Association for Information Systems
AIS Electronic Library (AISeL)

ICIS 2008 Proceedings

International Conference on Information Systems (ICIS)

2008

Relation-Centric Task Identification for Policy-Based Process Mining

Jiexun Li
Drexel University, jiexun.li@ischool.drexel.edu

Harry Jiannan Wang
University of Delaware, hjwang@lerner.udel.edu

Zhu Zhang
University of Arizona, zhuzhang@u.arizona.edu

J. Leon Zhao
University of Arizona, jlzhao@u.arizona.edu

RELATION-CENTRIC TASK IDENTIFICATION FOR POLICY-BASED PROCESS MINING

Identification des tâches centrées relation pour extraire les processus issus des politiques d'entreprise

Research-in-Progress

Jiexun Li
Drexel University
Philadelphia, PA 19104
jiexun.li@ischool.drexel.edu

Harry Jiannan Wang
University of Delaware
Newark, DE 19716
hjwang@lerner.udel.edu

Zhu Zhang
University of Arizona
Tucson, AZ 85719
zhuzhang@u.arizona.edu

J. Leon Zhao
University of Arizona
Tucson, AZ 85719
jlzhao@u.arizona.edu

Abstract

Many organizations use business policies to govern their business processes. For complex business processes, this results in huge amount of policy documents. Given the large volume of policies, manually analyzing policy documents to discover process information imposes excessive cognitive load. In order to provide a solution to this problem, we have proposed previously a novel approach named Policy-based Process Mining (PBPM) to automatically extracting process models from policy documents using information extraction techniques. In this paper, we report our recent findings in an important PBPM step called task identification. Our investigation indicates that task identification from policy documents is quite challenging because it is not a typical information extraction problem. The novelty of our approach is to formalize task identification as a problem of extracting relations among three process components, i.e., resource, action, and data while using sequence kernel techniques. Our initial experiment produced very promising results.

Keywords: Process mining, business policy, business process management, text mining, relation extraction

Résumé

L'extraction des processus à partir des politiques d'entreprise (PBPM) est une approche que nous proposons pour extraire automatiquement des modèles de processus à partir des documents de politique générale de l'entreprise. Cet article se concentre sur l'identification des tâches qui est une étape importante de l'extraction des processus et que nous présentons comme un problème d'extraction des relations entre les ressources, les actions et les données élémentaires. Nos expérimentations initiales fournissent des résultats très prometteurs.

Introduction

Business policies are developed to define the standard procedures and rules for daily organizational operations, such as Employee Standards of Conduct, Workplace Security, Information Security, Business Continuity Planning, etc. (Peltier 2004). Many business policies are used to define or constraint some aspects of the business processes, such as travel reimbursement, financial accounting, human resource management, and product development, which we refer to as *process policies*. For example, a travel reimbursement policy may define that "*If reimbursement amount exceeds the limit, a Travel Exception Form (TEF) must be filled.*" This policy specifies the condition under which TEF submission task must be executed. For complex business processes, huge amount of narrative policy documents need to be established. For example, we conducted a case study of 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. In another case study on the product development process of a medical equipment and reagent manufacturing company, the amount of process documents involved is even more tremendous (Karunakaran 2007). As required by regulations such as Sarbanes-Oxley, process policies must be frequently audited to ensure their correctness and consistency with respect to the actual business operations in the field. Therefore, the process owners, executives, internal and external auditors often have to study process policies that have hundreds of pages to understand the whole business processes, which is a time-consuming task and demands excessive cognitive load. We refer to this problem as *Process Policy Information Overload*, which calls for solutions that can help people discover process information from process policies more effectively.

