Information Resource Recommendation in Knowledge Processes

Tadej Štajner
Dunja Mladenić
Marko Grobelnik

Jožef Stefan Institute
Ljubljana, Slovenia

Introduction

Motivation

- Knowledge workers in modern work environments often suffer from information overload;
- Can we learn from their behaviour and assist them with retrieving information that they need in that point in time?

Goal

- Proactively assist knowledge workers with their workflows by **suggesting relevant information resources** by learning their knowledge process.

Introduction

Knowledge workers:

People whose work consists of manipulating information resources

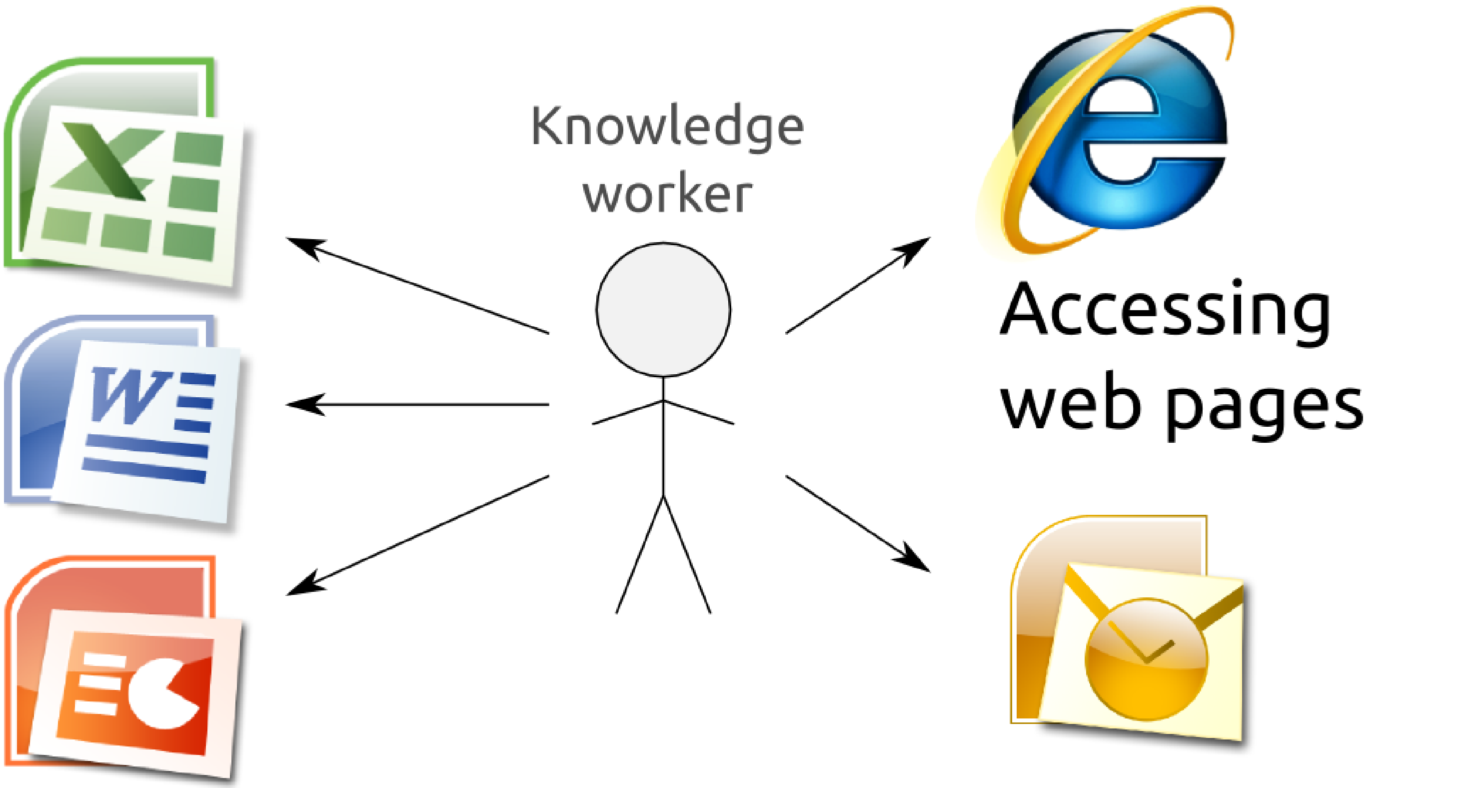
Information resources:

