Exploring contexts and actions in knowledge processes

Tadej Štajner, Dunja Mladenić, Marko Grobelnik

Jožef Stefan Institute Ljubljana, Slovenia, tadej.stajner@ijs.si

Abstract. This paper presents an approach to automatically discovering the contexts that a knowledge worker is working in and the different types of actions the knowledge worker is executing across contexts. We formulate the two scenarios as two feature selection variations of a clustering problem. The proposed approach is evaluated using realworld data that capture knowledge worker workflow. We evaluate context discovery on gold-standard contexts and action discovery on predictive power of process models, constructed from those actions. The obtained results suggest that we are able to successfully enrich the obtained events log data with additional context and action metadata.

Keywords: Context discovery, Action discovery, Process mining, Clustering, Knowledge worker, Information delivery, Data mining

1 Introduction

Imagine a scenario with a number of knowledge workers in an enterprise who are usually involved in several projects that require accessing different data sources, exchanging messages, browsing the Web etc. With the wide usage of computers in enterprises, one can expect that each knowledge worker has access to a personal computer. The computer can then have instrumentation that records activity on the level of complex events, such as, person P has accessed document D at time t. We will assume that each event is associated to a project and that it is possible to cluster the events so that we automatically identify which events belong to the same project. Each project has data collections associated to it and possibly interconnected with some kind of relation. For instance, a set of people working on the project are related to a collection of documents they have written during the project.

In our scenario, knowledge workers switch from working on one project to the other on weekly, daily or maybe even hourly bases. Ideally, with a press on a button, the desktop should switch context from one project to another showing relevant parts of the associated data collections and supporting the knowledge worker in communication, analysis and decisions.

2 Problem formulation

The problem of identifying parts of data collections relevant for each project can be approached as a clustering problem. For instance, assume that we gather three sets of data.

- 1. A set of people working on the projects, giving some basic information on each person.
- 2. A set of documents written in the projects, having text of each document and a list of its authors.
- 3. A set of events recording that a person has accessed a document at some time.

For example, we would like to cluster the events so that each partition of events is executed in the same project. Since not all work environments are organized in terms of projects, we introduce the notion of a *context* as a generalization. In this paper, we use the term *context* as a grouping of information for a particular need. Following that, a criterion for selecting information from a broader pool of information can be called *contextual model*. In these terms, a good contextual model is the one that provides the most relevant information for a particular need. A context is also characterized by the fact that information resources belonging to that context are accessed concurrently or within a short time span.

In practice, when working within a particular project, a worker may be more interested in information resources pertinent to that project as opposed to any other resources. For example, when a knowledge worker receives an e-mail message from a particular client, we would like to be able to correctly identify the context that this client represents. Given this piece of information, we are then able to confidently retrieve information relevant to that particular client.

In this framework we assume that while knowledge workers are working on one project, they perform actions that are of similar intent across different contexts. For example, reviewing a document, visiting a client web site, responding to a meeting invitation; these are all events which take may occur in multiple contexts. We refer to these context-independent partitions of events as *actions*. To sum up, each event is a trace of an action being executed in a context.

When approaching the domain from a data mining perspective, contexts and actions are reflected as two different partitionings of events using two different selections of features:

- Context mining is the task where we want to discover the different contexts that the knowledge worker is involved in. The contexts are obtained by performing semi-supervised clustering of events, where each cluster represents a distinct context in which the knowledge worker is working. In most work environments, contexts are most often seen as projects or clients, such as *Process mining research project* or *Proposal for client X*.
- Action Mining is the task where we wish to look at the events not in the scope of a context, but rather in the scope of a process. When identifying

actions, we are interested in discovering the distinct steps that the knowledge process is composed of. To achieve this, we perform clustering of a context-free representation of events. When events are stripped of context-describing features and given additional meta data, we are left with clusters, which describe generalized representations of actions, such as *Send e-mail to group of co-workers* or *View intranet website*, which may occur in multiple contexts on different occasions.