Currently, business process models are built by means of the so-called participative approach, which requires extensive meetings with process participants including managers, engineers, and clerical workers. Our previous experiences indicated that many process participants often refer to business policies during the meetings in order to learn more about the business processes. Therefore, any automated approach to task identification should be complementary to the manual process design via the participative approach. In our previous research (Li et al. 2007), we proposed 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 by automatically parsing process policies to identify process components such as tasks, data items, and resources, and their relationships, which can be used to construct a graphical process model. The resulting process model visualizes the narrative process policies and can greatly reduce the cognitive load of process analysts. In this paper, we revise our previous PBPM framework to reflect our new research developments and findings as shown in Figure 1.

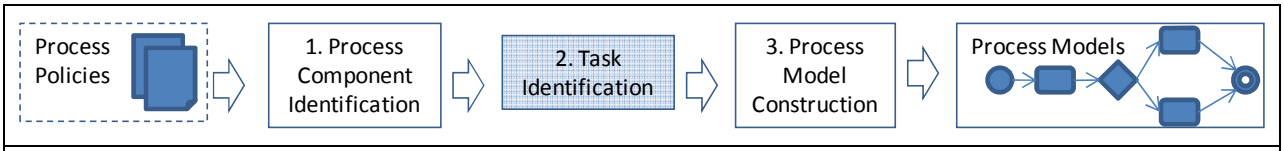


Figure 1. PBPM Framework

The inputs for PBPM are the process policies in natural language and the outputs are the graphical process models defined by the policies. PBPM consists of three major steps as briefly discussed below:

- Step 1: process component identification. Major perspectives for a process model include control flow, data flow, and its organizational model (Basu et al. 2000). Therefore, key process components are identified from process policies, including data, resources, and actions. For example, from the process policy statement "the traveler must submit a request for reimbursement to the department within 30 days upon completing the travel," we can identify a data item "request for reimbursement", two resources "traveler" and "department", and an action "submit". Data and resources can be directly used to construct data flow and organizational model, where tasks can be derived based on the relationships among resource, data, and action leading to the control flow model. We have conducted experiments for this step in our previous research and produced encouraging results.
- Step 2: task identification. Based on identified resources, data, and actions, tasks are defined as relations among those three components and are extracted in this step. For example, a task "traveler submits requests for exceptions" can be identified from the policy statement as a relation among the identified resource "traveler",

action "submit", and data item "request for reimbursement". This is the focus of this paper, which will be discussed in detail in the following sections.

- Step 3: process model construction. Given the identified tasks, process model can be constructed by analyzing the ordering sequence of those tasks. Visualization techniques are applied to help process analysts build process models. We have used Java Universal Network/Graph Framework (http://jung.sourceforge.net/) to visualize the relationships among resource, data, actions of identified tasks, which helps determine the task ordering sequence. Compared with unstructured process policies, the identified tasks can greatly reduce the cognitive load of process analysts and enable them to get a first-cut process model in a much more efficient manner.

In this paper, we extend our previous research by focusing on Step 2 of PBPM, i.e., task identification. In particular, we claim the following contributions: (1) identifying key challenges of task identification from unstructured policy documents, (2) formalizing task identification as a relation extraction problem and designing a procedure to address the challenges, (3) conducting experiments to validate the procedure. PBPM advocates a new way of discovering process models from unstructured documents. Our work in this paper further validates PBPM by providing a unique approach to extract task information, which to the best of our knowledge has not been reported in existing literature.

The rest of this paper proceeds as follows. In the next section, we briefly review the relevant literature. Then, we present the details on task identification. After that, we present the experiment design and results. Finally, we summarize our contributions and discuss our future research.

Brief Literature Review

Process mapping is a set of methodologies and tools that help organizations identify, understand, and improve their business processes (Hunt 1996). Participative process mapping approaches use interviews, meetings, and workshops 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, such as linear programming (Aldowaisan et al. 1999), process cost optimization (van der Aalst 2000), computational experiments (Hofacker et al. 2001), data dependencies analysis (Reijers et al. 2003), and probability theory (Datta 1998). 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 unstructured business policies using information retrieval and text mining techniques, which is the uniqueness of this paper.

