Flexible Querying of Lifelong Learner Metadata

Alex Poulovassilis, Peter T. Wood

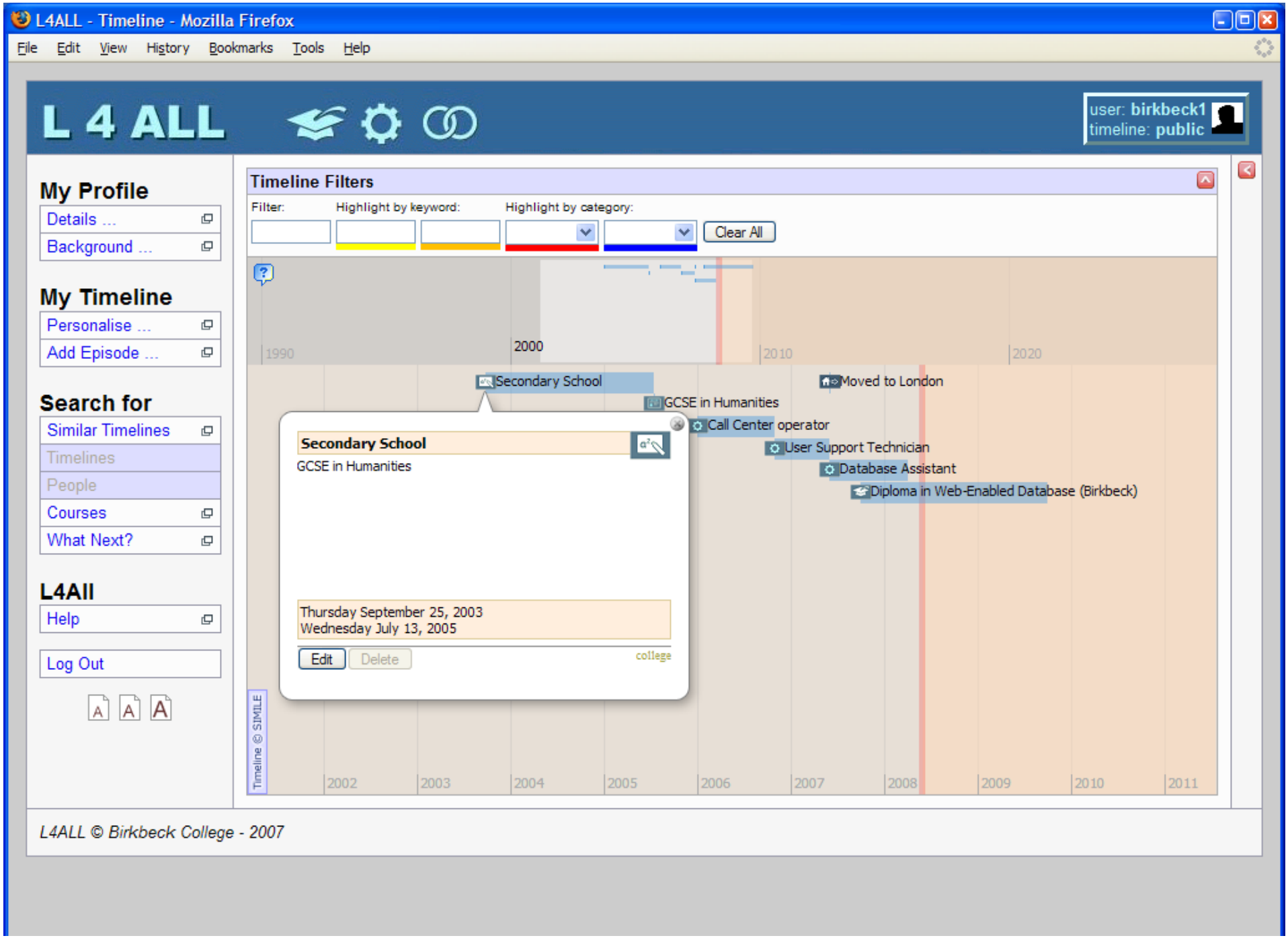
SemHE Workshop at ECTEL'09, October 2009

Outline of the talk

1. Motivation
2. Examples of ApproxRelax Queries
3. Single-Conjunct ApproxRelax Queries
4. General ApproxRelax Queries
5. Conclusions and Future Work

1. Motivation

- Face-to-face guidance and support for career and educational choices has been found to be patchy and uneven
- Transitions from one stage of education to the next are critical decision points in a learner's journey and better targeted information may make these transitions more successful
- There is also a link between careers guidance and learner retention, based on correcting false assumptions and creating learners' expectations in line with real-life experiences of others
- In this direction, the L4*All* system aims to support lifelong learners in exploring learning opportunities and in planning and reflecting on their learning
- It allows learners to create and maintain a chronological record of learning, work and personal episodes – their *timeline*


L4ALL - Timeline - Mozilla Firefox
File Edit View History Bookmarks Tools Help
L 4 ALL
user: birkbeck1
timeline: public
My Profile
Details ...
Background ...
My Timeline
Personalise ...
Add Episode ...
Search for
Similar Timelines
Timelines
People
Courses
What Next?
L4All
Help
Log Out
Timeline Filters
Filter:
Highlight by keyword:
Highlight by category:
Clear All
1990
2000
2010
2020
Secondary School
GCSE in Humanities
Moved to London
Call Center operator
User Support Technician
Database Assistant
Diploma in Web-Enabled Database (Birkbeck)
Secondary School
GCSE in Humanities
Thursday September 25, 2003
Wednesday July 13, 2005
Edit
Delete
college
Timeline © SIMILE
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
L4ALL © Birkbeck College - 2007

Motivation

- This approach is distinctive in that the timeline provides a record of lifelong learning, rather than at just one stage or period
- Learners can share their timelines with other learners, with the aim of fostering collaborative formulation of future learning goals and aspirations
- The focus is particularly on those post-16 learners who traditionally have not participated in higher education

Motivation

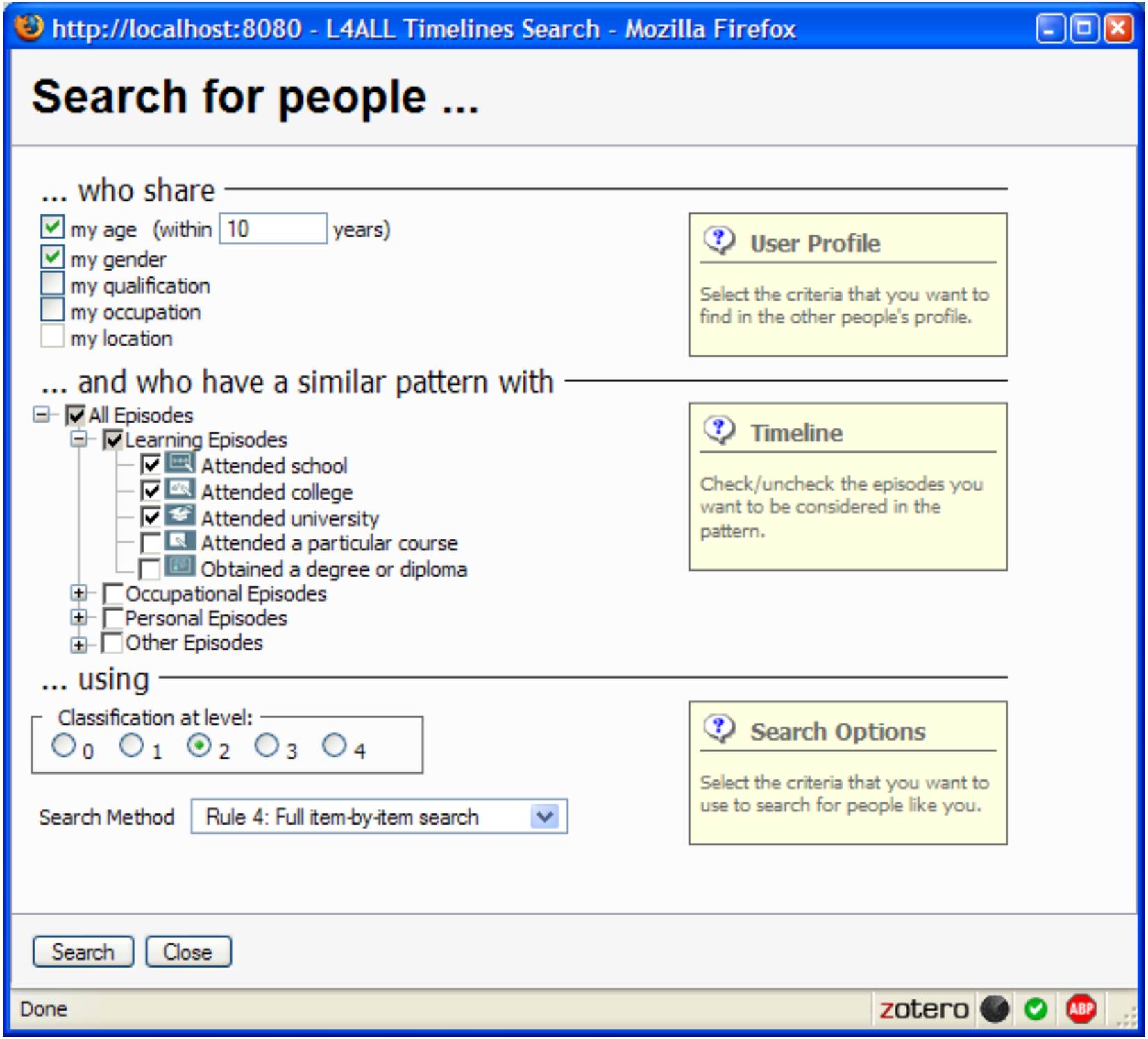
- The original L4*All* system was co-designed with FE/HE students, educators and other stakeholders in the London region
- A consultation process was undertaken at the project start to identify learners' educational goals and career objectives, with the aim of accommodating the needs of different user groups.
- Interviews were held with FE learners from Community College Hackney and City of Westminster College; and with HE mature learners at Birkbeck College, University of London

