SWISH:
Semantic Analysis of Window Titles and Switching History

Nuria Oliver, Greg Smith, Chintan Thakkar, Arun Surendran
Microsoft Research
Redmond, WA USA

Microsoft Research

Outline of the Talk

- Introduction and Motivation
- System's Architecture
- Automatic Window Clustering
  - Semantic Clustering
  - Temporal Clustering
  - Ensemble Classification
- Experimental Results
- Applications to Intelligent User Interfaces
- Conclusions and Future Work

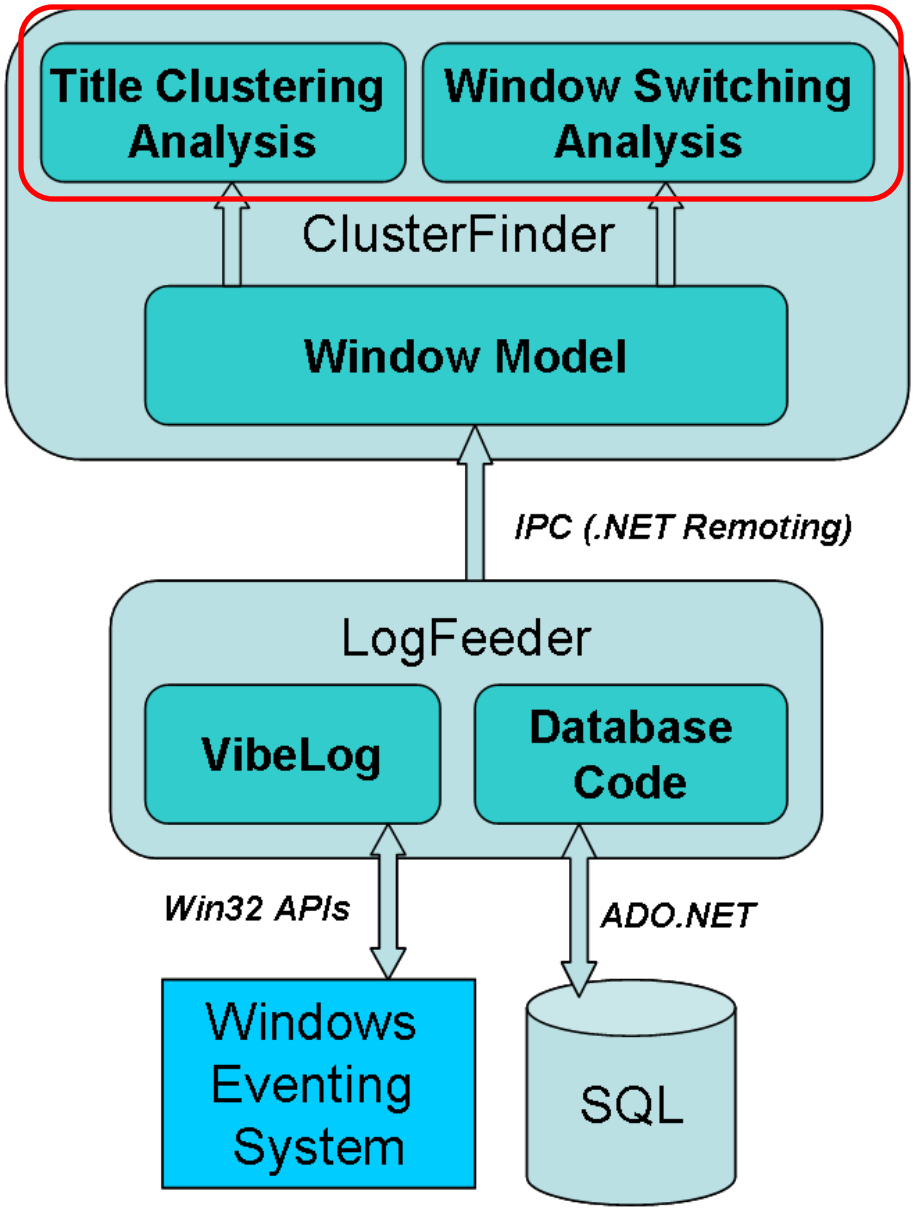
Automatic Window Clustering

- **Goal:**
  - Assist users in managing their tasks
  - Address the *task assigment* problem → automatically figuring out which window on the user's computer belongs to which task
- **Assumption:**
  - Windows belonging to the same task share some common features that can be identified from data
- In this paper we focus on:
  - **Window title-based clustering** ("Re: Dad, get me Potter-6" **&** "Amazon: Harry Potter and the Half Blood Prince")
  - **Behavior-based (switching history) clustering** (Looking into the MSDN Library while coding in Visual Studio)