Process mining (PM) has been extensively discussed in the literature and many tools and techniques have been developed (van der Aalst et al. 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). Research on Business Process Intelligence (BPI) has gained increasing attention, where classical data mining techniques are applied to the event logs to discover knowledge on various performance indicators, such as flow time, resource utilization, and cost (Grigori et al. 2004). Unlike PM and BPI, our policy-based process mining approach leverages 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.

Statistical learning, the main vehicle for finding patterns in data, 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, $X = (x_1, x_2, \ldots, x_n)$. 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 et al. 2000). They retain the original representation of objects and use the objects 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 $\phi_i(x)$ 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 et al. 2000), can train a model for future prediction.

In Natural Language Processing (NLP), various kernels have been applied to information extraction. For simple data representations (e.g., "bag-of-word") in which features can be easily extracted, some basic kernel functions such as linear kernel, polynomial kernel, and Gaussian kernel are often used. For data in structured representation,

convolution kernels are frequently used (Collins et al. 2002). Convolution kernels are a family of kernel functions, including string/sequence kernels (Lodhi et al. 2002), tree kernels (Zelenko et al. 2003), and so on. They define the similarity between objects as the convolution of "sub-kernels," i.e., the kernels for the decomposition of the objects. String kernels capture the sequence patterns in data instances. Lodhi et al. (2002) defined string kernels on letter or word sequence in sentences for text classification. Tree kernels capture the structure of syntactic parse tree and have been applied in relation extraction (Zelenko et al. 2003). Some recent studies have revised these tree kernels by incorporating richer semantic information (Bunescu et al. 2005; Culotta et al. 2004). Another advantage of kernel methods is that they transform different data representations into kernel matrices of the same format, which enables the integration of heterogeneous information (Cristianini et al. 2000).

Relation-centric Task Identification

In this section, we discuss the task identification step of policy-based process mining in detail. We first present some key challenges for this step, which make this research problem unique. Then, we explain how we address the challenges by formalizing task identification as a relation extraction problem and designing a procedure for this step.

Key Challenges of Task Identification

- *Challenge 1: Task definition from a text mining point of view.* A task or activity in a business process has a very simple definition in the business process management literature. Workflow Management Coalition defines an activity (task) as "a description of a piece of work that forms one logical step within a process" (WfMC 1999). However, in order to automatically identify task from policy documents, we need a new task definition from a text mining perspective. As we explain in the next section, based on the study of process component identification, i.e. step 1 of PBPM, we define tasks as relations between different process components to provide a new way of defining tasks in business processes for automatic task extraction.
- *Challenge 2: Formulating task identification as a relation extraction problem.* Relational data hidden in other types of text, such as news documents or biomedical literature, are often binary in nature, in the sense that they typically involve a pair of entities, e.g. a person and an organization, or a pair of proteins. For task identification in the context of process mining, the relations can be binary or ternary. For example, in the policy statement "the traveler must submit a request for reimbursement to the department within 30 days upon completing the travel", the task "traveler submits request for reimbursement" is a relation among three entities, i.e. resource (traveler), action (submit), and data (request for reimbursement).
- *Challenge 3: Developing appropriate relation extraction techniques.* Due to the existence of ternary relations in the task identification problem, existing relation extraction techniques may not be directly applicable. In particular, context information for each entity, i.e., process component, must been considered. For instance, in the previously mentioned policy example, there is another resource "department", but "department submits request for reimbursement" is apparently not a valid task for this statement. Furthermore, given that each sentence may contain multiple resources, data items, and actions forming exponential number of resource-action-data combinations as candidate tasks, how to identify valid and meaningful tasks imposes a great computational challenge.