Motivation

- The L4*All* user interface provides screens for the user to enter their personal details, to create and maintain a timeline of their learning, work and personal episodes, and to search for course information and for the timelines of other users
- A key aim of the system is to allow learners to search over the timeline data, and to identify possible choices for their own future learning and professional development by seeing what others with a similar background have gone on to do
- In particular, the ITS 2008 and ECTEL 2009 (forthcoming) papers by Van Labeke, Magoulas & Poulovassilis describe a facility for searching for "people like me" which takes users through a three-step process in specifying timelines that might be relevant to them.

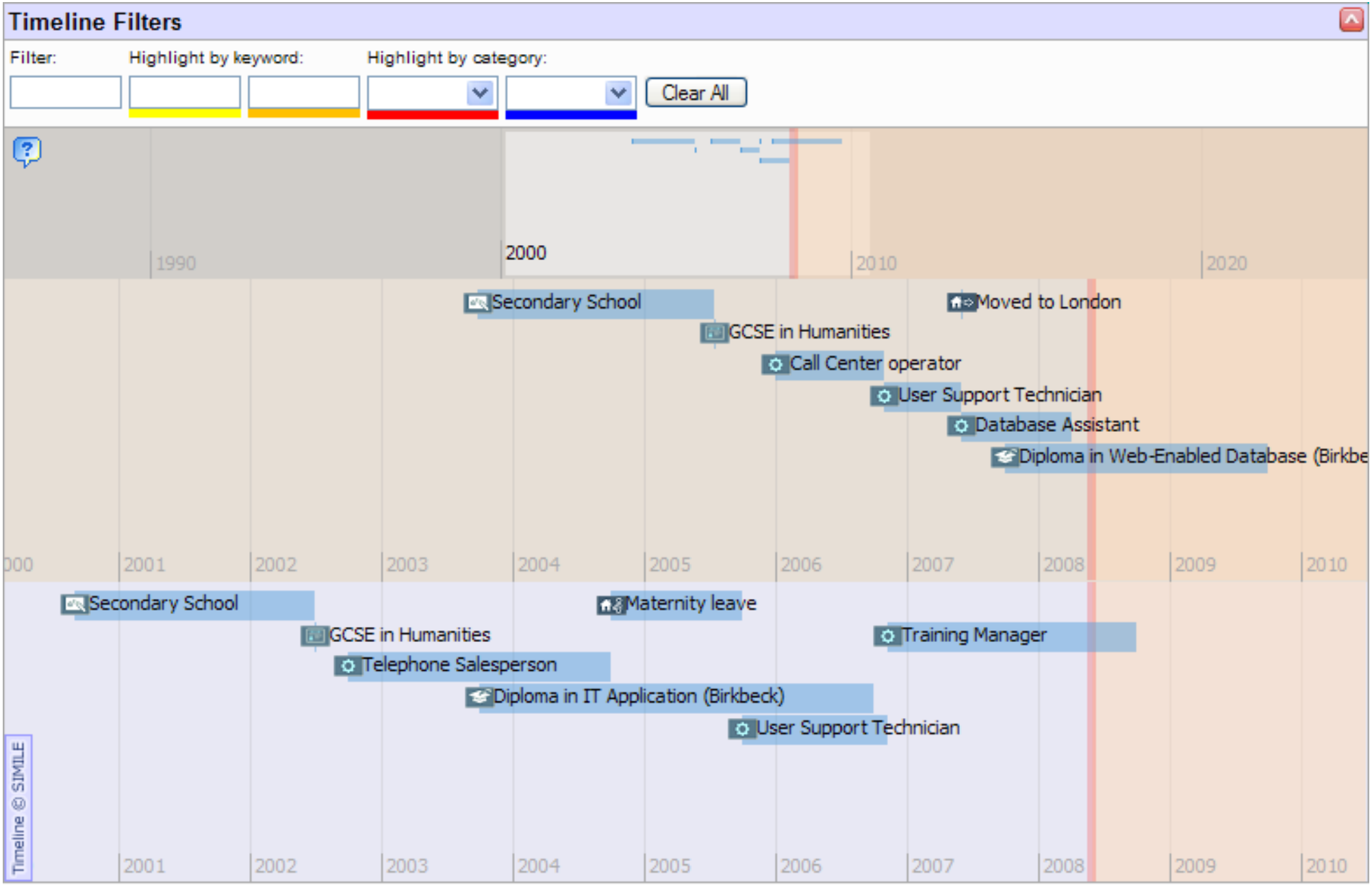
Motivation

- The ITS 2008 paper describes a method of encoding timelines as token-based strings, thereby allowing string similarity metrics to be applied to compare the user's own timeline with other users' timelines
- Four alternative metrics were deployed in the L4*All* system: Jaccard Similarity, Dice Similarity, Euclidean Distance, NeedlemanWunch Distance (all part of the SimMetrics Java package, see: www.dcs.shef.ac.uk/~sam/stringmetrics.html)
- A dedicated interface for the new search for "people like me" facility was designed taking users through the three-step process for specifying their definition of "people like me":


http://localhost:8080 - L4ALL Timelines Search - Mozilla Firefox
Search for people ...
... who share
my age (within 10 years)
my gender
my qualification
my occupation
my location
User Profile
Select the criteria that you want to find in the other people's profile.
... and who have a similar pattern with
All Episodes
Learning Episodes
Attended school
Attended college
Attended university
Attended a particular course
Obtained a degree or diploma
Occupational Episodes
Personal Episodes
Other Episodes
Timeline
Check/uncheck the episodes you want to be considered in the pattern.
... using
Classification at level:
0 1 2 3 4
Search Method
Rule 4: Full item-by-item search
Search Options
Select the criteria that you want to use to search for people like you.
Search
Close
Done
zotero