Atomic information objects of the work domain

Knowledge process:

A model that describes what kind of knowledge resources a user could need given his current situation

Knowledge work domain

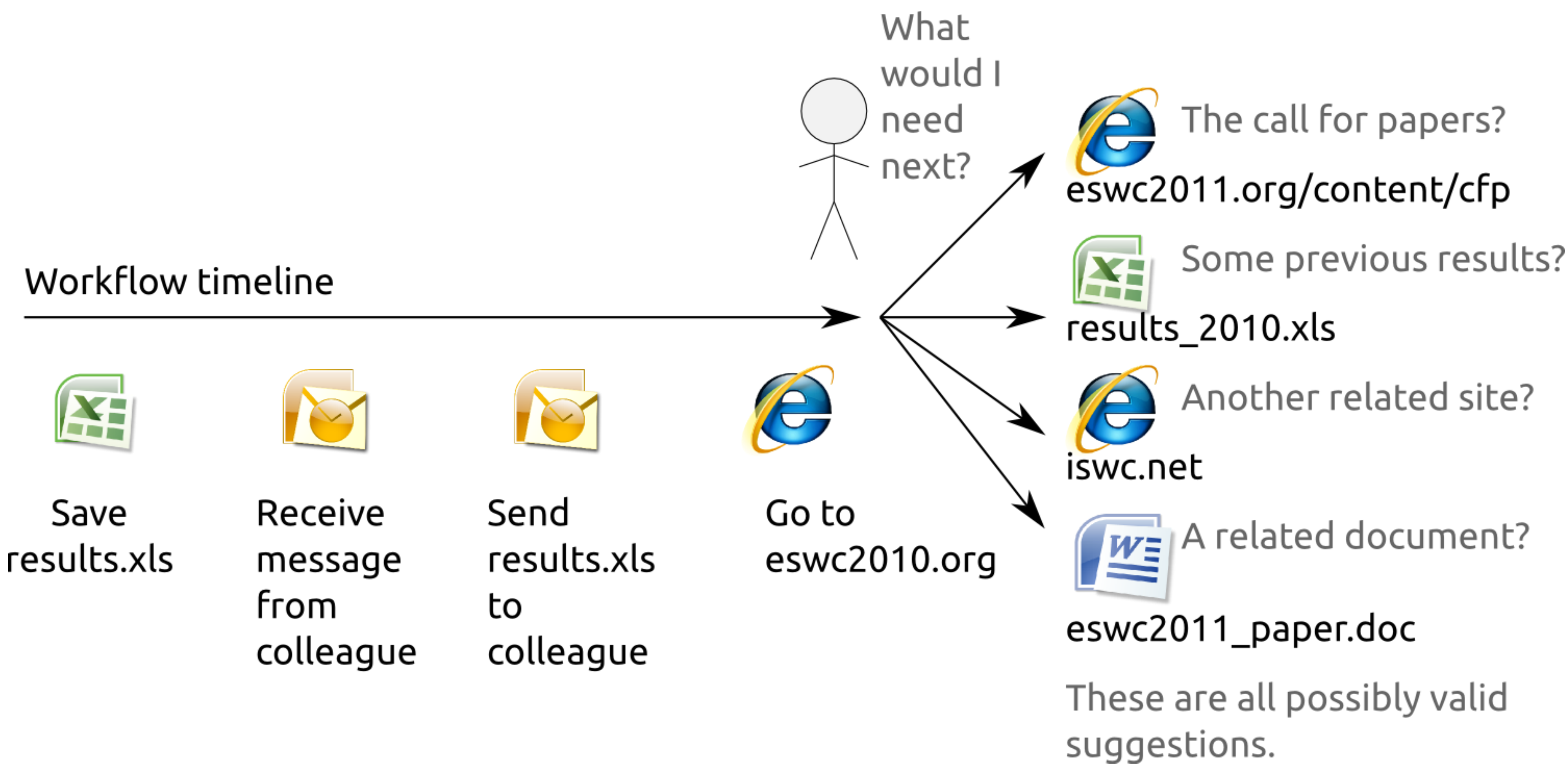


Manipulating
Office documents

Reading and
writing e-mail

... these are all **information resources**.

