Industry* (53:3) 2004.
- Wang, H.J., and Zhao, J.L. "Policy-driven Business Process Modeling in E-business," Fourth Workshop on E-business (WeB 2005), Las Vegas, Nevada, 2005.
- Wang, H.J., Zhao, J.L., and Zhang, L.-J. "Policy-Driven Process Mapping (PDPM): Towards Process Design Automation," The 2006 International Conference on Information Systems (ICIS 2006), Milwaukee, Wisconsin, 2006.
- Yu, H., Han, J., and Chang, K.C.-C. "PEBL: Web page classification without negative examples," *IEEE Transactions on Knowledge and Data Engineering* (16:1) 2004, pp 70-81.
- Zelenko, D., Aone, C., and Richardella, A. "Kernel methods for relation extraction," *Journal of Machine Learning Research* (3:6), Aug 15 2003, pp 1083-1106.
- Zhao, H.T., Yuen, P.C., and Kwok, J.T. "A novel incremental principal component analysis and its application for face recognition," *IEEE Transactions on Systems Man and Cybernetics Part B-Cybernetics* (36:4), AUG 2006, pp 873-886.
- Zhou, S., and Wang, K. "Localization site prediction for membrane proteins by integrating rule and SVM classification," *IEEE Transactions on Knowledge and Data Engineering* (17:12) 2005, pp 1694-1705.