SWISH

- Real-time prototype running in the background
- Constantly monitors users' desktop activities using a stream of windows events
- Logs and processes this raw event stream
- Implements two criteria for window relatedness:
  - Semantic similarity of their titles
  - Temporal closeness in the access patterns

SWISH Architecture



Automatic Window Clustering

- Tasks typically involve multiple windows that share some features
- In particular we focus on:

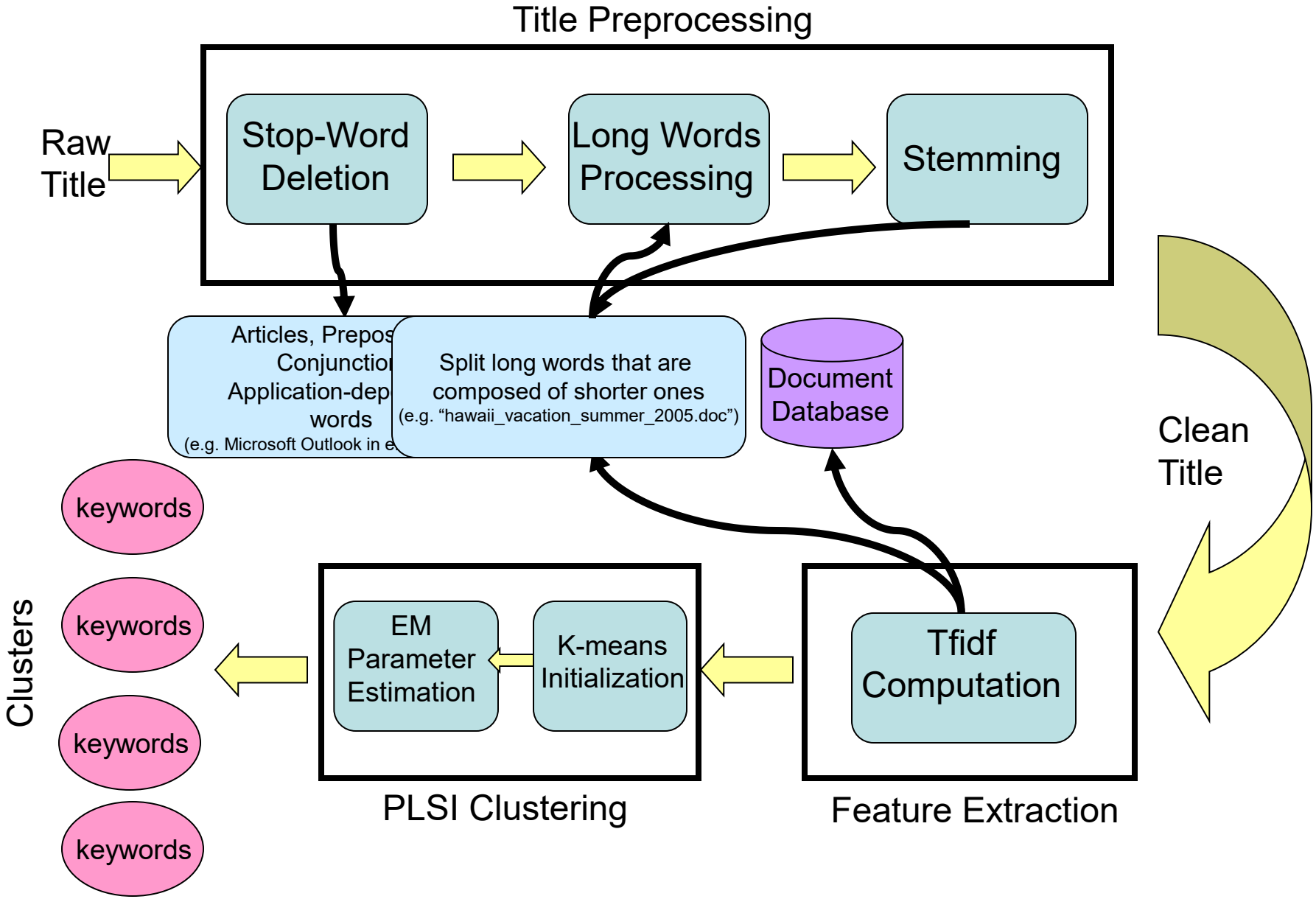
Clustering based on Window Titles

– Semantic Features: Windows belonging to the same task share common words in their content and window titles