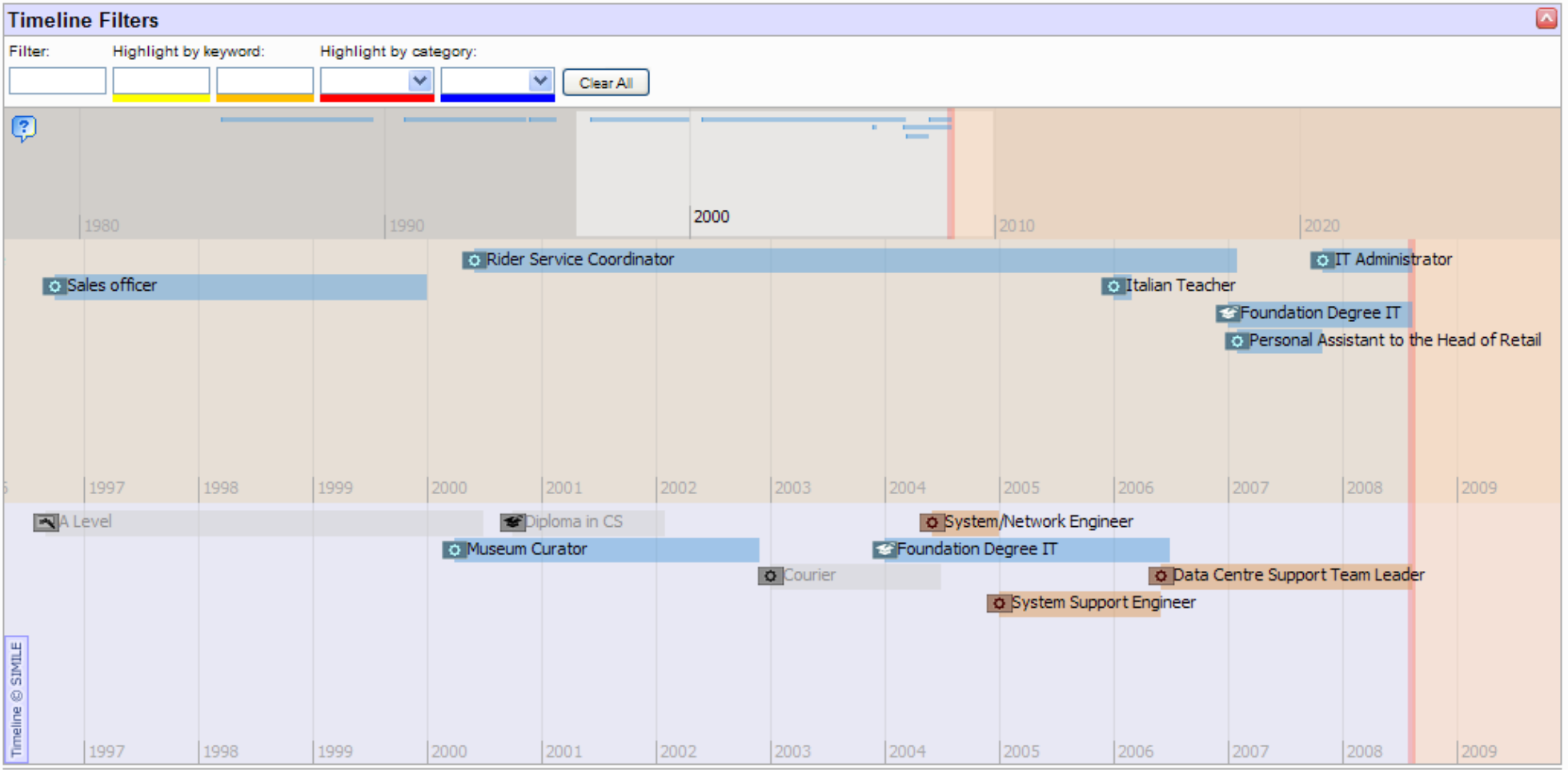
Motivation

- Once a definition of "people like me" has been specified, the system returns a list of all the candidate timelines, ranked by their normalised similarity
- The user can then select any of these timelines to visualise and explore
- The selected timeline is shown in the main page as an extra strip below the user's own timeline
- Episodes within the selected timeline that have been designated as being public by its owner are visible, and the user can select and explore these:


Timeline Filters
Filter:
Highlight by keyword:
Highlight by category:
Clear All
1990
2000
2010
2020
Secondary School
Moved to London
GCSE in Humanities
Call Center operator
User Support Technician
Database Assistant
Diploma in Web-Enabled Database (Birkbe
000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
Secondary School
Maternity leave
GCSE in Humanities
Training Manager
Telephone Salesperson
Diploma in IT Application (Birkbeck)
User Support Technician
Timeline © SIMILE
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010

Motivation

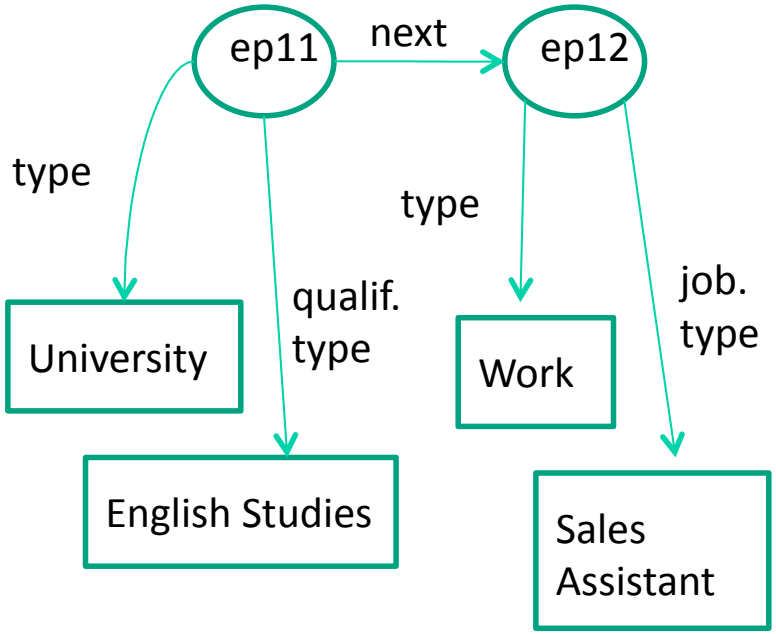
- The ECTEL'09 paper reports on the design and results of an evaluation of this search for "people like me" functionality undertaken with a group of mature learners
- In follow-on work, Van Labeke *et al* explored a more contextualised usage of timeline similarity matching, which explicitly indicates which episodes of the target timeline have no match within the user's own timeline
- Such episodes potentially represent episodes that the user may be inspired to explore or may even consider for their own future personal development:


Timeline Filters
Filter:
Highlight by keyword:
Highlight by category:
Clear All
1980
1990
2000
2010
2020
Rider Service Coordinator
IT Administrator
Sales officer
Italian Teacher
Foundation Degree IT
Personal Assistant to the Head of Retail
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
A Level
Diploma in CS
System/Network Engineer
Museum Curator
Foundation Degree IT
Courier
Data Centre Support Team Leader
System Support Engineer
Timeline © SIMILE