Furthermore, sequences of actions within a context appear in patterns from which it is possible to reconstruct a *process model* : a probabilistic model of transitions from one action to another, which can give insight to the dynamics of knowledge processes. From the perspective of knowledge processes, we can consider all events executed in a particular context an instantiation of some process. Therefore, to consider learning a process model from a sequence of actions, we must also be aware of the context that these actions were performed in.

This paper describes the application of clustering techniques on discovery of contexts and actions. We evaluate context discovery with respect to the use case requirements of contextual information delivery and action discovery with respect to the use case requirements of process mining.

3 Related work

Knowledge workers often interleave multiple contexts. However, those switches might not always happen voluntarily. As reported in [3], a notable amount of context switches in a typical knowledge worker environment comes from external interruptions, such as receiving telephone calls or e-mails. They have also identified that recovering from a context switch is considered a difficult task by knowledge workers, noting that the current state of operating systems allows for plenty room for improvement in managing contexts and tasks. One of the possible solutions they have identified as a feasible prototype would be a custom task bar which would be aware of the context the knowledge worker is performing and assist context switching by remembering window layout in each context.

Work on the ACTIVE project [10] continues on the idea of assisting knowledge workers through operating system add-ons. More specifically, one of the focuses is on contextual information delivery to mitigate information overload [5]. Automatic learning of associations between information objects and contexts is also one of the mentioned open research questions.

The relationship between the process and the context has also been discussed in more specific domains, such as medicine [4], where the authors focus on identifying contexts, given that they already have knowledge of a formal process. In contrast, our approach focuses on discovering the models out of raw event streams without an existing process model. Another interesting approach to identifying different contexts using the sequence information has also been shown as useful especially when dealing with multiple simultaneous processes being executed [1]. Context and task discovery and classification was also identified as a crucial component to enable effective e-learning while executing informal knowledge processes [8].

4 Context discovery

To obtain a contextual model, we resort to automated ways of discovering contexts out of event logs. At the data level, we can distinguish between different contexts by different content keywords, resources and people, involved in the events. This means that context discovery uses literal names and affiliations of people as features, as well as the literal contents of the document, since this are the features we have determined to be important for describing a context.

Furthermore, the knowledge workers that participated in the data acquisition also had the ability to assert the context that they were currently working in. This is later used in evaluation of context discovery: since we use normalized mutual information (NMI) [2] to evaluate the quality of the resulting clustering, we also require a gold-standard labelling to which we compare our obtained clusters. Ideally, the clustering would correspond to this gold-standard labelling. NMI measures the degree to which the mined clusters correspond to the goldstandard ones.

Another implementation of context mining technology that is possible besides context discovery is *context detection*. Since knowledge workers' workstations are instrumented, we are able to track their activity and automatically suggest a context switch in case a new event is more similar to another context, further streamlining the contextual information delivery. However, good performance on context discovery is paramount to good performance of context detection.

5 Action discovery

We define actions as atomic steps in executing processes. The events that are logged are in fact manifestations of actions of a knowledge process being executed. This way, actions may be best described as context-independent abstractions of events. In practice, the algorithm to obtain actions is the same as the one to discover contexts with the important difference in the feature sets. For instance, we display actions as feature patterns like *manager sent email to project partner* or *technical consultant prepared proposal*. The following transformations are performed to obtain an appropriate representation of events:

- 1. Person features using only their meta-data, such as organizational role (i.e., manager, researcher, administrative, domain expert), project role (i.e., project partner or not) or a descriptive role (i.e., academic, industry partner) without concrete identifiers.
- 2. Identify whether the event involved only one person, two people or a group.
- 3. Identify whether the people involved with the event are within the same institution, with another partner institution or from multiple different institutions.

4. Extract the named entities using Enrycher [9] from the textual content and remove their mentions within the content. This is done to remove the references to concrete people or organizations which are more likely to be associated to a particular context, which we want to avoid.

