Darmstadt University of Technology

Telecooperation

Prediction Algorithms for User Actions

Melanie Hartmann, Daniel Schreiber

Telecooperation

- Motivation
- Sequence Prediction Algorithms (SPAs)
- Evaluation of SPAs
- Experiment
- Outlook and Summary

Telecooperation

- **Motivation**
- Sequence Prediction Algorithms (SPAs)
- Evaluation of SPAs
- Experiment
- Outlook and Summary

AUGUR

Build Intelligent User Interface to reduce users' cognitive load by guiding and teaching them while they perform a task

Focusing on form-based web applications

Features of AUGUR

Support mechanisms

provide online help that adapts to the user and his current working context

Interface Adaptation

adapt the interface to the user's needs and preferences, available devices

Content-Prediction

based on previous interaction, learned patterns and context information (direct and indirect usage)

Automation

allow to automate repetitive tasks by recognizing usage patterns

Context Knowledge

User Model

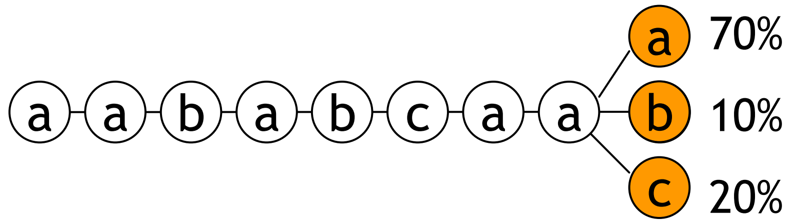
Interface Adaptation

Ranges from process guidance by highlighting to simplifying interface (e.g. to display it on smaller screens)

→ Need to know next action or next actions

→ We need a Sequence Prediction Algorithm (SPA)

Sequence Prediction



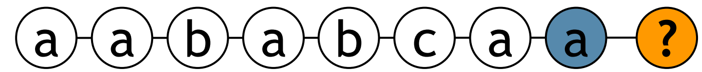
Definition:

- Given: first i symbols $a_1 \ldots a_i$ $(i \leq n)$ of sequence
- SPA returns probability $p(x \mid a_1 \ldots a_i)$ for each $x \in \Sigma$

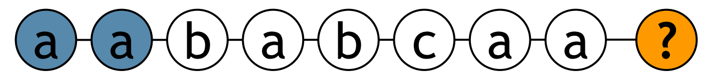
- Most algorithms consider only last k elements of the input sequence

Existing Algorithms

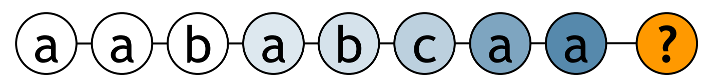
- IPAM:
  - First order Markov model (`k=1`)
  - Uses ageing



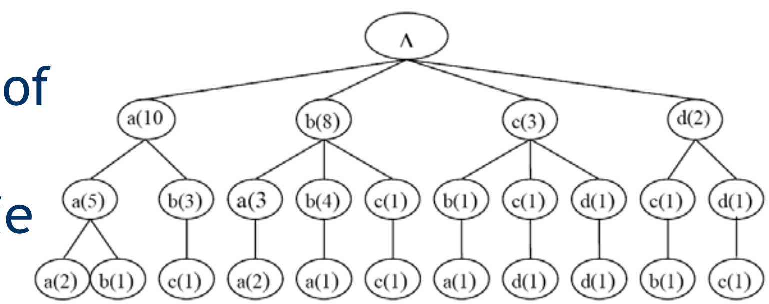
- ONISI:
  - k-nearest neighbor approach



- Jacobs and Blockeel (JB):
  - Mixed order Markov models
  - Builds upon IPAM (uses ageing)

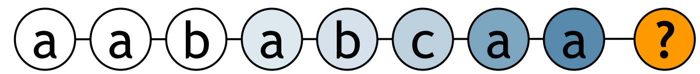


- ActiveLeZi (ALZ):
  - Mixed order Markov model
  - Datastructure build using extension of compression algorithm LZ78
  - Not every sequence stored in the trie
  - Trie grows with sequence length