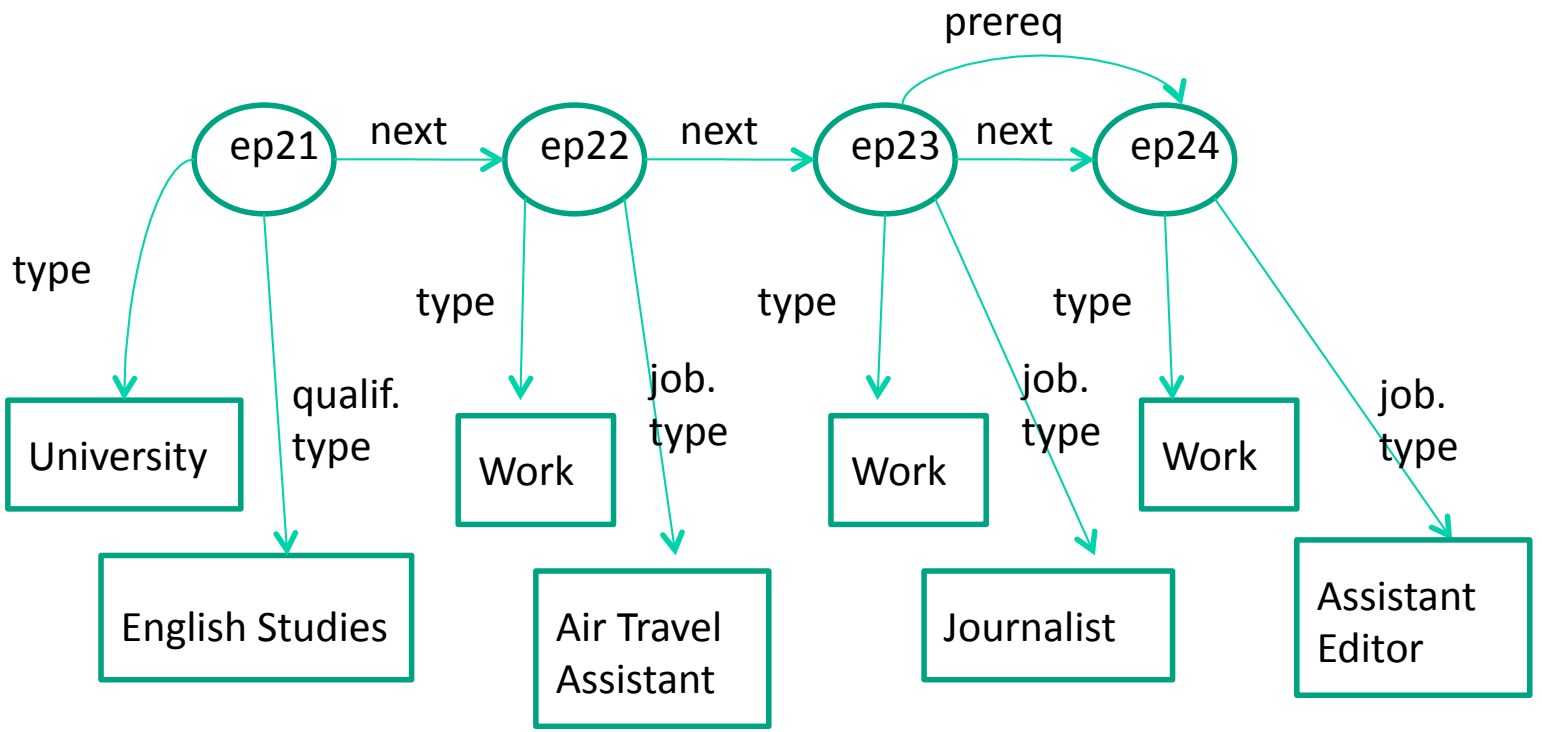
Motivation

- Although this facility certainly helps users to find and explore relevant timelines and episodes, it is rather rigid for a number of reasons:
  - it offers a fixed set of similarity metrics over the timeline data
  - it allows just a single level of detail to be applied to the classifications of the selected categories of episode
  - the similarity matching is applied to all these categories of episode in the user's timeline and the target timelines
- Thus, there is limited flexibility for users to formulate their requirements for the timeline search, and to explore alternative formulations of selected parts of their query
- This paper explores flexible querying techniques for supporting users' search over timelines