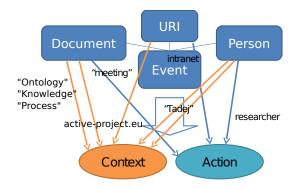


Fig. 1. Different feature representations of the same data for context and action discovery

Figure 1 illustrates the two distinct data representations for clustering in order to obtain two different sets of clusters for the same event log. As with contexts, we also obtain actions by clustering the event data. The difference between the approaches is only in the feature selection. Our concrete implementation uses k-means clustering to obtain the event partitions. The action definitions are then visualized as features of the cluster centroid with the highest weights.

6 Experiments

In this section, we describe the process used to transform raw logs from different sources into a common *TNT* (text, network, time) event model [6]. We outline the required transformation steps and describe the additional background knowledge used. An actual example data set of real-world knowledge worker activities was collected from instrumenting the workstations of three knowledge workers within a large telecommunications company for two months. The instrumentation logs activity on productivity tools: Microsoft Office Word, Excel, PowerPoint, Outlook and Internet Explorer. Using application add-ons, the monitor gathers the following types of events along with their associated content: web page navigation events, opening or saving a document or viewing, sending or replying to an e-mail message. We then generalize the different types of events to a common framework of text, social network and time. In this domain, we can look at events as data points, containing the following sets of features:

- 1. Content, associated with the event (the document, the website, the e-mail message)
- 2. Time and type of the event (navigate to web page, view or send e-mail, open document)
- 3. The people, associated with the event (e-mail sender, e-mail recipients, institution)

The different feature construction approaches for particular applications are described in the following sections. The actual clustering algorithm is used as follows: first, we represent data points as feature vectors including textual and social network data with an additional temporal dimension of the event. The similarity function is therefore defined as a weighted sum of text, social network and time similarities. More precise, individual similarity functions are:

Text: cosine similarity over TF-IDF feature vectors (each feature is a word);

Social network: cosine similarity over binary feature vectors (each feature is a person);

Time: exponential decay of time difference -

 $sim_{time}(t_1, t_2) = 1 - \exp(c \cdot (t_2 - t_1))$, c being a damping coefficient.

One of the sub-problems in constructing a good model for a particular use case is selecting appropriate weights for contributions of individual components. We discuss this within evaluation of context discovery.

6.1 Evaluation of context discovery

The main motivation for context discovery is improving information resource delivery. However, we first need to be certain that the context that we obtain correspond well to the contexts in the users' minds.

The feature set for context mining includes the explicit text, social network and time features from the events: the bag-of-words representation of the content and explicit people names for the social network part, followed by the time stamp.

Since we use NMI to evaluate the quality of the resulting clustering, we also require a gold-standard labelling to which we compare our obtained clusters. Ideally, the clustering would correspond to the gold-standard labelling. NMI measures the degree to which the mined clusters correspond to the gold-standard ones.

Our gold standard clusters are manually labelled by the knowledge workers that participated in the data collection. Since k-means is seeded by probabilistic initialization, we report the average result of ten runs. This particular experiment measures the effect of selecting different weights for TNT components and their effect on the resulting clusters, measured via NMI.

Table 1 shows several examples of weight parameters that we have experimented on and demonstrates that are indeed differences in selecting different weights for individual components. However, two rather similar configurations are both significantly better than other while being statistically indistinguishable between each other: the 0.1/0.1/0.8 and 0.1/0.3/0.6 settings, closely (but

6

Social Network	Time	Text	NMI
0.1	0.1	0.8	0.4513
0.45	0.1	0.45	0.4085
0.8	0.1	0.1	0.3791
0.1	0.3	0.6	0.4505
0.35	0.3	0.35	0.4163
0.6	0.3	0.1	0.3875
0.1	0.5	0.4	0.4412
0.25	0.5	0.25	0.4291
0.4	0.5	0.1	0.4108

 Table 1. The quality of context mining given different settings of feature contribution weights

not significantly) followed by 0.1/0.5/0.4. While these numbers are highly dependent on a data set and should not be considered as a final prescription, it suggests that the document text is the most important component in context definition, followed by the time. This is expected since different contexts are often topically different and events in the same context tend to be close in time. A more surprising observation is that the social network information does not convey much information with regard to users' definition of contexts. Upon inspecting the raw data, we found that in our case, there is a tendency for many of the same people appearing in most of the contexts, making it hard to distinguish between contexts based on social network alone. On the other hand, there is still a possibility that some other types of organizations may have contexts which are topically similar, but with strictly distinct social networks.