Situation


What would I need next?
Workflow timeline
Save results.xls
Receive message from colleague
Send results.xls to colleague
Go to eswc2010.org
The call for papers?
eswc2011.org/content/cfp
Some previous results?
results_2010.xls
Another related site?
iswc.net
A related document?
eswc2011_paper.doc
These are all possibly valid suggestions.

Problem description

Provide relevant ranking of information resources

- Given that we are monitoring the user's workstation and know what information resources were accessed when;

Constraints:

- Learn only from usage logs, without explicit user supervision
- As opposed to classic process models, actions are not well-defined.

Knowledge process

A model that describes what kind of information resources a user could need given his current situation;

How can we use it?

- Given a user's current sessions of resource usage, provide a probability distribution on which information resource d would be used next:

$$P\left(d_i \middle| d_{currentsession0},\ldots d_{currentsessionj}\right)$$

- There are several possible ways on how to estimate that

Data

- Every manipulation of an information resource is a TNT event, having these basic properties:
    - Text content of the resource
    - Network (social network context – i.e. e-mail recipients)
    - Time of occurrence
- Events are partitioned into sessions.

Knowledge process framework

We separate the concerns into three sub-models:

- **Event model**: how to represent event features?
- **Action model**: how to represent individual steps within a process?
- **Process model**: how to represent the transition probabilities between actions?

Event representation model

- We represent events in a vector space model;
- Feature construction:
  - Each property of event is a feature
    - Event type (send, receive, save, ..)
    - Media type (document, e-mail, web site)
    - Social roles of participants (inside or outside of organization, manager, developer, researcher, private or mutiple people, single or multiple organizations)
    - Bag-of-words of resource content
  - Weighed using the TF-IDF scheme.

Event representation model (2)

- Alternative representation: we can also encode the features of events in the same session within an event;
- Feature-based with session information
  - Along with its own features, concatenate features of events within the same session.
  - Crude but efficient way to encode the knowledge process