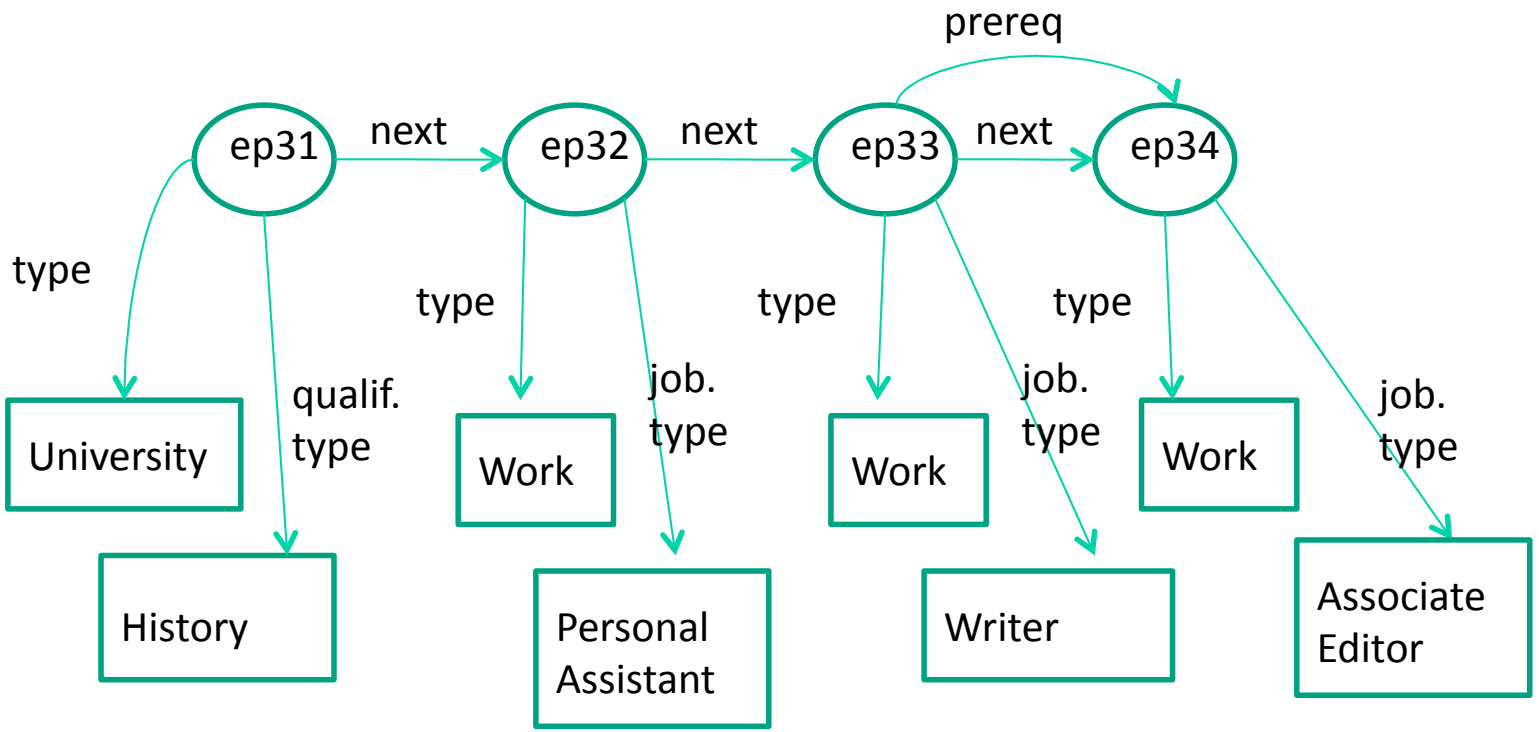
2. Examples of ApproxRelax Queries



A fragment of **my timeline** data and metadata

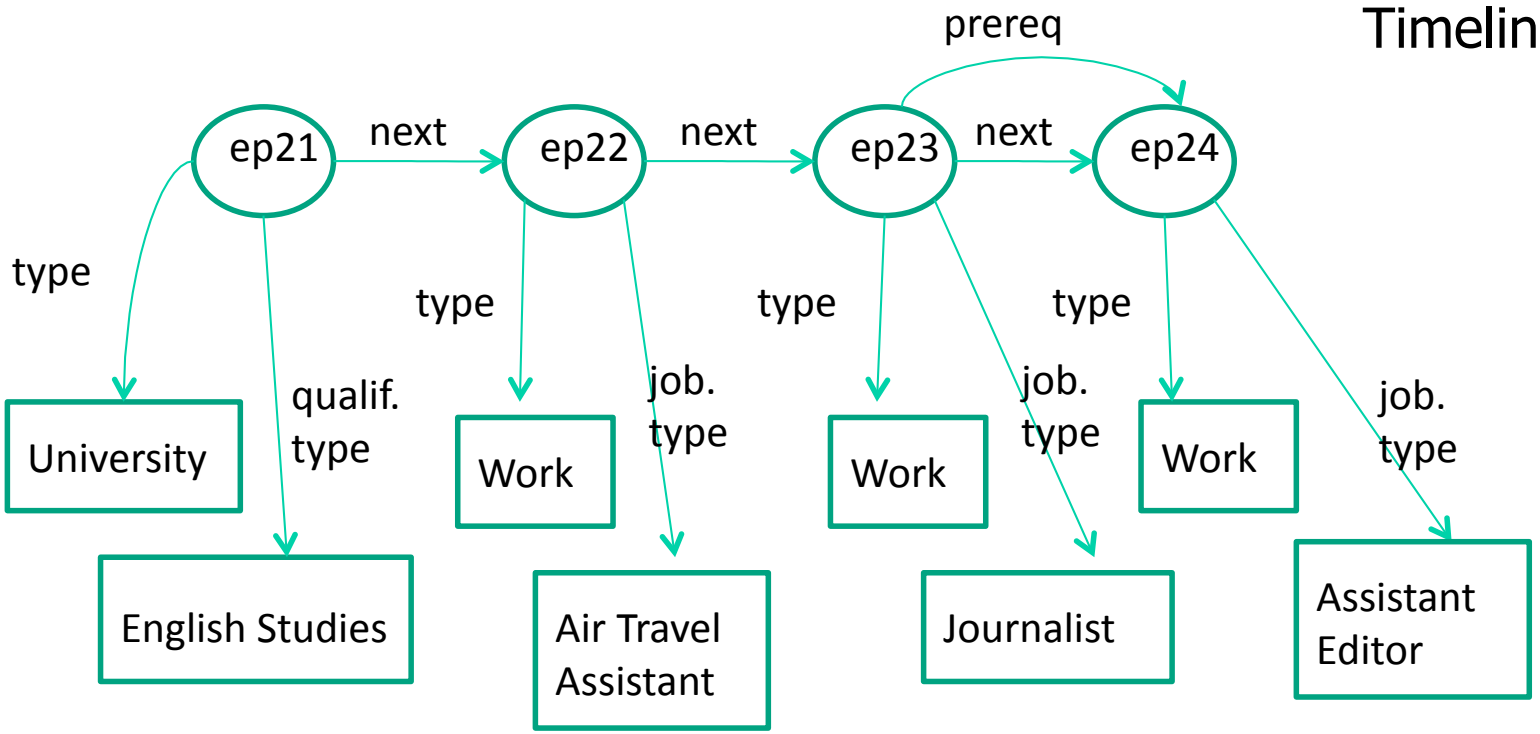


Another user's timeline, User 2, where "prereq" indicates that this user believes that undertaking an earlier episode was necessary in order for them to be able to proceed to/achieve a later episode.



Another user's timeline, User 3

Timeline of User 2

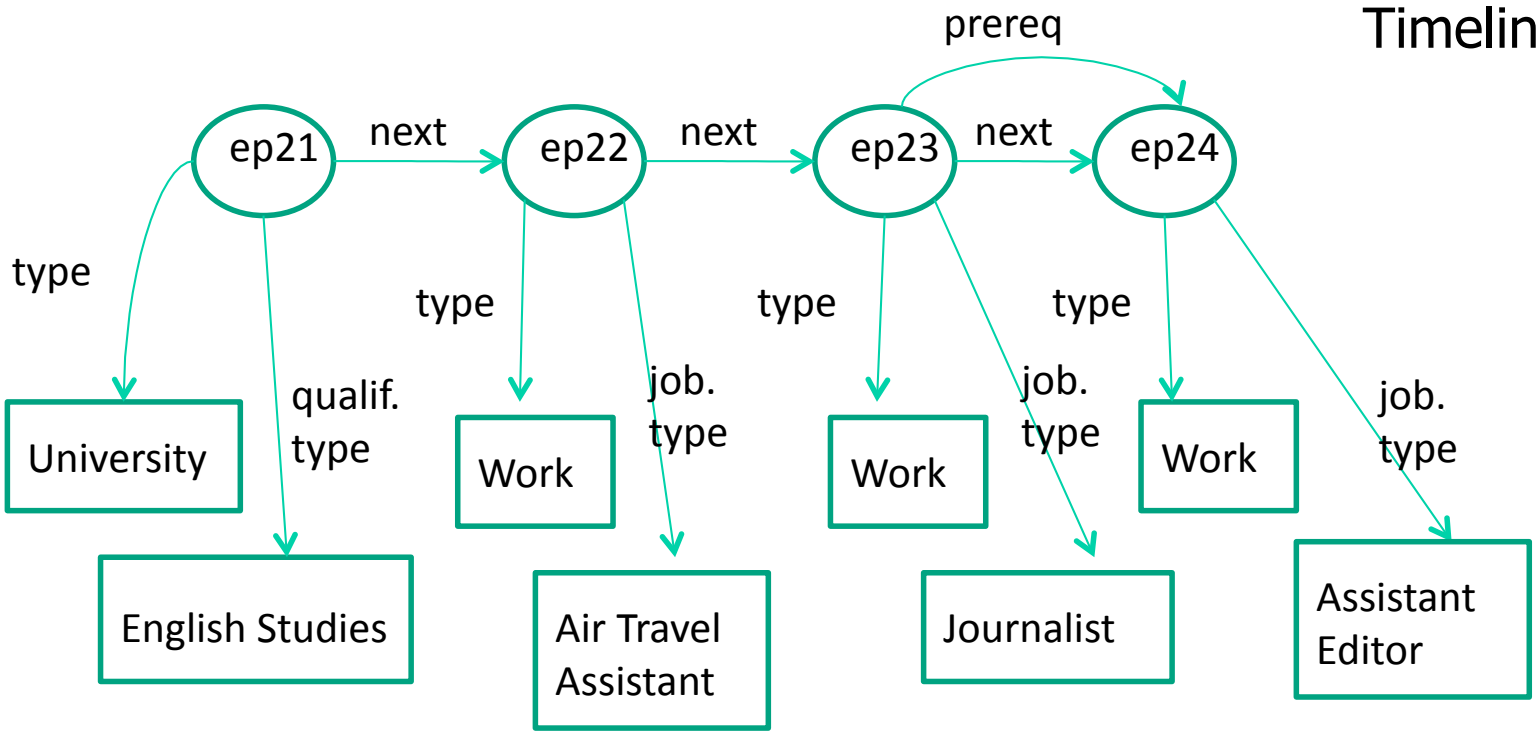


Query type 1: I've done this so far; what work positions can I reach and how? E.g. selecting just the relevant prefix of my timeline (my English degree, rather than my temporary work as a Sales Assistant):

(?E2,?P) ← (?E1,type,University),(?E1,qualif.type,EnglishStudies),
(?E1,prereq+,?E2),
(?E2,type,Work),(?E2,job.type,?P)