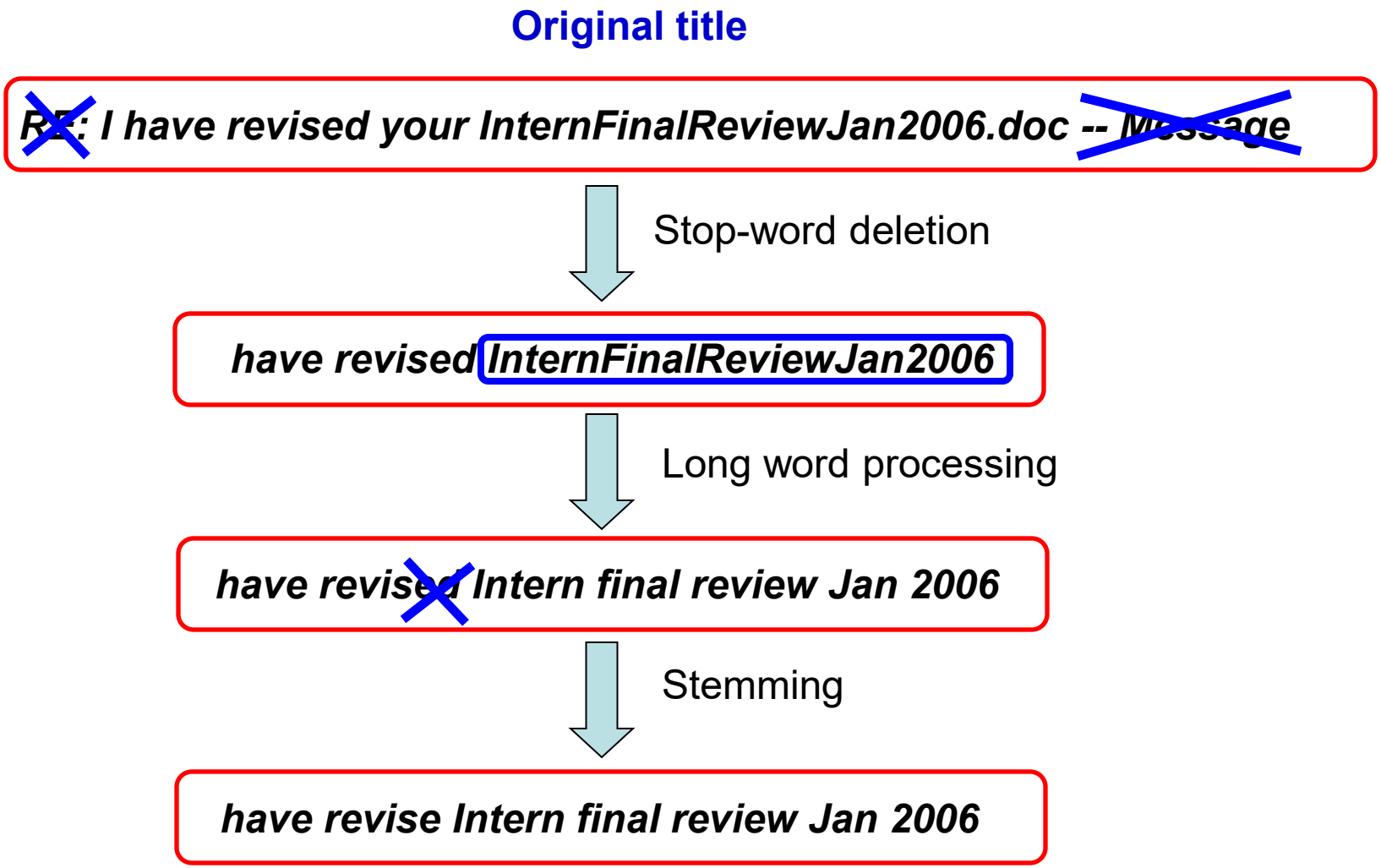
Clustering based on History of Window Switches

– Temporal Features: Windows belonging to the same task are accessed in temporal proximity to each other → focus switches between windows in the same task occur more frequently than switches to windows outside the task