6.2 Evaluation of action discovery

Besides information delivery, another motivation for discovering actions is constructing a process mode. Since our data set does not have gold-standard labels for actions, we employ an extrinsic measure and evaluate the quality of discovered action definitions by constructing a process model out of them and evaluating that. The process model is constructed from selecting only statistically significant transitions between individual actions. We evaluate the performance of the obtained process models by splitting the actions into training and test partitions, constructing a process model q with the training data and validating it with test data C using cross-entropy.

$$H(C,q) = -|C|^{-1} \cdot \sum_{x \in C} \log_2 q(x)$$
(1)

This metric will then give us insight on the predictability of the obtained process model. The lower the cross-entropy, the more predictable is the process model. The entropy score for a given process model and a test event log is the one where the probability distribution of q(x) is completely uniform, meaning that the process is random. Therefore, our goal is to construct a process model with minimal cross-entropy given a test event log. Besides looking at the effect of action definition on process model construction, we also inspect the influence of pruning the process model on the quality of the model.

Since one of the use cases this research supports is information resource delivery, we examine the possibility of using action-awareness. To put it concretely, we will focus on using the process mining results to enable resource prediction based not only on context, but also the users point in the process. We wish to enable technical experts to locate other solution designs that are similar to, or have similar elements to, the solutions that are currently being worked on. The actions themselves are obtained by performing action mining and reducing the individual primitive events to only their action label.

In the sense of information resource delivery, this gives us two use cases for resource suggestion:

- Suggesting a resource from the same context. For instance, when preparing a bid for a client, suggesting a clients request for proposal.
- Suggesting a resource from action prediction. For example, lets assume that we have a process model where we identify that preparing a bid is often followed by preparing technical documentation. Therefore, when preparing a bid, suggest other similar bids and technical documentation resources, also from other contexts.

We propose providing the output of this model via a custom desktop task bar suggesting a list of resources that will probably be used needed by the user in the forthcoming actions.

The experiments focus on predictability of the mined process. We have done this in order to observe the feasibility of using the process data to predict actions, which is a necessary step given that we will use this for resource suggestions. Since the definitions of actions are not given and need to be discovered, we predict that they may be significant discrepancies between process models constructed from different actions. For instance, given two different sets of actions, we also have different action sequences which lead to different process models. One of the tasks of this experiment is to find out whether there is any obvious most appropriate k value which would result in a process model which best describes the process with the best possible predictability.

The experimental setting is as follows: we wish to observe the different effects of action and process mining on the predictive power of the obtained process model. Since different definitions of actions may produce different patterns, we need to consider the predictability of those patterns.

Since the obtained model can be very noisy and dense due to either intrinsic irregularities in knowledge processes or due to noise in context or action discovery, we need to determine whether the existence of a particular transition between states is sensible enough so that the added complexity does not outweigh the improved coverage. A practical solution to this problem is to prune the model so that it is less complex, but still retains useful coverage of the log while avoiding reporting of false discoveries. To achieve this, we prune all transitions within the model that are not statistically significant within a certain error rate and given a sample size using statistical sequence mining techniques to determine constraints for inclusion of individual transitions in [7].

Since manually selecting a good threshold for including a transition in the model is challenging one does not have an estimate of data quality, a benefit of using a statistical approach is that the only parameter that the process analyst needs to specify is the risk factor, which corresponds to the expected false positive rate and is easier to understand than some arbitrary probability threshold. In practice, the advantage of this approach is that it gives us the ability to express ourselves in terms of allowed error rate risk (α).