However, this will return no results relating to User 2

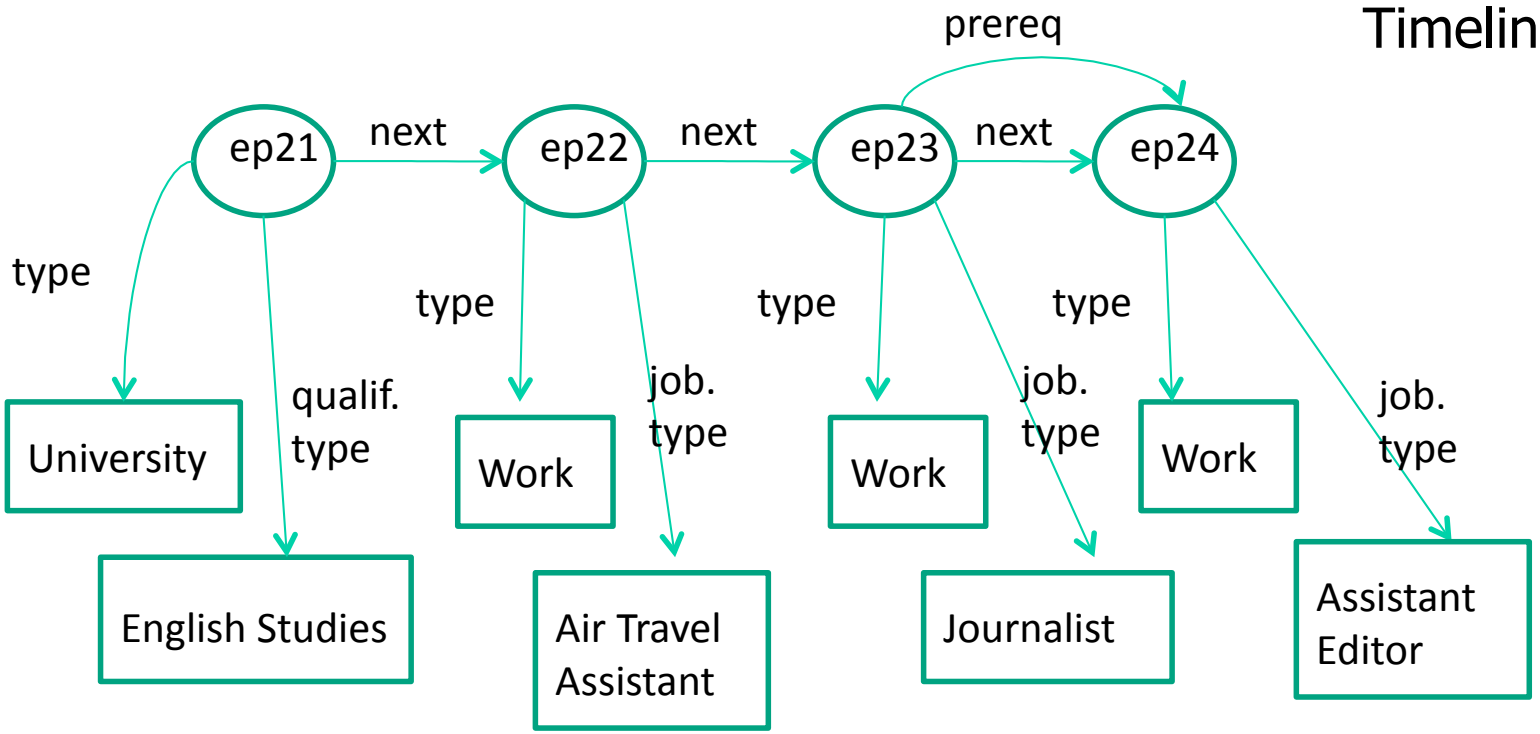
Timeline of User 2



Allowing query approximation can yield some answers. In particular, allowing replacement of the edge label "prereq" by the label "next", at an edit cost of 1, we can submit this query:

(?E2,?P) ← (?E1,type,University),(?E1,qualif.type,EnglishStudies),
APPROX (?E1,prereq+,?E2),
(?E2,type,Work),(?E2,job.type,?P)

Timeline of User 2

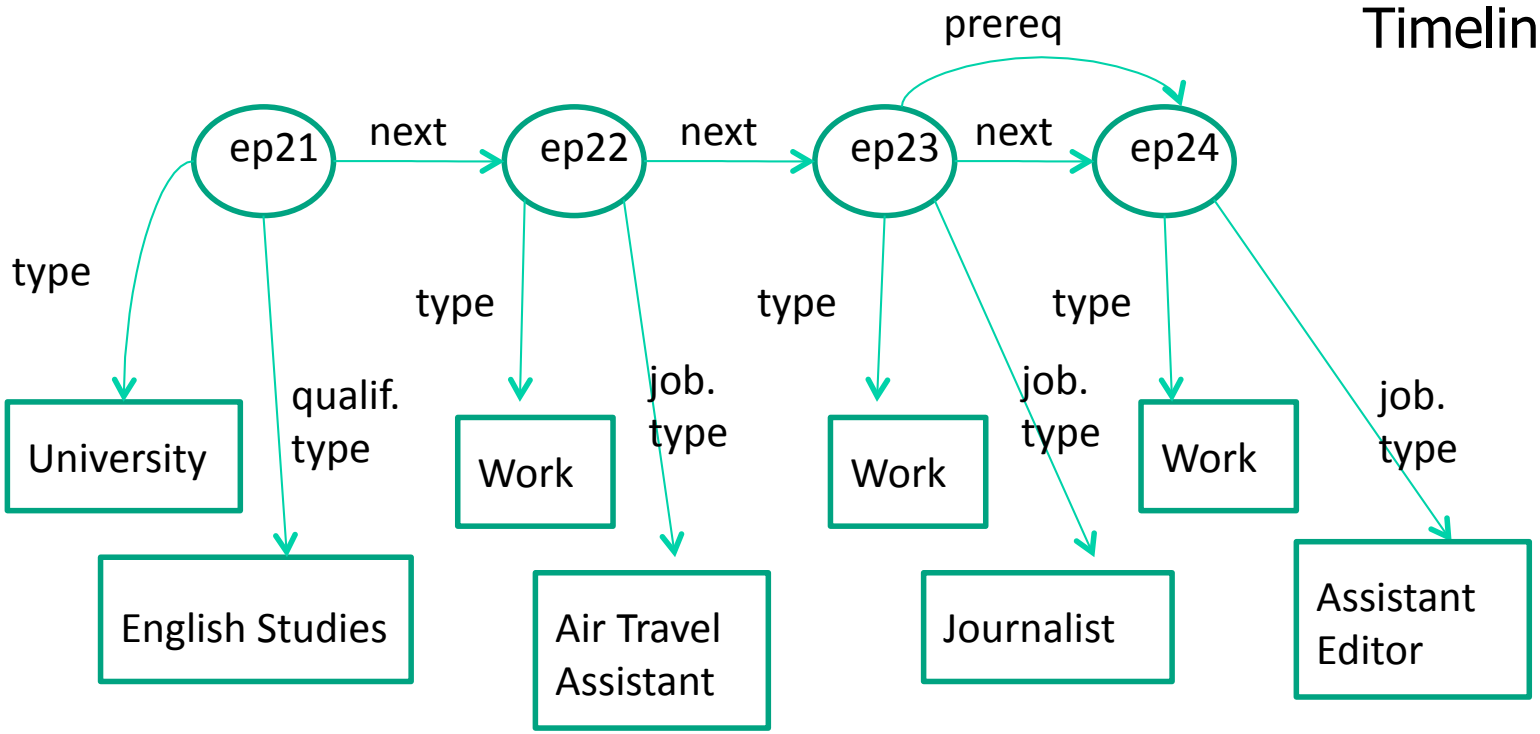


The regular expression prereq+ can be approximated by next.prereq* at edit distance 1. This allows the system to return

(ep22,AirTravelAssistant)

We may judge this result to be not relevant and seek further results from the system

Timeline of User 2

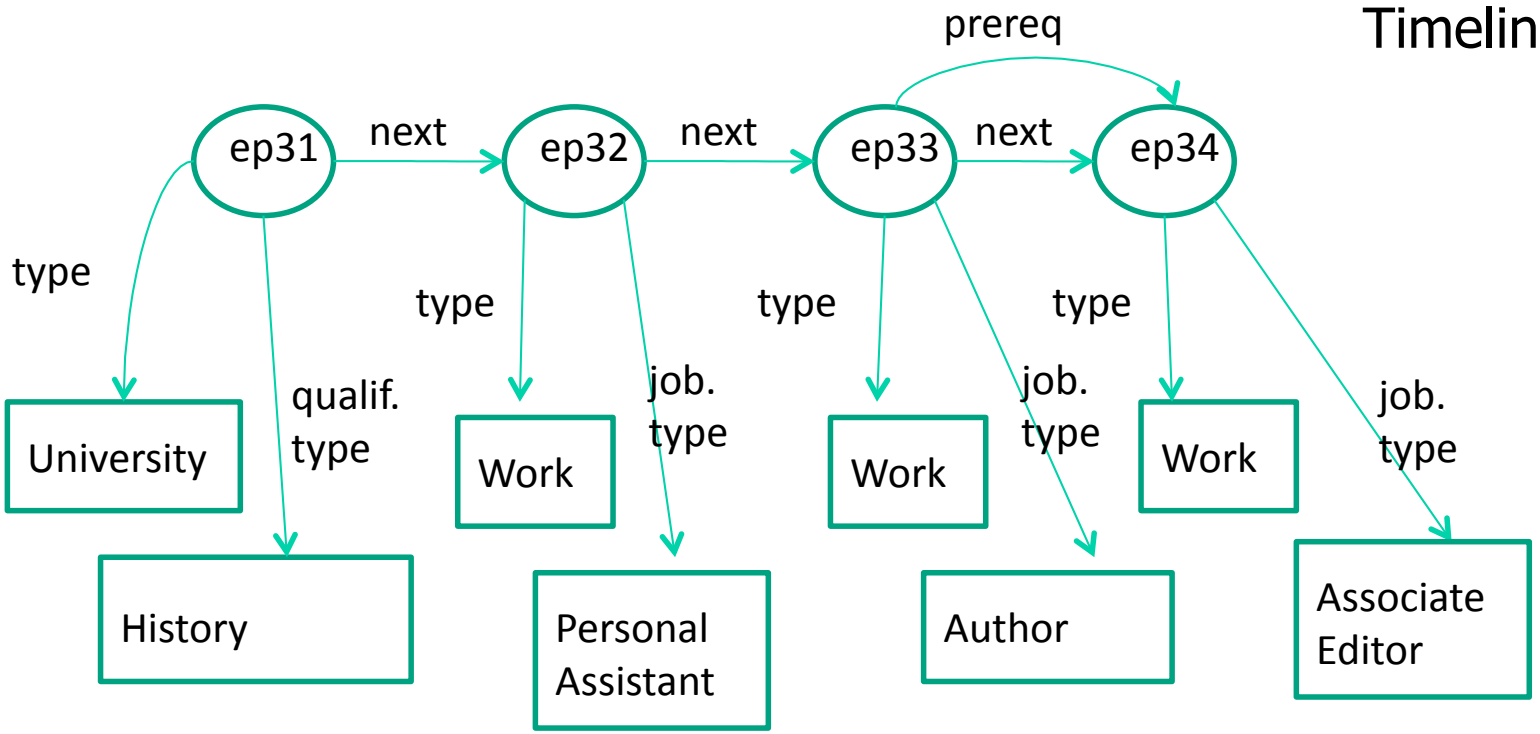


next.prereq* can be approximated by next.next.prereq*, now at edit distance 2. This allows the following answers to be returned:

(ep23,Journalist), (ep24,AssistantEditor)

We may judge both of these as being relevant, and can then request the system to return the whole of User 2's timeline to explore further.