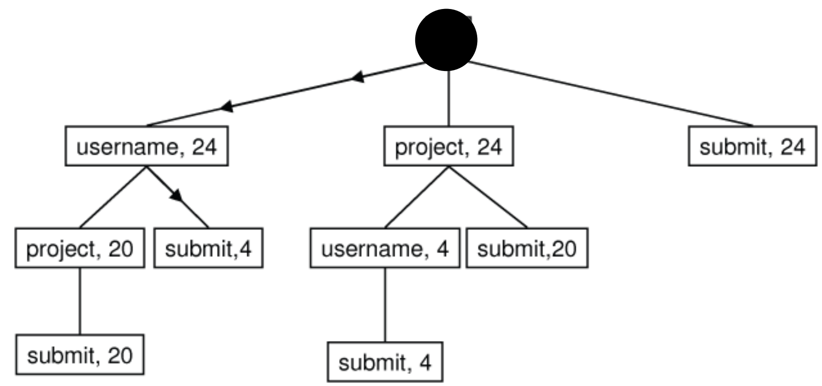
Our Approach - FxL

- Considers different order Markov models
- Storing n-grams up to length `k`



- Calculates a score for each action `x` (needs to be normalized to get the probability)

$$score(x) = \sum_{j=1}^{k-1} w(j) fr(a_{i+1-j} \ldots a_i \circ x)$$



- For FxL (frequency times length) we use `w(j)=j`
- For example: given following trie and recent history `username, project` results in following scores:

  score(username) = w(1)*4+ w(2)*0 = 4

  score(submit) = w(1)*20 + w(2)*20 = 60

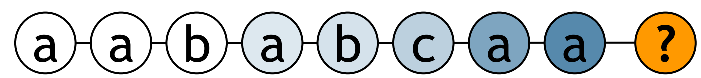
  score(project) = w(1)*0+w(2)*0 = 0

Our Approach - Adaptive FxL

- Adapts to the specific features of a dataset
- Considers predictive quality of the different order models
- Predictive quality q_i of ith order model is given by:

$$\frac{\text{how often model made a correct prediction}}{\text{how often this model was able to make a prediction}}$$

- $w(j) = j \; q_j \; f_j$
  f_j: Probability that all
  higher order models
  make a wrong prediction



- Example:
  3rd order model: q_3=0.7 → w(3) = 3*0.7*1
  2nd order model: q_2=0.6→ w(2) = 2*0.6*0.3
  1st order model: q_1=0.4 → w(1) = 1*0.4*(0.4*0.3)

Telecooperation

Requirements of AUGUR

- Requirements for an Ideal Online Learning Algorithm [Davison, Hirsh 1998]:
  - has high predictive accuracy
  - operates incrementally
  - does not need to retain a copy of the user’s full history of actions
  - outputs a list of predictions, sorted by confidence
  - is fast enough for interactive use
  - learns by passively watching the user
  - applies even in the absence of domain knowledge
  - [...]
- High applicability
- Work with small amount of data

Evaluating Performance of SPAs

Metrics for comparing SPAs

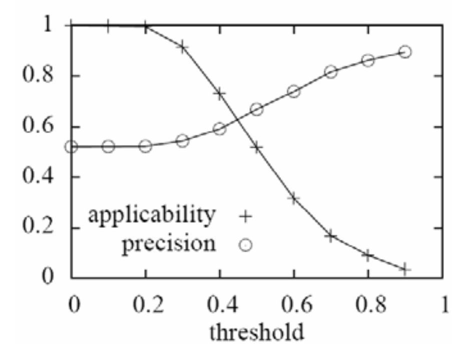
- **Prediction accuracy** pr_{ac}: reflects how
suggested on average
- **Prediction probability** pr_p: probability w
was predicted (averaged over all predictio
- **Applicability** ap: ratio how often algorithm was able to make a prediction

Consistent results

we are more interested in most probable next action than its probability

Parameters of the input sequence that influence the result

1. Available dataset size
2. Distribution of repetitive sequences (if it occurs at least m times in the dataset)
3. Noise in the repetitive sequences
4. Applicability ap: ratio how often algorithm was able to make a prediction



Telecooperation

Used Datasets

- Word:
  - Usage data from MS word
  - Contains about 40 000 commands from 16 users
- Greenberg dataset
  - Contains over 225 000 UNIX commands from 168 users
  - Widely used in literature
- Crossdesktop (XD) dataset
  - Log data from a web application for managing files and emails
  - About 200 000 requests from 37 users

```
FileOpen
FileNew
FormatBold
FormatCenterPara
FormatRightPara
FormatCenterPara
FormatBold
FormatLeftPara
```

```
pix
umacs
pix
umacs
script
lpr
lpq
lquota
```