Clustering Based on Window Titles



Title Pre-processing: Example



Feature Extraction

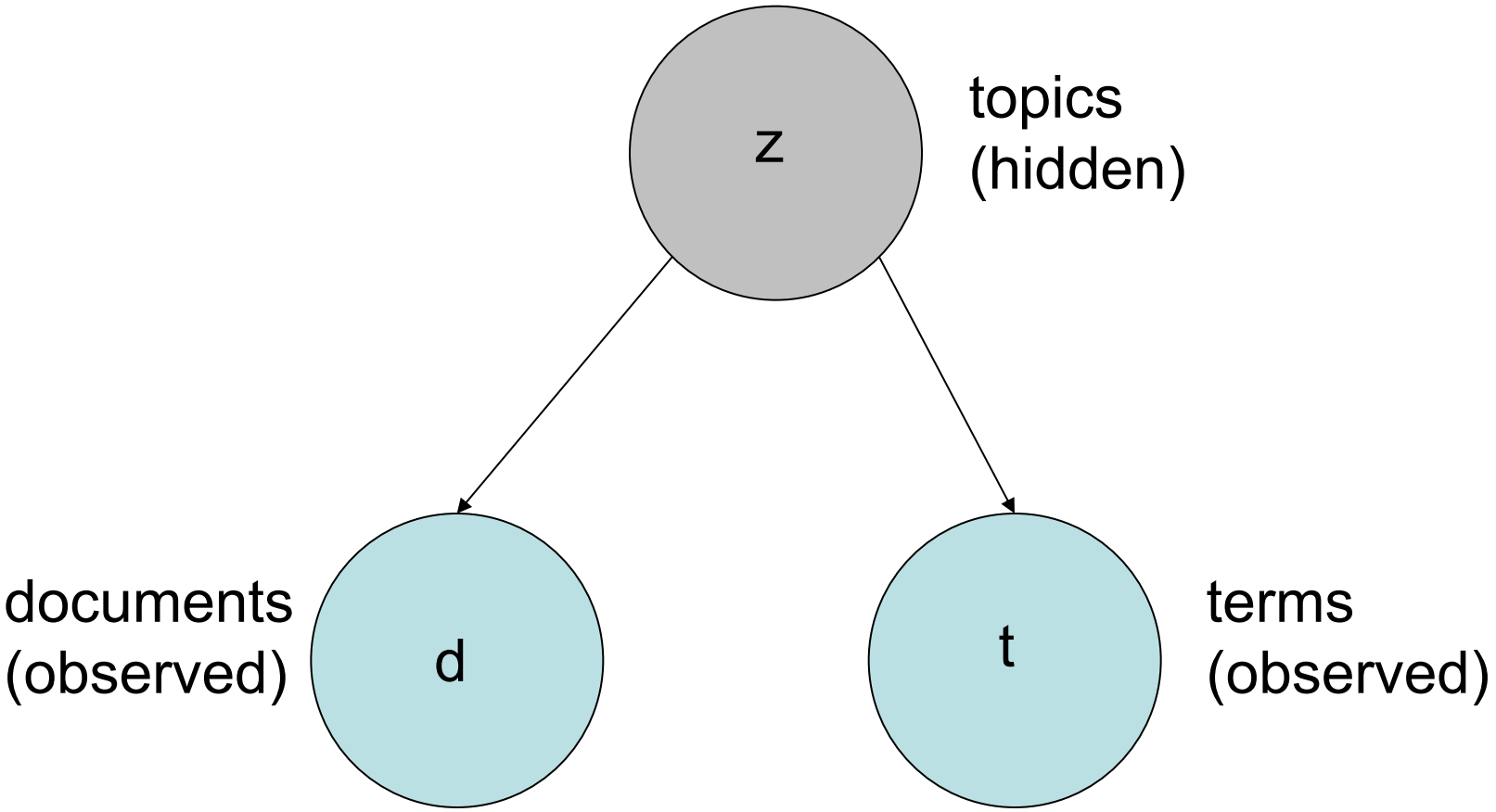
- Inverse Document Term frequency measure

  *tfidf = tf * idf,* where

  - tf_i is the frequency of occurrence of term *i* in the documents
  - $idf_i = \log \frac{|D|}{(|D_i|+1)}$ is the inverse frequency of documents, with
    - |D| = total number of documents
    - |Di| = number of documents containing term *i*

- The *idf* term scales down commonly occurring terms and scales up words that are rare within the document corpus