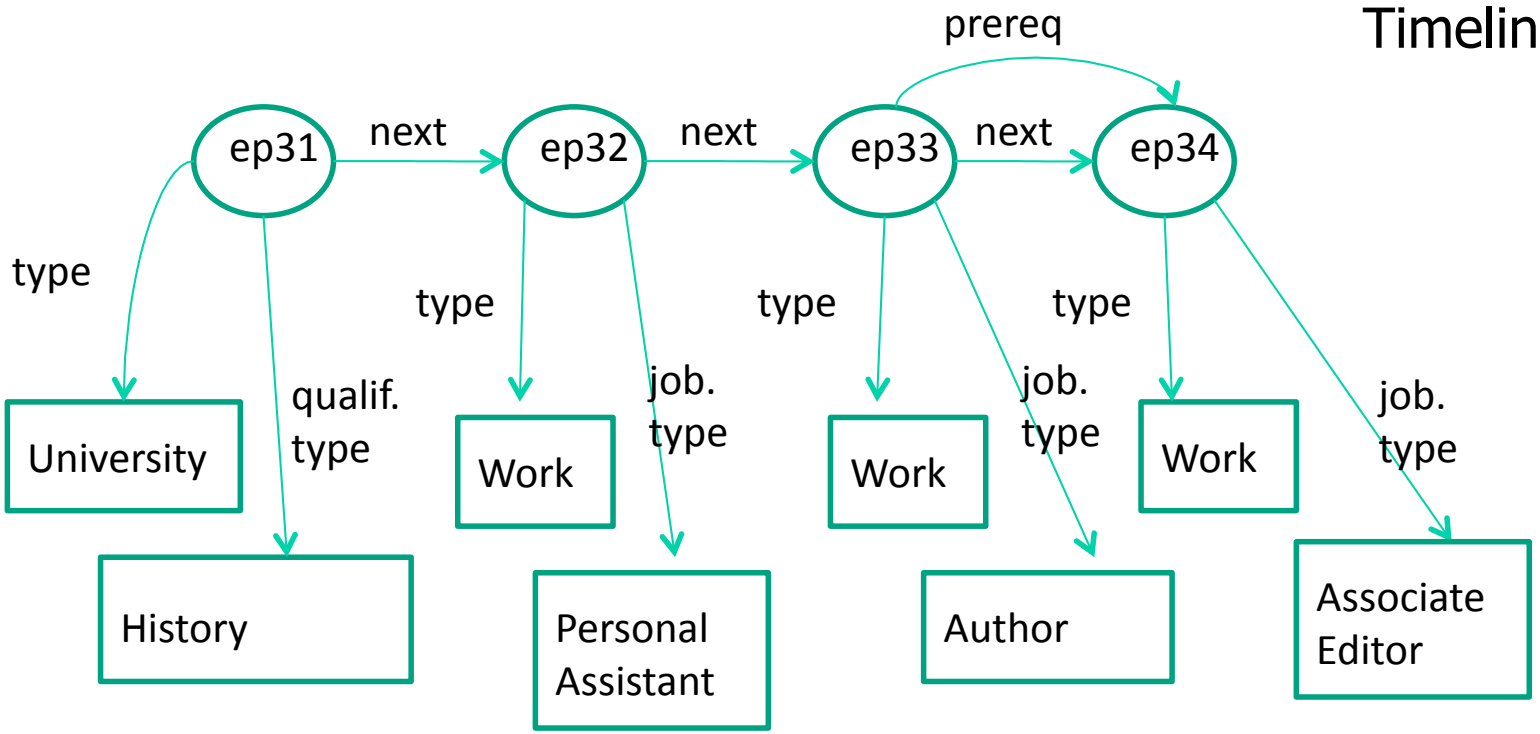
Timeline of User 3



Query type 1, relaxed: What jobs are open to me if I study English, or something similar, at University?

(?E2,?P) ← (?E1,type,University),(?E1,qualif, ?D),
RELAX (?D,type,EnglishStudies),
APPROX (?E1,prereq+,?E2),
(?E2,type,Work),(?E2,job.type,?P)

In addition to the answers obtained by the previous query, we also have answers from the timeline of User 3:

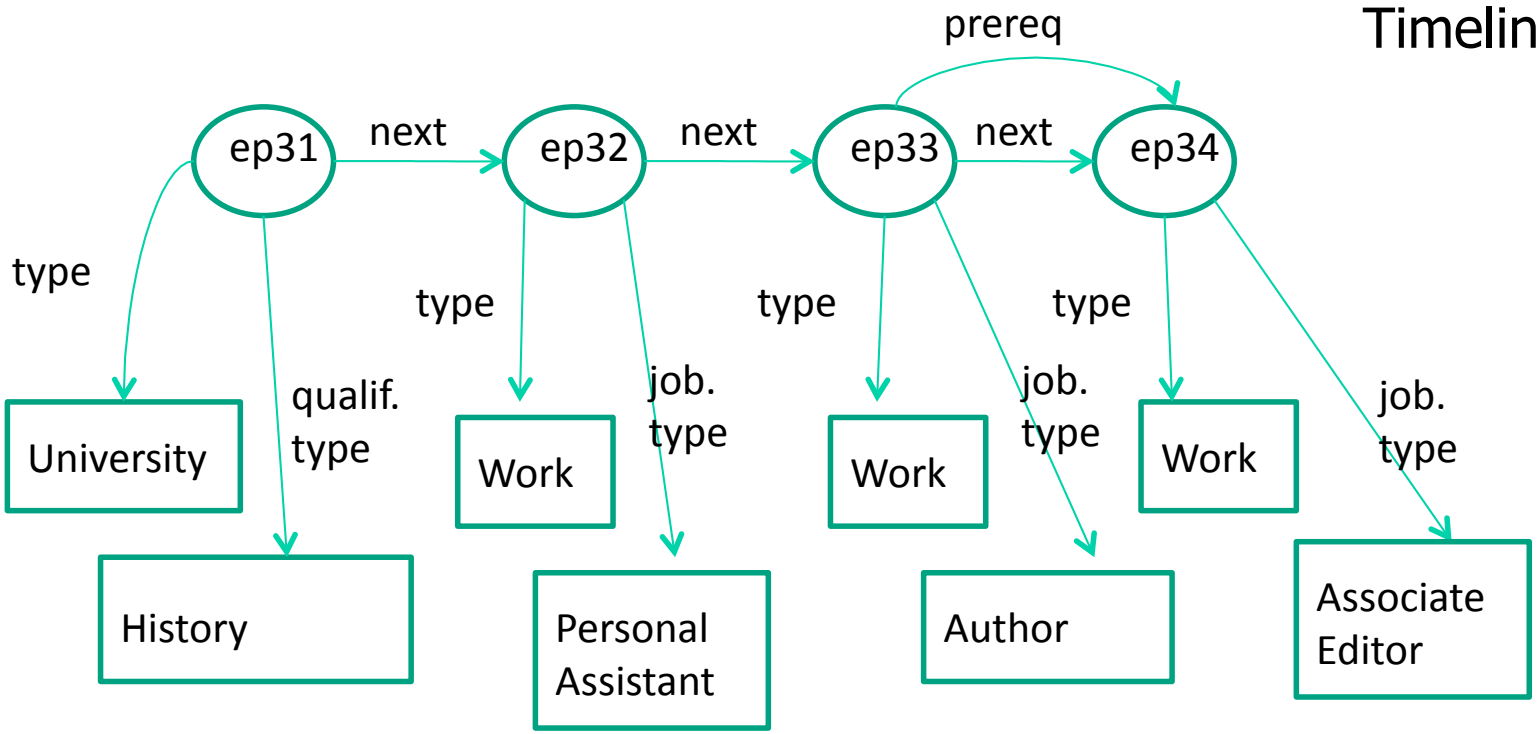


The regular expression prereq+ can be approximated by next.prereq* at edit distance 1, and English Studies can be relaxed to Humanities at relaxation distance 2 (see Fig. 1 of the paper), thus encompassing History. This allows the system to return

(ep32,PersonalAssistant)

at overall distance 3 from the original query, assuming the same weighting is applied to approximation and relaxation.