We apply the *proportion constraint* :

Let $w = \langle x_1, \ldots, x_l \rangle$ be a pattern of actions and q_0 the beginning action and P(q, w) the probability that a path that starts in state q contains the pattern w. Given a risk factor α and a event log size N, the proportion constraint only allows the patterns which satisfy the following constraint:

$$P(P(q_0, w) > k) \text{ iff } k = z_{\alpha_1} \sqrt{\frac{P(q_0, w) - (1 - P(q_0, w))}{N}}$$
(2)

 z_{α_1} is the $(1-\alpha_1)$ -percentile of the distribution of p(w), a normal distribution in our case. Since manually selecting a good threshold for including a transition in the model is challenging when we do not have an estimate of data quality, a benefit of using a statistical approach is that the only parameter that the process analyst needs to specify is the risk factor, which corresponds to the expected false positive rate and is easier to understand than some arbitrary probability threshold.

We measure the predictability of the model in terms of the percentage of transitions that were valid within the obtained model, varying the number of actions (k in action mining) and the allowed error rate (α). The values were obtained as averages on five-fold cross-validation. The data set used is a log of knowledge worker activity containing 15384 events for three knowledge workers over a period of two months.

Figure 2 shows that while the predictability varies quite a lot within different k values, it varies to a smaller degree across different allowed error rates. The latter behaviour is desirable; it suggests that pruning the model (and therefore simplifying it) does not have a too adverse effect on prediction performance. As observed in Table 2 and Figure 2, the best performing model for this particular dataset with regard to k is when k = 11, exhibiting around predictability of roughly sixty per cent. Its performance is also one of the examples that are relatively invariant to pruning. In general, this method may not be a generally practical method for selecting a good k value for clustering, but it shows to illustrate the fact that a particular example of partitioning may have a significant effect on the predictive power of the process model.

Since we are also interested in minimizing complexity of the process models, we also show the expected number of transitions within a model.

k	0.05	0.1	0.25	0.4	0.5
5	0.22	0.28	0.42	0.41	0.54
6	0.47	0.47	0.49	0.49	0.49
	0.03	0.08	0.13	0.09	0.18
8	0.08	0.08	0.05	0.11	0.12
9	0.13	0.24	0.38	0.42	0.34
10	0.35	0.35	0.55	0.44	0.4
11	0.58	0.6	0.58	0.6	0.59
12	0.13	0.15	0.18	0.29	0.23
13	0.39	0.4	0.43	0.5	0.51
14	0.46	0.39	0.42	0.47	0.44
15	0.08	0.06	0.1	0.11	0.1
16	0.37	0.42	0.49	0.48	0.45
17	0.32	0.43	0.36	0.41	0.4
18	0.33	0.36	0.3	0.4	0.38
19	0.27	0.31	0.35	0.35	0.41
20	0.3	0.37	0.34	0.42	0.43

Table 2. The predictive power of different process models with varying number of actions and pruning parameters.

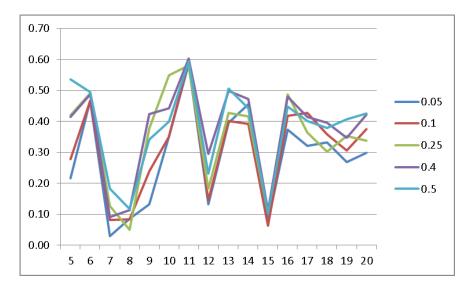


Fig. 2. The plot of predictability with different k and α ; note the peak at k = 11

As expected, the number of transitions in the model grows with number of actions and drops with stricter pruning parameters, as shown in Table 3 and Figure 3. In practice, we prefer to have a compromise between complexity and predictability a complex model might describe a process very well but will be

k	0.05	0.1	0.25	0.4	0.5
5	11.92	17.82	21.02	25.38	26.46
6	20.04	22.26	26.36	30.1	30.76
7	16.88	18.94	27.64	27.82	29.84
8	21.8	24.04	33.04	37	36.66
9	33.32	41.54	49.92	52.26	52.48
10	52.3	57.56	66.3	66.6	65.02
11	36.88	55.5	63.14	68.68	72.52
12	41.92	48.62	58.74	66.78	64.78
13	52.98	56.72	70.78	78.24	79.76
14	55.36	60.58	74.8	83.54	88.02
15	24.46	30	43.64	45.1	45.8
16	48.08	60.56	78.28	85.66	88.2
17	53.3	62.18	78.2	86.84	93.28
18	51.56	60.62	77.56	89.18	96.4
19	63.96	74.34	100.16	110.06	116.06
20	53.98	64.34	81.54	98.4	102.56