Model Learning: PLSI



$$P(d_i, t_i) = P(d_i) \sum_{z \in Z} P(t_j \mid z_k) P(z_k \mid d_i)$$

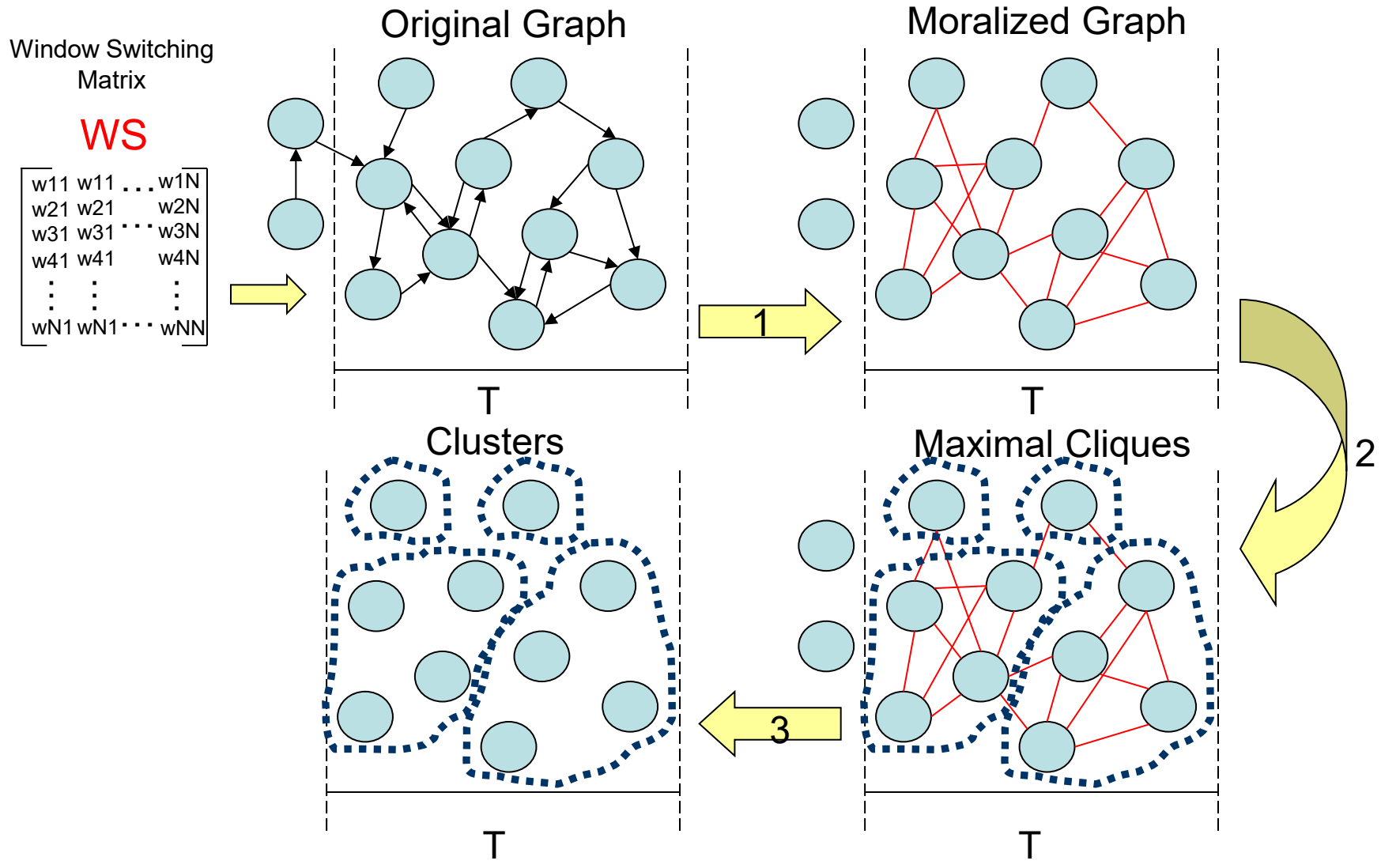
Model Learning: PLSI

- Local optimization of parameters via EM
- Initialization with K-means
- Average performance on 30 runs of the algorithm with 30 different initializations
- Soft clustering technique
- Best performance in our experiements when compared to
  - CEM
  - K-means