Formalizing Task Identification as Relation Extraction

In step 1 of the PBPM framework, process components such as resources (*R*), actions (*A*), and data items (*D*) are identified from policy sentences (Li et al. 2007). Given Challenge 1, we studied five sets of tagged policies on different universities' travel regulation, aiming to discover linguistic patterns for tasks. We found that tasks are always associated with an action and some data items, such as "submits (action) form (data)" and "issue (action) check (data)". We also noticed that some tasks also explicitly contain resources, such as "department head (resource) approves travel", and "executive director (resource) reviews travel exceptions". Based on the case study of the policies, we define a task as a triple consisting of a resource, an action, and a data item, in the form of "*R*----*A*----*D*." A task must contain an action *A*. However, either *R* or *D* (not both) may not be explicitly expressed. Hence, both "*NULL*----*A*----*D*" and "*R*----*A*----*NULL*" are still valid tasks. Because a sentence may contain multiple *R*'s, *A*'s, or *D*'s, multiple potential task instances can be derived from a sentence but not all of them are meaningful. In order to discriminate meaningful (true) tasks from the others, we can formalize this as an intra-sentence relation extraction problem. Specifically, given a sentence S consisting of resources $\{R_1, \ldots, R_i\}$, actions $\{A_1, \ldots, A_j\}$, and data items

$\{D_1, \ldots, D_k\}$, we aim to derive all potential task instances T_{ijk} = <R_i----A_j----D_k> and identify all meaningful tasks among them using binary classification.

Although there are many different types of learning methods, in this study we choose kernel-based learning methods for task extraction for three reasons. First, unlike feature methods, kernel methods do not require explicit elaboration of all features for learning. Second, kernel methods can capture complex structural information in data instances (i.e., sentences in our case). Third, kernel methods also allow defining a composite kernel to combine different information in different formats. Among various popular convolution kernels used in NLP, we choose to use a sequence kernel in this study. The sequence kernel function is defined on an ordered list of tokens in a sentence. Such a sequence can be further decomposed into subsequences of *n*-grams. The string kernel proposed by Lodhi et al. (2002) was first used for text classification by analyzing the subsequences of letters or words. It has been extended by incorporating additional information (e.g., part-of-speech) of tokens and applied to relation extraction from text (Bunescu et al. 2006). Specifically, the simple "bag-of-words" kernel can be regarded as a special case of sequence kernel when only unigram words are considered. We did not use other more complex kernels such as tree kernels (Zelenko et al. 2003), because policy documents often contain sentences of unusual structures (e.g., long list of terms) which often cause errors for many syntactic tree parsers.

Task Identification Procedure

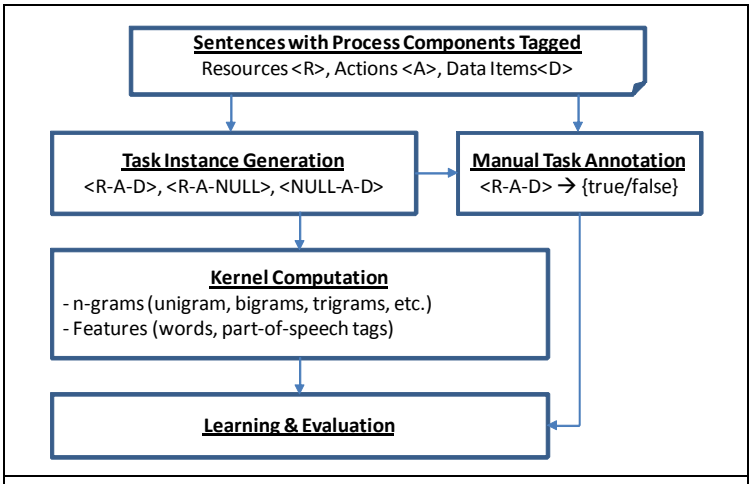


Figure 2. Task Identification Procedure in PBPM

In order to address Challenge 3, we designed a procedure for task identification in policy based process mining as shown in Figure 2. This procedure shows the detailed steps for task identification and the algorithms. We use the following sentence as an example to illustrate the detailed steps of the task identification procedure: "*The traveler must submit a request for reimbursement to the department within 30 days upon completing the travel*."