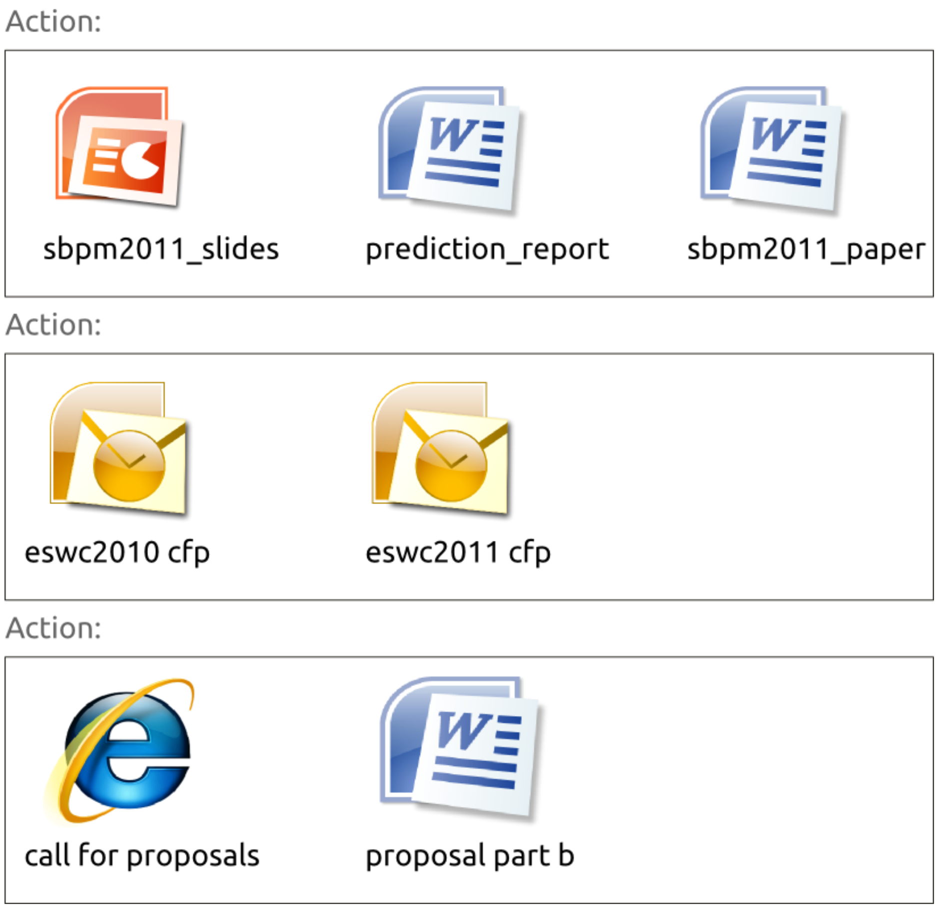
Action model

- How to efficiently represent the actions in the knowledge process so that is provides relevant feedback and is easy to compute?
- Problem: We have high dimensionality in event features.
  - Approach 1: automatically construct action definitions out of data by clustering events, reducing the dimensionality of the feature space;
  - Approach 2: assume conditional independence of individual event features to make computing the probability of candidate resources tractable (remove infrequent features).

Action model by clustering (1)

- Cluster all known events into *k* clusters;
- We treat the cluster definitions as actions
  - The membership of event in a cluster denotes its action
  - From this point on, we only view at the cluster that an event belongs to;
- Result: the process model now needs to model transition probabilities only between *k* different actions;

Action model by clustering (1)



- Once we construct the clusters, we only consider the cluster membership of the events;
- In the example, the events can be reduced to three categories (actions)

1. scientific paper
2. call for papers
3. proposal

Action model by independent features (2)

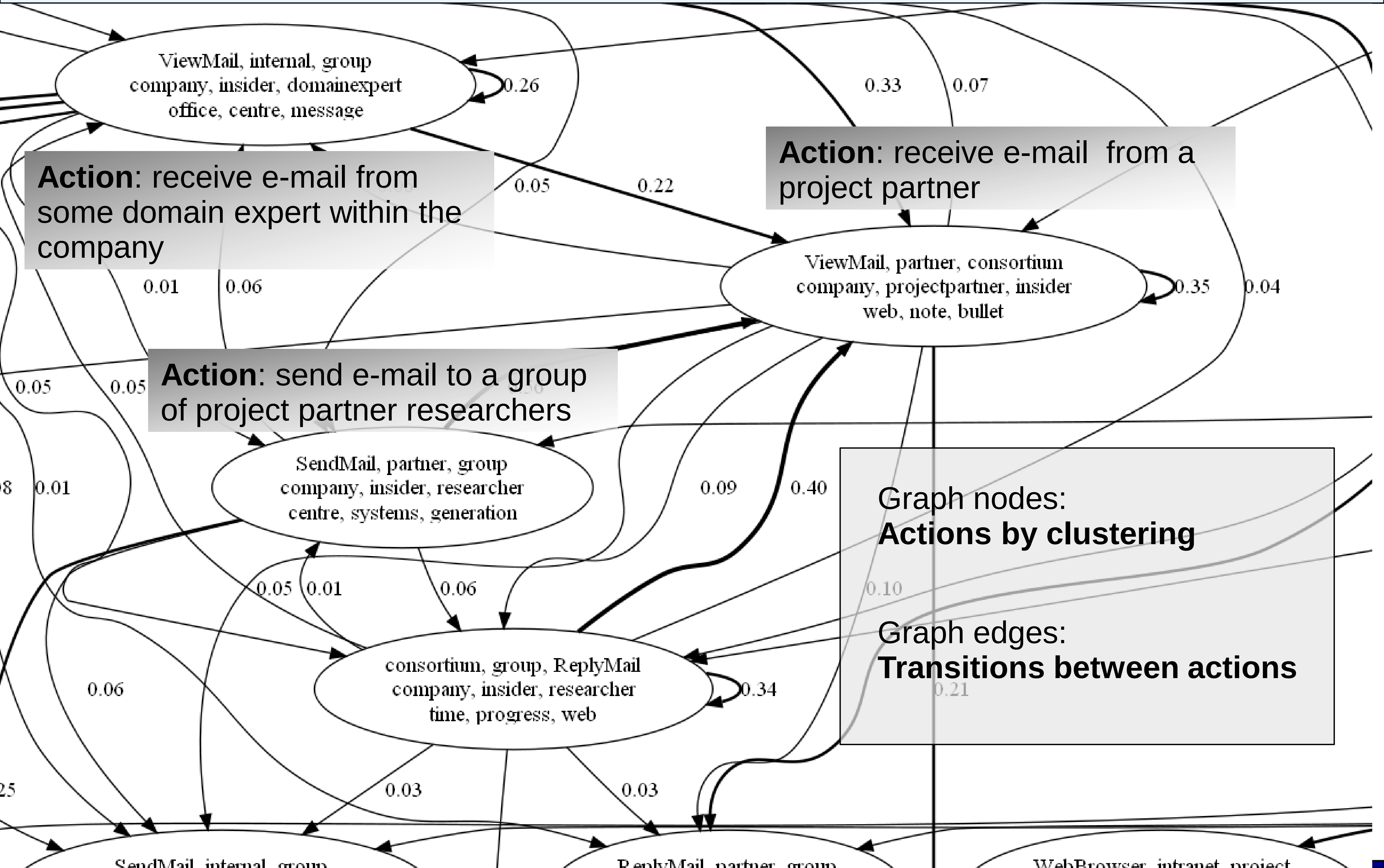
- How is this different from the action model by clustering?
  - We do not assign a single action to an event;
  - We assume conditional independence between two features co-occurring in the same event;
  - We model the process on probabilities of transitions from event with a feature f_i to another event with a feature f_j.