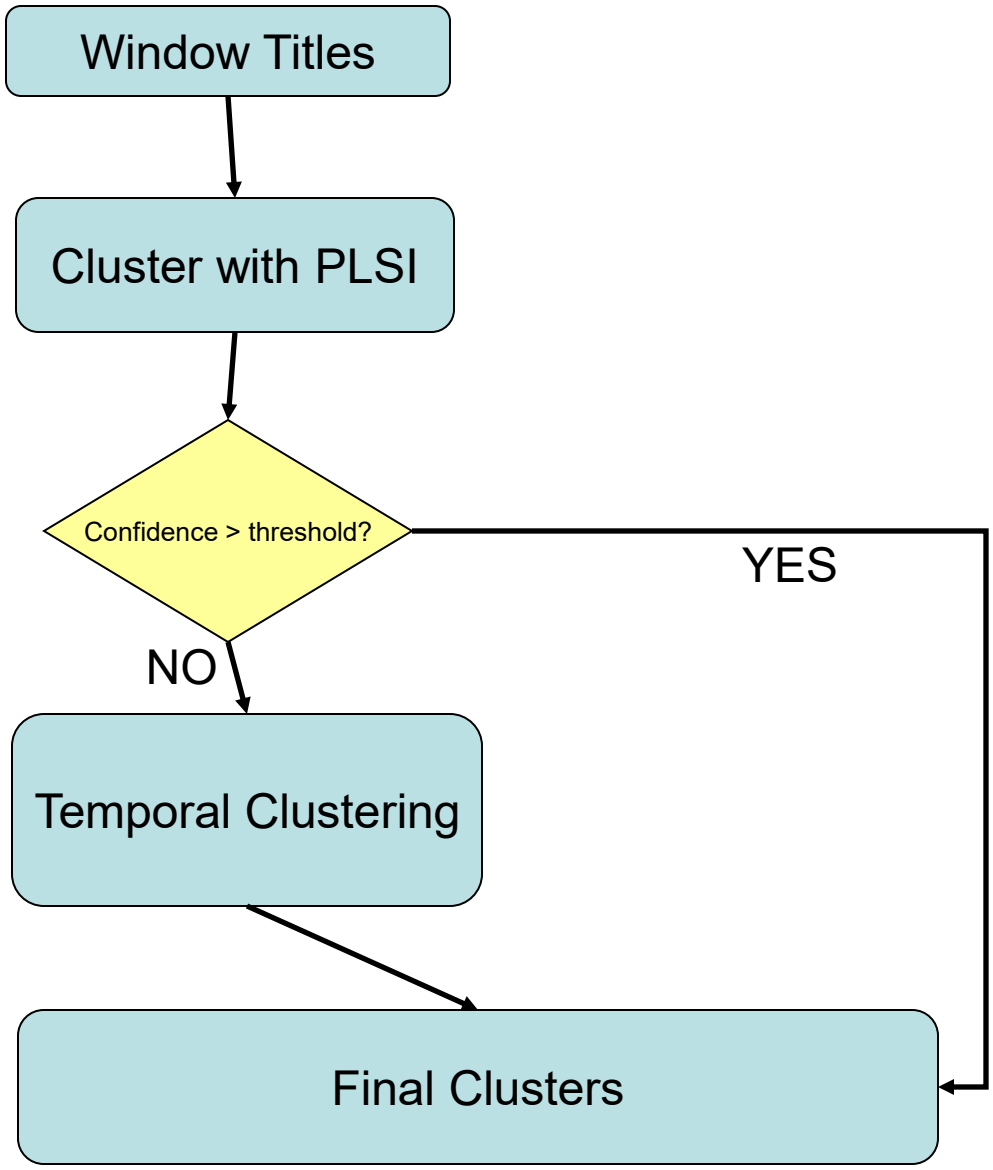
Clustering based on History of Window Switches

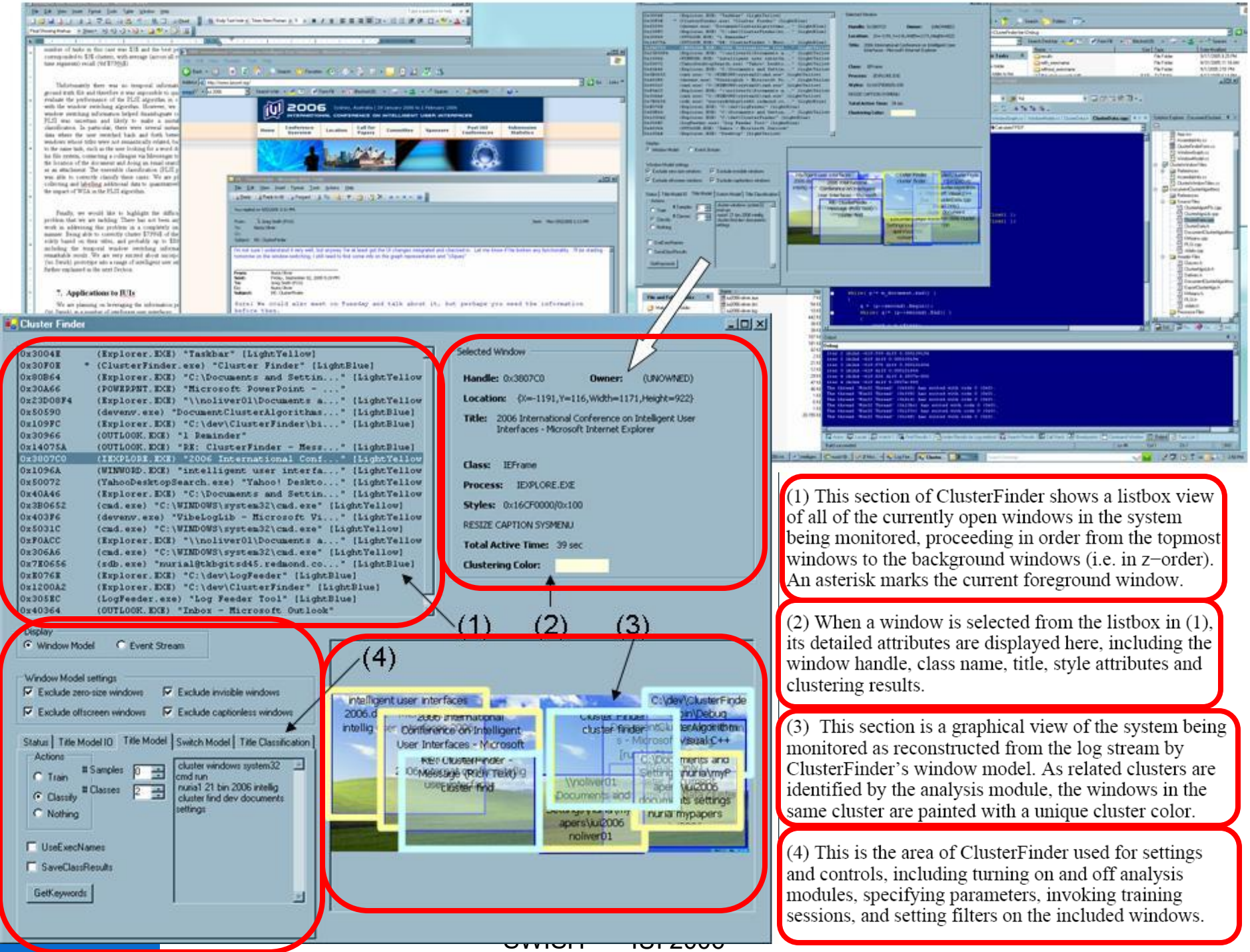
- SWISH keeps track of all window switching events that have taken place during a time interval of length T
- Builds a *window switching matrix* WS where each element *wsij* is proportional to the number of times the user switched from window *wi* to window *wj* during T
- T = 5 minutes in the experiments