Table 3. Complexity: average number of transitions in the model with regard to number of states and pruning parameter

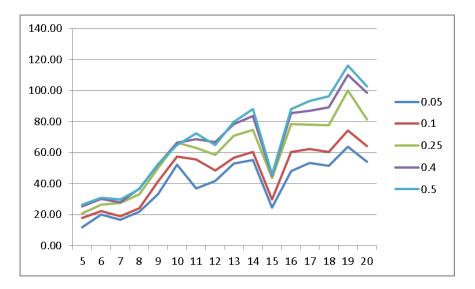


Fig. 3. Model complexity (number of transitions) with regard to k and α

hard to interpret, whereas a simple model might not have enough correspondence with the data itself.

Figure 4 visualizes the trade-off between predictability and complexity as a ratio between them. It turns out that at the most optimal k, where pruning

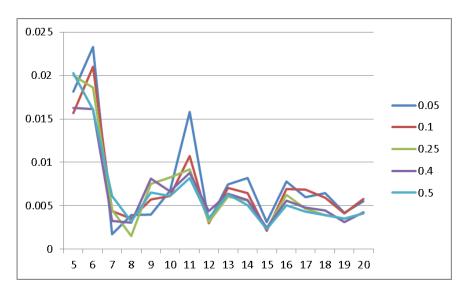


Fig. 4. The ratio of predictability vs. complexity with regard to k and α

does not affect predictability to a great extent, the pruned model is also one of the most desirable ones. This metric also favours very simple models with k = 5 or k = 6, although the actions obtained via those are often too coarse for interpretation.

In Figure 5, we show the visualization of the process model, constructed with the parameters which achieve the highest predictability score. The obtained models show some regularity. For instance, the transitions between actions related to internal communication inside the company have higher transition probabilities between each other than to other actions (e.g., such as actions related to project consortium communications). Whereas e-mail communication actions repeat in shorter bursts, the web browsing navigation actions were in much longer sequences.

These measurements show that selecting different k values has a very big effect on the predictive power of the model. In terms of interpretation of the model, it is often worth pruning the models as it does not have a too adverse effect on predictive power.

7 Conclusion and future work

Evaluation of context discovery shows that the problem of selecting good weighing parameters for individual TNT components is indeed important. Results show that context in this particular case study dataset are mostly determined by document content and time of access, followed by social network. A valuable extension on this front would be an automatic way to determine those contri-

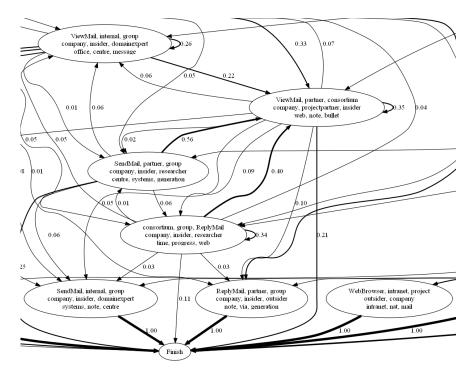


Fig. 5. Fragment of a visualization of a process model, constructed with the bestperforming parameters for this dataset $(k = 11, \alpha = 0.05)$;

bution weights, possibly by machine-learning-assisted feature selection. In terms of absolute value, the obtained clusters may not be entirely overlapping with the gold standard ones, but a NMI in the range of 0.44 already demonstrates relatively useful correspondence of partitionings.