- Result: the process model now needs to model transition probabilities only between *m* different features.

Process model

- How to model the transition probability of one action to the next one?
  - Using the Markov model over actions we can predict which action is the most likely successor;
- Problem: when predicting using conditional probabilities, we must not have zero probabilities.
  - Solution: Laplace (add-one) smoothing $P(a_i \mid a_j) = \frac{1 + c(a_i a_j)}{K + c(a_j)}$
  - *c(ab)*: number of occurrences transitioning from a to b
  - *c(a)*: number of occurrences of a
  - *K*: number of distinct actions

Process model (example)



Ranking

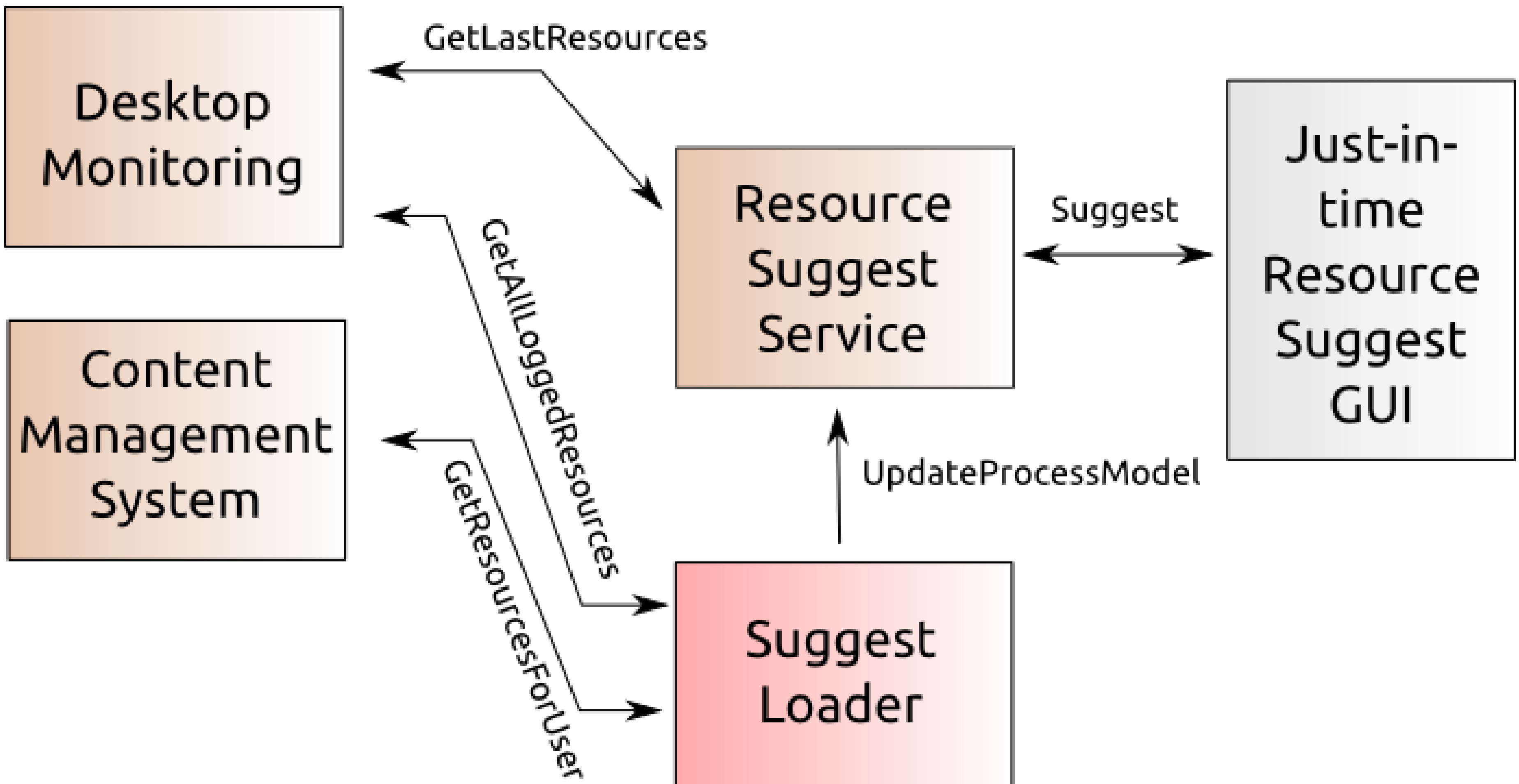
- Given that we have a probability distribution over the actions that will follow, how do we translate that to concrete information resources?
- Given a user's session, for each candidate resource, we combine the following:
  - The probability of the action that the resource would represent (computed using the process model);
  - The average similarity between the candidate resource and the other resources in the session;