Task Instance Generation: given the policies tagged with resources <R>, actions <A>, and data <D>, we need to first generate all possible <R>-<A>-<D>, <R>-<A>, and <A>-<D> tuples. The words corresponding to "actions" are also stemmed to facilitate annotation. We developed a program to automatically conduct this task. Given two Rs ("traveler" and "department"), one A ("submit"), and one D ("request for reimbursement") in the sentence, five candidate task instances can be derived as follows:

<R | traveler>-<A | submit>-<D | request for reimbursement>
<R | department>-<A | submit>-<D | request for reimbursement>
<R | traveler>-<A | submit>
<R | department>-<A | submit>.
<A | submit>-<D | request for reimbursement>

Manual Task Annotation: although statistical learning does not require manual encoding of rules, we need a corpus with all valid tasks annotated by domain experts to train a statistical model. Specifically, the domain expert will go over all possible <R>-<A>-<D>, <R>-<A>, and <A>-<D> tuples and label each tuple as either 1 (valid task) or 0 (invalid task). The five task instances are annotated as follows:

<R | traveler>-<A | submit>-<D | request for reimbursement>	1
<R | department>-<A | submit>-<D | request for reimbursement>	0
<R | traveler>-<A | submit>	1
<R | department>-<A | submit>	0
<A | submit>-<D | request for reimbursement>	1

Kernel Computation: for each potential task instance T_{ijk}, we need to determine the sequence(s) to be considered as its context in the sentence. As we discussed in Challenge 2, the definition of a sequence for a task instance is not trivial. Obviously, it is not a good idea to use the whole sentence as the sequence for a task instance. Otherwise, task instances derived from one sentence will be represented in the same sequence and hence the learning method could not learn the discriminant patterns among them.

Based on a basic principle that sequences to represent different task instances from the same sentence should be different, we define two action-centered sequences ($S[R…A]$ and $S[A…D]$) for each task instance. Specifically, $S[R…A]$ represents the word sequence between (and including) R and A, and $S[A…D]$) represents the word sequence between (and including) A and D. It is worth noting that in a sentence R, A, and D, if any, may be in any order, but $S[R…A]$ and $S[A…D]$ should always follow the linear word sequence in the sentence. In this study, we do not consider the before- or after-component-pair sequences so as to simplify the computation and to reduce the similarity between task instances from the same sentences. For example, task instance T_1, <*R* | traveler>-<*A* | submit>-<*D* | request for reimbursement>, is represented by the following two sequences:

$S_1[R…A]$: *traveler* must *submit*
$S_1[A…D]$: *submit* a *request for reimbursement*

Similarly, <*R* | department>-<*A* | submit>-<*D* | request for reimbursement> is represented by:

$S_2[R…A]$: *submit* a request for reimbursement to the *department*
$S_2[A…D]$: *submit* a *request for reimbursement*

The mining algorithm follows the kernel-based statistical learning framework reviewed before. Given two task instances T_1 and T_2, we calculate the similarity between $S_1[R…A]$ and $S_2[R…A]$ and the similarity between $S_1[A…D]$ and $S_2[A…D]$ using a sequence kernel function K_S, respectively. We implement the sequence kernel function by following the string kernel proposed in (Lodhi et al., 2002).

A subsequence is a finite sequence of tokens. For sequence s, t, we denote by $|s|$ the length of the sequence. $s = s_1…s_{|s|}$. The sequence $s[i:j]$ is the subsequence $s_i…s_j$ of s. We say that u is a subsequence of s, if there exist indices $\mathbf{i} = (i_1,…,i_{|u|})$, with $1 \le i_1 < … < i_{|u|} \le |s|$, such that $u_j = s_{i_j}$ (the i_jth token in s), for $j = 1, …, |u|$, or $u = s[\mathbf{i}]$ for short. The length $l(\mathbf{i})$ of the subsequence in s is $i_{|u|}-i_1+1$. Thus, a sequence s can be mapped into a high-dimensional feature space, in which each dimension $\phi_u(s)$ corresponds to a subsequence u. We define

$$\phi_u(s) = \sum_{\mathbf{i}:u=s[\mathbf{i}]} \lambda^{l(\mathbf{i})}$$

where $0 < \lambda \le 1$. These features measure the number of occurrences of subsequences in the s weighting them according to their lengths. λ is the decay factor to penalize subsequences with more interior gaps and therefore longer length. Hence, the inner product of the feature vectors for two sequences s and t gives a sum over all common subsequences weighted according to their frequency of occurrence and length.