Timeline of User 3



next.prereq* can be approximated by next.next.prereq*, now at edit distance 2, with English Studies again relaxed to Humanities at relaxation distance 2 . This allows the following answers to be returned:

(ep33,Author), (ep34,AssociateEditor)

at overall distance 4 from the original query. We may judge both of these as being relevant, and can then request the system to return the whole of User 3's timeline to explore further.

Query type 2: I've done this so far, I want to attain to a certain work position, how might I do it? E.g. I've studied English at University, how might I become an Assistant Editor?

(?E2,?P) ← (?E1,type,University),(?E1,qualif.type,EnglishStudies),
APPROX (?E1,prereq+,?E2),
(?E2,job.type,?P)
APPROX (?E2,prereq+,?Goal),
(?Goal,type,Work),(?Goal,job.type,AssistantEditor)

At distance 0 there are no results from the timeline of User 1.
At distance 1 there are still no results.
At distance 2, the following answers are returned:

(ep22,AirTravelAssistant), (ep23,Journalist)

the second of which gives us potentially useful information on how to become an Assistant Editor having studied English at University

Query type 2, relaxed: I've done this so far, I want to attain to a certain work position or something like it, how might I do it? E.g. I've studied English at University, how might I become an Assistant Editor or something similar?

(?E2,?P) ← (?E1,type,University),(?E1,qualif.type,EnglishStudies),
APPROX (?E1,prereq+,?E2),
(?E2,job.type,?P)
APPROX (?E2,prereq+,?Goal),
(?Goal,type,Work),(?Goal,job,?AG)
RELAX (?AG,type,AssistantEditor)

Query type 2, further relaxation: What jobs similar to Assistant Editor may be open to someone who's studied English or something similar, and how could they attain such jobs?

(?E2,?P) ← (?E1,type,University),(?E1,qualif,?D)
RELAX (?D,type,EnglishStudies),
APPROX (?E1,prereq+,?E2),
(?E2,job.type,?P)
APPROX (?E2,prereq+,?Goal),
(?Goal,type,Work),(?Goal,job,?AG)
RELAX (?AG,type,AssistantEditor)

In addition to the earlier results from User 2, English Studies can be relaxed to Humanities at distance 2 (thus encompassing History) and Assistant Editor to Editor at distance 1 (thus encompassing Associate Editor) – see Fig. 1 of the paper – thus returning the following answers from User 3's timeline at distance 5 (edit distance 2 + relaxation distance 3):

(ep32,PersonalAssistant), (ep33,Author)

3. Single-Conjunct ApproxRelax Queries

- A *single-conjunct query* consists of an expression of the form:

  $Z_1, Z_2 \leftarrow (X, R, Y)$

  where X and Y are constants or variables, R is a regular expression over the alphabet of edge labels, and each of Z_1 and Z_2 is one of X or Y
- An *approximated single-conjunct query* consists of an expression of the form:

  $Z_1, Z_2 \leftarrow$ APPROX (X, R, Y)

  where X, Y, R, Z_1 and Z_2 are as above
- A *relaxed single-conjunct query* consists of an expression of the form:

  $X \leftarrow$ RELAX $(X, type, c)$

  where X is a constant or variable, and c a constant

Single-Conjunct ApproxRelax Queries

- In the paper, we describe the semantics and evaluation of the above three types of single-conjunct queries
- For queries of the first type, the exact answer can be computed in polynomial time in the size of the database graph and R (Mendelzon & Wood, VLDB 1989)
- For queries of the second type, our ESWC 2009 paper shows how the top-k answer can be computed in polynomial time in the size of the database graph and R
- This computation can be done incrementally
- For queries of the third type, we show in this paper that these too can be computed in polynomial time in the size of the database graph (this is special case of our more general ontology-based query relaxation work, JoDS 2008)
- Again, the computation can be done incrementally

4. General ApproxRelax Queries

- As we saw in the earlier examples, a general ApproxRelax query Q may comprise any number of the three above types of conjuncts on its right hand side, and any number of variables Z_1 ,..., Z_m on its left hand side

- Given any matching θ from variables to the nodes of graph G, the tuple $\theta(Z_1, ...,Z_m)$ has a ***distance to Q*** given by summing the total edit distance of the instantiated APPROX conjuncts and the total relaxation distance of the instantiated RELAX conjuncts (possibly with different weightings applied to each summand)

- The ***top-k answer of Q on G*** is a list containing the k tuples $\theta(Z_1, ...,Z_m)$ with minimum distance to Q, ranked in order of increasing distance to Q

General ApproxRelax Queries

- To ensure polynomial time evaluation, we require that the conjuncts of Q are *acyclic*
- This implies the existence of a join tree induced by the conjuncts of Q, consisting of nodes denoting join operators and nodes representing conjuncts of Q
- For each conjunct of the form (i) (X_i, R_i, Y_i) or (ii) $(X_i, type, c_i)$ of Q, we can use our incremental evaluation techniques for single-conjunct queries to compute a relation r_i
  - In case (i) r_i has scheme (X_i, Y_i, ED, RD) where ED is the edit distance for any tuple in r_i and RD is 0 for all tuples
  - In case (ii) r_i has scheme (X_i, ED, RD) where RD is the relaxation distance of any tuple in r_i and ED is 0 for all tuples

Evaluation

- Construct the join tree E of Q
- Initialise two hash tables for each node JN of E that denotes a join operator
- These tables will hold the results being incrementally computed by the two children nodes of JN
- An initially empty priority queue is allocated for each node JN
- During the evaluation, result tuples will be placed on this priority queue in order of increasing distance value:
  - The attribute ED (RD) of a result tuple u resulting from the join of two tuples s and t holds the combined edit (relaxation) distance value of s and t
  - The overall distance value of u is given by summing the ED and RD values, with possibly different weightings applied to each summand

Evaluation

- Incremental evaluation proceeds by calling a function *getNext* with the root of the join tree E
- If its argument is a query conjunct, *getNext* is as discussed earlier for single-conjunct queries
- If its argument is a join operator, *getNext* is based on the *hash ripple join* algorithm of Ilyas, Aref, Elmagarmid 2004 – see paper for details

5. Conclusions

- Facilitating the collaborative formulation of learning goals and career aspirations has the potential to enhance learners' engagement with the lifelong learning process
- The L4All system offers similarity matching over learners' timelines in order to identify possible choices for future learning and professional development
- Here, we have explored how combining query approximation and query relaxation techniques could provide greater flexibility in querying of heterogeneous timeline data
- Using these querying techniques, users would be able to specify flexible combinations of approximations and relaxations to be applied to their original query, and the relative costs of these
- Query results would be returned incrementally by the system, e.g. as requested interactively by the user, ranked in order of increasing "distance" from the original query

Further Work

- design and prototyping of suitable visual querying interface(s) for users to specify their query approximation and relaxation requirements, and to explore query results
- empirical evaluation of our query processing algorithms over timeline data
- evaluation of the new querying facilities with groups of lifelong learners at our institution and other partner FE/HE institutions
- extending our query processing techniques to return instantiations for *path variables* within queries (see our FQAS'09 paper, forthcoming); these could be used to match *sequences* of timeline episodes rather than just single episodes
- extending our query processing techniques to support querying over *linked data* repositories of timeline information