Cross-Desktop

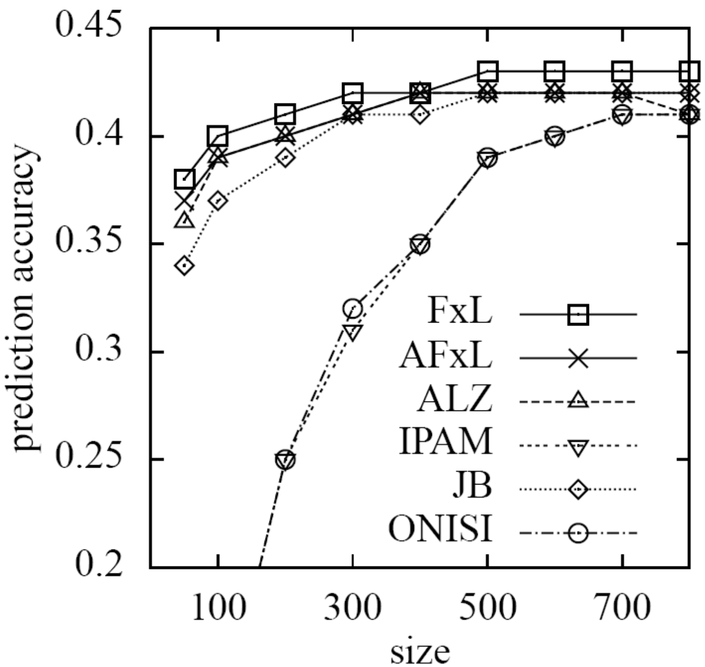
```
login.cgi
crossDesktop.cgi
checkMail.cgi
calendarReminder.cgi
status.cgi
calendarReminder.cgi
fileListing.cgi
fileListing.cgi
```