$$K_S(s,t) = \sum_u \phi_u(s) \cdot \phi_u(t) = \sum_u \sum_{\mathbf{i}:u=s[\mathbf{i}]} \sum_{\mathbf{j}:u=t[\mathbf{j}]} \lambda^{l(\mathbf{i})+l(\mathbf{j})}$$

By introducing some additional functions, we can compute the kernel function in a recursive and efficient manner (Lodhi et al., 2002). Besides, we may also consider tokens' different attributes such as word and part-of-speech (POS) tag when calculating the sequence similarity. Last, we can combine the two sequence kernels into a composite kernel for two task instances:

$$K(T_1, T_2) = K_s(S_1[R…A], S_2[R…A]) + K_s(S_1[A…D], S_2[A…D]).$$

Learning and Evaluation: after the model is trained using the annotated corpus, we need to evaluate its performance against a testing set. A standard cross-validation method can be used for evaluation. Some common evaluation metrics in information extraction are used including accuracy, precision, recall, and F-measure. The details of this step are further discussed in the next section.

Experiments and Results

Test-bed

We used a set of travel policy documents from a large public university in the US. These policy documents are publicly available as HTML webpages. We downloaded the documents and segmented them into individual sentences. Some sentences including tables and long lists of objects were regarded as noise and thus removed. In total, our test-bed contains 202 sentences from the travel policy. Our proposed approach requires training a model by learning from an annotated dataset. We had an expert in business process modeling manually annotate the key process components in the 202 sentences. In total, 530 process components (i.e., 180 resources <R>, 75 actions <A>, 275 data items <D>) appear in these sentences. There are 62 sentences containing at least one A and at least one R or one D, resulting in 378 candidate task instances. Among them, 193 instances (51.06%) are meaningful tasks and are labeled as 1, whereas the other 185 (48.94%) are labeled as 0.

Experimental Design

In our experiments, we compared sequence kernels under four different settings. The four kernels vary in two aspects: the sequence length (n-grams) and the attributes considered for each token, as summarized in Table 1. In particular, the first kernel K_I is simply a "bag-of-words" kernel in which only unigram words are used in kernel computation. The fourth kernel K_{IV} considers both unigrams and bigrams of word and POS features in the sequence kernel.

Table 1. Four sequence kernels for comparison

Kernels	n-grams	Attributes
K_I	unigrams	Word
K_{II}	unigrams	Word + POS
K_{III}	unigrams + bigrams	Word
K_{IV}	unigrams + bigrams	Word + POS

A popular POS tagging tool, StanfordPOSTagger (http://nlp.stanford.edu/software/tagger.shtml), was used to automatically tag sentences in the test-bed. We chose a widely used SVM package, LibSVM, for kernel learning (www.csie.ntu.edu.tw/~cjlin/libsvm). In our experiments, we conducted a cross-validation to estimate the performances of task identification. Cross-validation is a standard evaluation methodology for classification in machine learning research (Kohavi 1995). Specifically, we performed a leave-one-out cross-validation (LOOCV) at the sentence level. Each fold contains task instances derived from one single sentence. For each round, one fold was left out as the testing set to validate the model trained on the instances from the remaining folds. The testing results for all folds were then averaged to get the overall estimate of the classification performance. Standard evaluation metrics for information extraction, i.e., accuracy, precision, recall, and F-measure, were used in our evaluation. Specifically, accuracy evaluates the overall correctness. Precision, recall, and F-measure evaluate the correctness for each class. Precision indicates the correctness of identified tasks and recall indicates the completeness of identified tasks. F-measure is the harmonic mean of precision and recall. Accuracy, precision, recall and F-measure are formally defined as follows:

$$\text{Accuracy} = (\#\text{ of all correctly identified instances}) / (\text{total }\#\text{ of instances})$$