Clustering based on History of Window Switches



Ensemble Classification





(1) This section of ClusterFinder shows a listbox view of all of the currently open windows in the system being monitored, proceeding in order from the topmost windows to the background windows (i.e. in z−order). An asterisk marks the current foreground window.

(2) When a window is selected from the listbox in (1), its detailed attributes are displayed here, including the window handle, class name, title, style attributes and clustering results.

(3) This section is a graphical view of the system being monitored as reconstructed from the log stream by ClusterFinder's window model. As related clusters are identified by the analysis module, the windows in the same cluster are painted with a unique cluster color.

(4) This is the area of ClusterFinder used for settings and controls, including turning on and off analysis modules, specifying parameters, invoking training sessions, and setting filters on the included windows.

Experiments

- Unsupervised system → hard to evaluate
- Need for ‘ground truth’ → manually label real data
- Cluster identification carried out by looking at the % of documents in a cluster that belong to a class. The largest % is assumed to be the label of the class

Evaluation Measures

- *Precision:* What % of documents in a cluster belong to the class assigned to that cluster
- *Recall:* What % of documents that belong to that class appear in this cluster
- *F1 Measure:* Balances both precision and recall:

$$F1 = \frac{2 * precision * recall}{precision + recall}$$

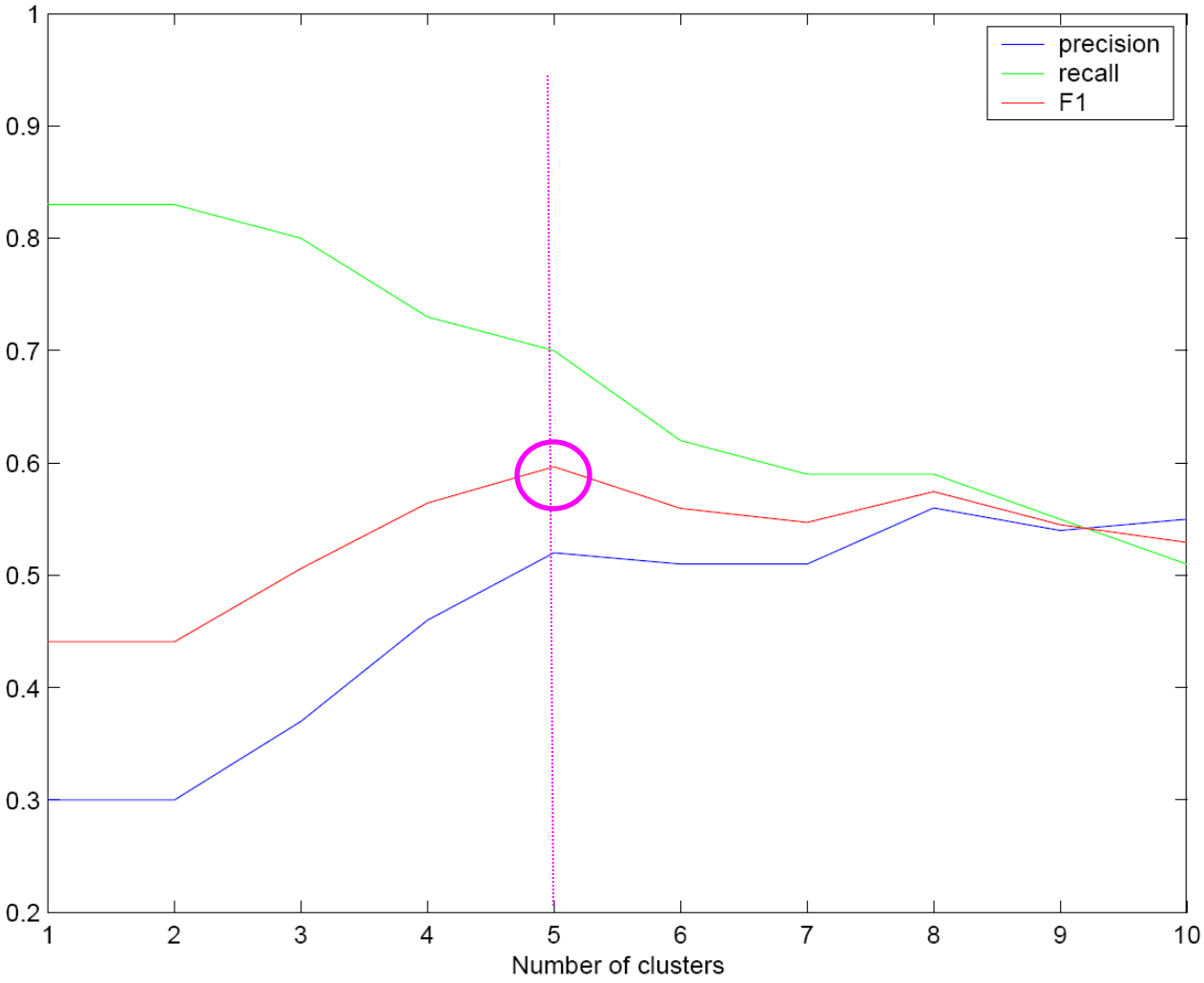
- The higher F1 the better the performance of the algorithm