Implementation

- reading a documentation of a backend system for process mining

- opening a technical presentation on machine learning

Architecture



Experiments

- Data: 31182 events from three knowledge workers in a telecommunications company within three months;
- Partitioned into sessions;
- Testing scenario:
  - Use a subset of sessions for training, remainder for testing (10-fold cross validation);
  - When testing, take a subsequence and withhold the last event;
  - Using the approach presented, get a ranking over all candidate resources;
  - Observe the rank of the correct resource (that was withheld);

Metrics

- Based on the rank r of the resource for observation i.
  - *N*: number of observations
  - *c(r >= k)*: number of observations where correct resource is in top k
- Mean Reciprocal Rank

$$MRR = \frac{1}{N} \times \sum_{i=0}^{N} \frac{1}{r_i}$$

- Percentage of correct result in top k elements

$$P_{TopK} = \frac{c(r \geq k)}{N}$$

Evaluation set-ups

Event models

IDF: standard feature representation using IDF weighing;

SessionIDF: including features of events within same session (history)

Action models

Clustered-k: define actions by clustering using k as number of clusters

Independent: assume conditional independece of features

Process models

None: baseline – every action has same probablity

Laplace: Markov-model Laplace-smoothed process model

Results

Event Model	Action Model	Process Model	Reciprocal rank	Percentage in top 20
IDF	Independent	None	0.0612	0.2220
IDF	Independent	Laplace	0.0803	0.2377
IDF	Clustered:10	None	0.0794	0.2697
IDF	Clustered:10	Laplace	**0.1076**	**0.3485**
IDF	Clustered:30	None	0.0853	0.3081
IDF	Clustered:30	Laplace	0.0797	0.2490
SessionIDF	Independent	None	0.0774	0.2895
SessionIDF	Independent	Laplace	0.0750	0.2674
SessionIDF	Clustered:10	None	0.0756	0.2807
SessionIDF	Clustered:10	Laplace	0.0701	0.2384
SessionIDF	Clustered:30	None	0.0832	0.3013
SessionIDF	Clustered:30	Laplace	0.0874	0.3051

Too many clusters increase sparsity

History context instead of process model

Conclusions

- Best scenario: standard feature representation, relatively low number of clusters, using a process model
  - We are able to put the correct resource in the top 20 list over one third of occasions
- Using the process mining we can not only predict resources, but also have a look at how the workflow takes place;
- Using session information within an event model (SessionIDF) is in some cases better than standard feature representation, but still below the best performing setup
  - Slightly lower performance, but very simple implementation

Future work

- Expand event model to more than a vector space model
  - The events can be viewed as nodes in a graph with people, resources and other entities;
  - Issue with current approach: flattening to vector space loses information;
  - Employ machine learning techniques that natively work on complex graph data;
  - Complex graphs are much closer to semantic representations;
- Evaluate the approach in a contextual recommender system setting