The evaluation of processes shows that obtained processes are predictable, which is useful for using the underlying models for predictive behaviour. We have found that determining a good k in action mining has a big influence on the predictability of the obtained process model. In the most optimal scenario, we observe relatively high predictability of around sixty per cent. Moreover, we show that pruning of probabilistic process models does not have a too adverse effect on predictability and proves to be an effective way of making a compromise on ease of interpretation for better predictive power.

This paper described the complementary use cases of action and context discovery from raw event logs obtained from instrumenting common knowledge worker tools. Demonstration on real data shows that we can interpret several patterns using this model. For instance, the transitions between project-related actions are more common than transitions to administration-related actions. Also, web browsing events tend to have longer homogeneous action sequences than e-mails.

The feedback from the testing of the implementation by the knowledge workers that have provided the dataset has shown that the quality of context discovery and switch detection is paramount for wider acceptance, since it directly affects their workflow by offering switch suggestions or context-dependent information resources, such as files, associated with the context.

Future research in this area will include using a complex graph representation of data so that we can avoid flattening the social network structure into event features. To take advantage of the structural information and to correctly handle differences in distributions across people, events and resources, present in the multi-relational representation, we will need to employ multi-relational clustering algorithms which are able to handle such datasets. This sort of approach may not only improve the clustering quality, but also report the clustering output of people and resources, respectively.

We expect that the impact of the proposed approach will materialize in several incarnations. The first one is introducing an contextual information delivery mechanisms, enabled by context mining that would enable a semi-automated context mining and switching functionality for the end user. The second goal is expanding the applicability of action and process mining into an analytic environment for managers to observe the dynamics of their processes without additional task management infrastructure. The third application involves using the obtained models to aid user with suggesting action- and context-specific resources or text fragments when editing a particular document or a message.

8 Acknowledgements

This work was supported by the IST Programme of the EC under ACTIVE (IST-2008-215040) and PASCAL2 (IST-NoE-216886).

References

- Bose, R., van der Aalst, W.: Context aware trace clustering: Towards improving process mining results. In: SIAM International Conference on Data Mining. pp. 401–412 (2009)
- 2. Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schutze: Introduction to Information Retrieval. Cambridge University Press (2008)
- Czerwinski, M., Horvitz, E., Wilhite, S.: A diary study of task switching and interruptions. In: Proceedings of the SIGCHI conference on Human factors in computing systems. pp. 175–182. ACM (2004)
- Ghattas, J., Peleg, M., Soffer, P., Denekamp, Y.: Learning the context of a clinical process. In: Business Process Management Workshops. pp. 545–556. Springer (2010)
- Gomez-Perez, J., Grobelnik, M., Ruiz, C., Tilly, M., Warren, P.: Using task context to achieve effective information delivery. In: Proceedings of the 1st Workshop on Context, Information and Ontologies. pp. 1–6. ACM (2009)

- Grobelnik, M., Mladenić, D., Ferlež, J.: Probabilistic Temporal Process Model for Knowledge Processes: Handling a Stream of Linked Text. Proceedings of SiKDD 2009 (Conference on Data Mining and Data Warehouses) (2009)
- Jacquemont, S., Jacquenet, F., Sebban, M.: Mining probabilistic automata: a statistical view of sequential pattern mining. Machine Learning 75(1), 91–127 (2009)
- Lokaiczyk, R., Faatz, A., Beckhaus, A., Goertz, M.: Enhancing just-in-time elearning through machine learning on desktop context sensors. In: Proceedings of the 6th international and interdisciplinary conference on Modeling and using context. pp. 330–341. Springer-Verlag (2007)
- Štajner, T., Rusu, D., Dali, L., Fortuna, B., Mladenić, D., Grobelnik, M.: Enrycher
 Service oriented text enrichment. Proceedings of SiKDD 2009 (Conference on Data Mining and Data Warehouses) (2009)
- Warren, P., Kings, N., Thurlow, I., Davies, J., Brger, T., Simperl, E., Ruiz, C., Gmez-Prez, J., Ermolayev, V., Ghani, R., Tilly, M., Bsser, T., Imtiaz, A.: Improving knowledge worker productivity the ACTIVE approach. BT Technology Journal 26(2) (2009)