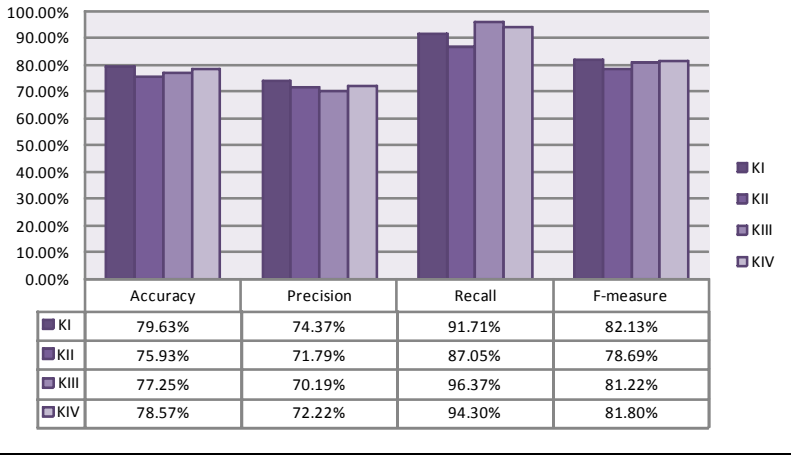
$$\text{Precision}(i) = (\#\text{ of correctly identified instances for class } i) / (\text{total }\#\text{ of instances identified as class } i)$$

$$\text{Recall}(i) = (\#\text{ of correctly identified instances for class } i) / (\text{total }\#\text{ of instances for class } i)$$

$$\text{F-measure}(i) = (2 \times \text{Precision}(i) \times \text{Recall}(i)) / (\text{Precision}(i) + \text{Recall}(i))$$

Experimental Results and Discussion

Figure 3 shows the performances of the four kernels for task extraction. Even with such a small corpus, these four kernel methods demonstrated satisfactory classification performances. As compared to random classification by prior probability (51.06%), all our methods achieved more than 75% classification accuracy. In particular, our approach missed very few true tasks (~95% recall), and users only need to review the identified tasks and screen out a few (less than 30%) false positives among them. Although an automatic system with 75% accuracy and 80% F-measure may not be sufficient to replace human effort, our approach should offer significant assistance for task identification by reducing users' cognitive load.



	Accuracy	Precision	Recall	F-measure
KI	79.63%	74.37%	91.71%	82.13%
KII	75.93%	71.79%	87.05%	78.69%
KIII	77.25%	70.19%	96.37%	81.22%
KIV	78.57%	72.22%	94.30%	81.80%

Figure 3. Results of Kernel-based Task Extraction

Surprisingly, the simple word kernel K_I outperformed the others in terms of accuracy (79.63%), precision (74.37%), and F-measure (82.13%), except that the kernel K_{III} using unigrams and bigrams of words achieved the highest recall (96.37%). When taking into account bigrams, sequence kernels, K_{III} and K_{IV}, tend to enlarge the similarity scores between task instances and therefore be less conservative in predicting instances as true tasks, as indicated in the lower precisions and higher recalls. Furthermore, POS tags of words in sequences do not seem to contribute much to task identification in our experiments. However, we should not rule out the possibility that our cross-validation evaluation was conducted on a small corpus with only one policy document. Given the limited vocabulary in the test-bed, word similarity may play a dominant role in kernel-based learning. We expect that, when our approach is used to identify tasks from new documents or new domains, word similarity will be less significant whereas syntactic features such as POS tags will become an important complement for kernel-based learning. A larger test-bed and further experiments will be needed to validate this hypothesis in our future research.