Results

- Collected and manually labelled 4 hours of user data for a single user
- We identified 5 different tasks that the user worked on during that time period
- About 20-30% of the windows were spurious windows that didn't belong to any of the main tasks
- Recall is therefore more meaningful than precision in this context

Automatically Identifying the Number of Tasks

- Average precision, recall and F1 measure on PLSI over 30 runs with an increasing number of clusters



Impact of Title Preprocessing

- Question: Is it worth doing all the word preprocessing?
- Compared *Precision* and *Recall* with and without preprocessing the window titles

	With	Without
Precision	**0.49**	0.39
Recall	**0.72**	0.65

- Highly significant positive impact of preprocessing
- Results with 5 clusters and averages over 30 runs

Clustering Smaller Sets

- Divided data into 1-hour chunks
- Clustered those smaller sets
- Average number of tasks was 3
- Best performance corresponded to 3 clusters with average recall of 76%

So…what's in a cluster?

- 30 minutes of data
- Optimal number of clusters = 3
- Note how titles belong to very different applications:
  - Microsoft Word
  - Microsoft PowerPoint
  - Internet Explorer
  - Microsoft Outlook
  - Internet Messenger

A few exemplary titles

Top 3 keywords

Cluster 1

amazon harry potter halfblood prince book explore similar items
buy harry potter halfblood prince book rowling isbn 0439784549
all harry potter book products stores online bizrate
amazon harry potter halfblood prince book explore similar items
ebay harry potter
ebay item 8315147980 ends jun3005 194054 pdt harry potter halfblood prince book

Harry Potter Book

Cluster 2

final review donna malayeri
donna malayeri conversation

Donna Review Malayeri

Cluster 3

expedia travel discount airfare
browser check
flights
expedia is searching flights please wait
expedia is verifying price this trip
expedia sign
saved itineraries favorite destinations

Expedia Flight Trip

Applications to IUI

- **GroupBar**
  - Automatic or semi-automatic grouping
  - Automatic keyword extraction → labels for the groups
- **Implicit Query:**
  - Display relevant information to the current window
- **Integration with a task management prototype**
- **Automatic Window Clean-up Application**
  - Users open dozens of Explorer windows, and are too lazy to kill them
  - Collapse unused, unrelated windows to a single cluster
  - Provide an option after a timeout, to kill all together

Summary and Future Work

- With increased multitasking, support for task management is becoming an important task in supporting the rest of the information worker's work
- SWISH is a prototype that automatically detects groups of windows that are related to each other:
  - Semantically, from their window titles
  - Temporally, from the user's access patterns
- Future work includes
  - Incorporating SWISH in a number of IUIs
  - Exploring new strategies for 'ensemble classification'
  - Combining long-term with short-term user models
  - Extending our features to improve task assignment results

Thank you!!
Questions??