Experiment

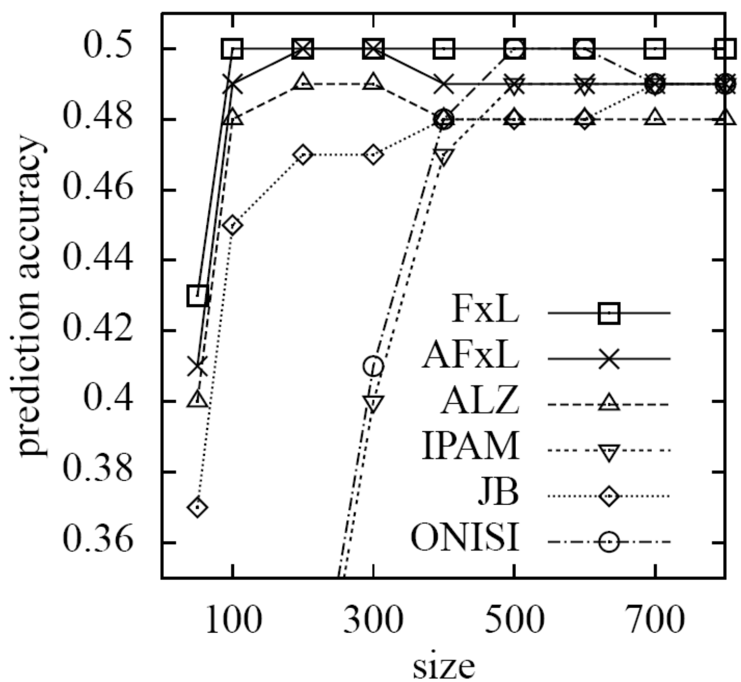
Used 20% of the data for training

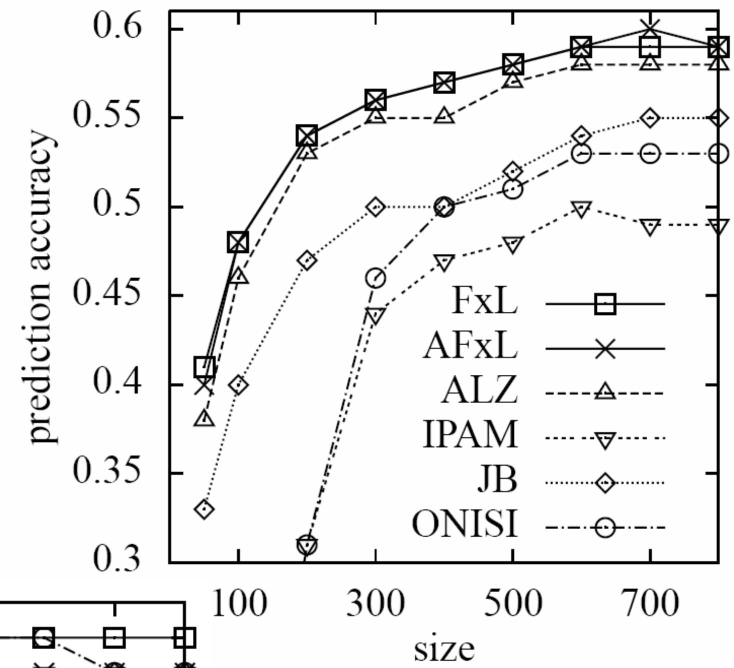
Parameter 1: size of dataset

Greenberg



Word

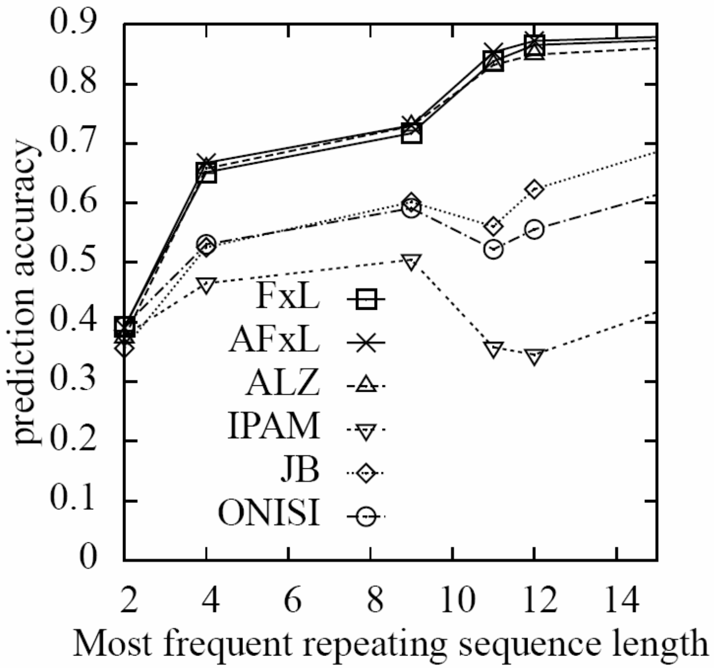




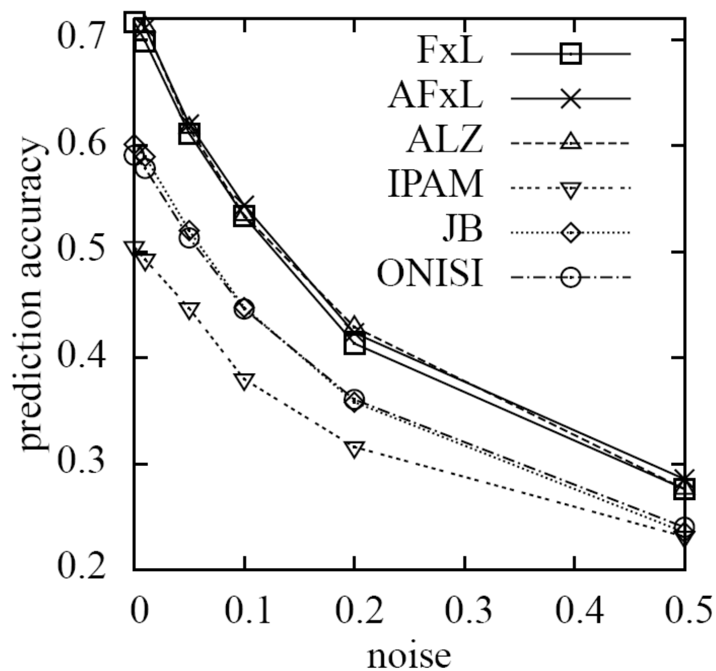
XD

Experiment

Parameter 2: Distribution of repetitive sequences



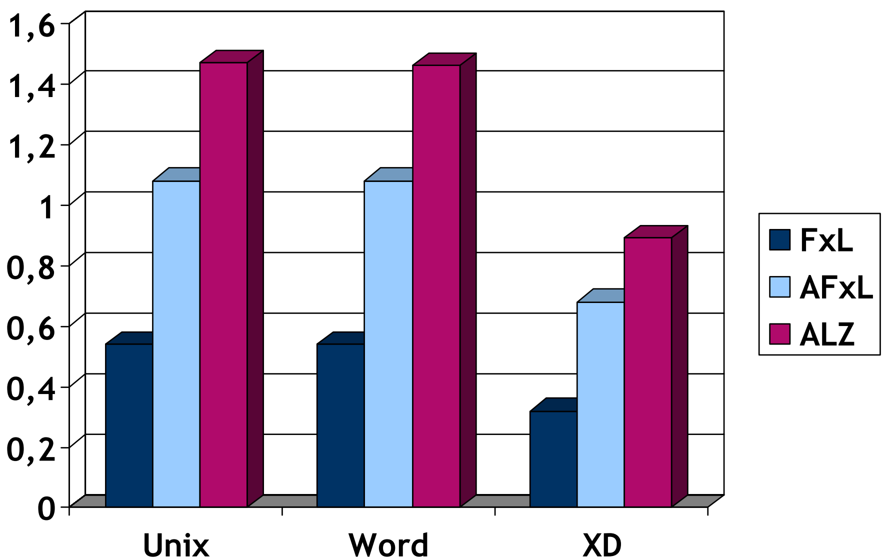
Parameter 3: Noise



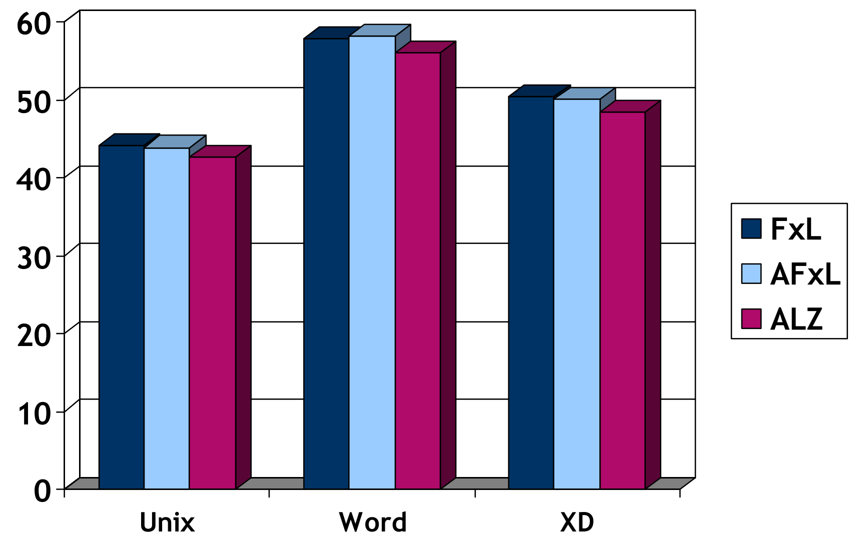
➔ FxL, AFxL and ActiveLeZi perform best

Experiment

- To run it on a mobile device, we need (besides optimal performance)
  - Low computation time
  - Low memory requirements

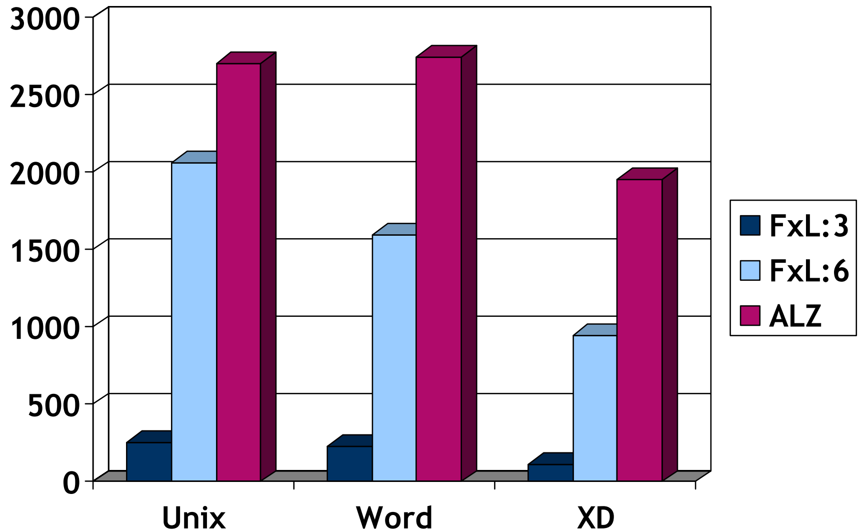


Average computation time [s] per sequence



Predictive accuracy

Experiment



Stored keys

- The required memory for (A)FxL is limited by the maximal trie depth k
- In the experiments a trie depth of k=4 was sufficient to reach the optimal performance

→ FxL optimal candidate for applying in AUGUR

Telecooperation

Motivation

Sequence Prediction Algorithms

Evaluation of SPAs

Experiment

Outlook and Summary

Summary and Outlook

- We introduced two new SPAs
- We showed method how to systematically analyze SPAs
- We showed that FxL is prime candidate for applying in AUGUR

- However, the performance of pure statistical approaches is limited ➔ how can task and context knowledge be used to improve the results?

Thanks for your attention