However, our current study has the following limitations. In the experiments, we only evaluated the task identification approach on a small corpus of travel policy. There are still some space for improving the performances, especially precision and accuracy. It is worth noting that manual task annotation from sentences is often a costly process and takes significant efforts for annotators. Nevertheless, once we have a model trained on a reasonably good annotated corpus, we do not necessarily need to repeat annotation for every new policy document. The portability of statistical models across multiple domains still calls for further investigation.

Conclusions

In this paper, we extended our previous research on Policy-based Process Mining by developing and validating text mining techniques for task identification, thus dealing with the three challenges, i.e., definition, identification, and extraction of tasks. Our key contribution in this paper is formalizing tasks as relations among three different process components, i.e. resource, action, and data. As a result, we introduced a new text mining application in the context of automatic task identification. We applied four different sequence-kernel-based learning methods to task identification and conducted experiments on real-world policy documents to evaluate their performance. Our experiment results showed that relation-centric task identification gains satisfactory accuracy, thus achieving reasonable success in dealing with the three challenges aforementioned. While the initial results are far from perfect, they should offer valuable assistance to business process developers.

In the future, we plan to continue this research in the following directions. First, we will increase the size of the dataset to further test the performance of different sequence-kernel-based learning methods. In particular, we plan to validate our hypothesis on why simple word kernel method outperformed other methods in the experiment presented in this paper. Second, we will conduct experiments on process policies from different business domains to test the portability of our approach. Finally, we will continue the development of process construction algorithms (supported by visualization tools) to provide further guidance on building process models based on identified process components and tasks.

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 Blanning, R.W. "A formal approach to workflow analysis," *Information Systems Research* (11:1), Mar 2000, pp 17-36.

Bunescu, R., and Mooney, R. "A shortest path dependency kernel for relation extraction," Proceedings of Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT/EMNLP). , Morristown, NJ, 2005, pp. 724-731.

Bunescu, R., Mooney, R., Weiss, Y., Schölkopf, B., and Platt, J. "Subsequence kernels for relation extraction," *Advances in Neural Information Processing Systems* (18) 2006, pp 171-178.

Cobb, C.G. *Enterprise Process Mapping: Integrating Systems for Compliance and Business Excellence* ASQ Quality Press, 2004, p. 128.

Collins, M., and Duffy, N. "Convolution Kernels for Natural Language," Proceedings of Advances in Neural Information Processing Systems 14, MIT, 2002.

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," Proceedings of 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.

Karunakaran, B. "Analysis and Design of a Policy Driven Workflow Model for Product Development," in: *Department of Management Information Systems*, University of Arizona, Tucson, AZ 85721, 2007.

Kohavi, R. "A study of cross-validation and bootstrap for accuracy estimation and model selection," Proceedings of Fourteenth International Joint Conference on Artificial Intelligence, 1995, pp. 1137-1143.

Li, J., Wang, H.J., Zhang, Z., and Zhao, J.L. "Mining Business Policy Texts for Discovering Process Models: A Framework and Some Initial Results," Proceedings of the Sixth Workshop on E-business (WeB 2007), Montreal, Quebec, Canada, 2007.

Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N., and Watkins, C. "Text classification using string kernels," *Journal of Machine Learning Research* (2:3) 2002, pp 419-444.

Madison, D. *Process Mapping, Process Improvement and Process Management* Paton Press, 2005, p. 320.

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, 2000.

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., Zhao, J.L., and Zhang, L.-J. "Policy-Driven Process Mapping (PDPM): Towards Process Design Automation," Proceedings of the 2006 International Conference on Information Systems (ICIS 2006), Milwaukee, Wisconsin, 2006.

WfMC "Workflow Management Coalition Terminology & Glossary, WfMC-TC-1011, Issue 3.0," Workflow Management Coalition, Winchester, Hampshire, UK, p. 65 1999.

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.