Combining Flexible Queries and Knowledge Anchors to facilitate the exploration of Knowledge Graphs

Alexandra Poulovassilis[1], Marwan Al-Tawil[2], Riccardo Frosini[1], Mirko Dimartino[1], Vania Dimitrova[2]

[1]Knowledge Lab, Birkbeck University of London, UK
[2]School of Computing, University of Leeds, UK

Abstract. Semantic web and information extraction technologies are enabling the creation of vast information and knowledge repositories, particularly in the form of knowledge graphs comprising entities and the relationships between them. Users are often unfamiliar with the complex structure and vast content of such graphs. Hence, users need to be assisted by tools that support interactive exploration and flexible querying. In this paper we draw on recent work in flexible querying for graph-structured data and identifying good anchors for knowledge graph exploration in order to demonstrate how users can be supported in incrementally querying, exploring and learning from large complex knowledge graphs. We demonstrate our techniques through a case study in the domain of lifelong learning and career guidance.

1 Introduction

Semantic web and information extraction technologies are enabling the creation of vast information and knowledge repositories, particularly in the form of *knowledge graphs* comprising entities and the relationships linking them. As the volume of such data on the web continues to grow, many applications seek to take advantage of knowledge graphs to enable users' knowledge expansion, in domains such as web information retrieval, formal and informal learning, health informatics, entertainment, and cultural heritage preservation, to name a few. However, users are unlikely to be familiar with the full (often very complex) structure and vast content of such datasets. Crucially, users may not have sufficient knowledge to undserstand all the domain entities that they encounter, and hence need to be assisted by intelligent tools that support user-centred interactive exploration of the knowledge graph.

Recent work [16, 4, 20, 8] has proposed techniques for automatic *approximation* and *relaxation* of users' queries over knowledge graphs, allowing query answers to be incrementally returned in order of their 'distance' from the original form of the query. In this context, 'approximating' a query means applying an *edit operation* to the query so that it can return possibly *different* answers, while 'relaxing' a query means applying a *relaxation operation* to the query so that it can return possibly *more* answers. More specifically, the edit operations include the insertion, deletion or substitution of an edge label, while the relaxation operations include replacing a class by a superclass, and replacing a property by a superproperty.

The benefits of supporting such *flexible query processing* over knowledge graphs include: (i) correcting users' erroneous queries; (ii) finding additional relevant answers that the user may be unaware of; and (iii) generating new queries which may return unexpected results and bring new insights. Several example scenarios are presented in [8]. The flexible query processing techniques have been empirically evaluated over standard datasets such as LUBM[1] [4] and YAGO[2] [8], as well as - in [20] - a dataset relating to students' episodes of work and learning arising from the L4All project [5, 16]. However, although flexible query processing allows broadening a user's perspective of the knowlewdge domain, it can return a large number of query results, all at the same 'distance' away from the user's original query. Therefore, a key challenge is *how to faciliate users' meaning making from flexible query results.*

Meaning making is related to users' domain knowledge and their ability to make sense of entities retrieved through their interactions with the knowledge graph. Supporting users' sensemaking and knowledge expansion via interactive nudges has been investigated in [21, 22]. Empirical studies have suggested that paths which *start with familiar entities* and *bring something new* can be beneficial for making sense of complex knowledge graphs [2]. Detecting which entities from a large knowledge graph a user may be familiar with (e.g. by analysing interaction logs) is a tedious and computationally challeging task. Moreover, when the user has had limited interaction with the system, there is the well-known 'cold start' problem, i.e. the system will be unable to detect which parts of the graph the user is familar with. It is therefore necessary to find ways to automatically identify which entities may be close to the users' cognitive structures, and may offer *knowledge anchors* for information exploration. Recent work [1] has proposed an approach to identifying knowledge anchors that adopts the Cognitive Science notion of basic-level objects in domain taxonomies [19], presenting a formal framework for identifying knowledge anchors in knowledge graphs using two complementary approaches: *distinctiveness* and *homogeneity*.

In this paper, we draw on these two recent strands of work (namely, flexible querying of graph-structured data and identification of good anchors for knowledge graph exploration) in order to support users in incrementally querying, exploring and learning from large, complex knowledge graphs. We illustrate this integrative approach through a case study in exploring career options. We begin with a review of related work in Section 2. We describe the case study and the related knowledge graph in Sections 3 and 4. Section 5 describes the derivation of knowledge anchors, and Section 6 demonstrates the integration of flexible querying and knowledge anchors for making sense of query answers. Section 7 gives our concluding remarks.

[1] http://swat.cse.lehigh.edu/projects/lubm/

[2] http://www.mpi-inf.mpg.de/yago-naga/yago/

2 Related Work

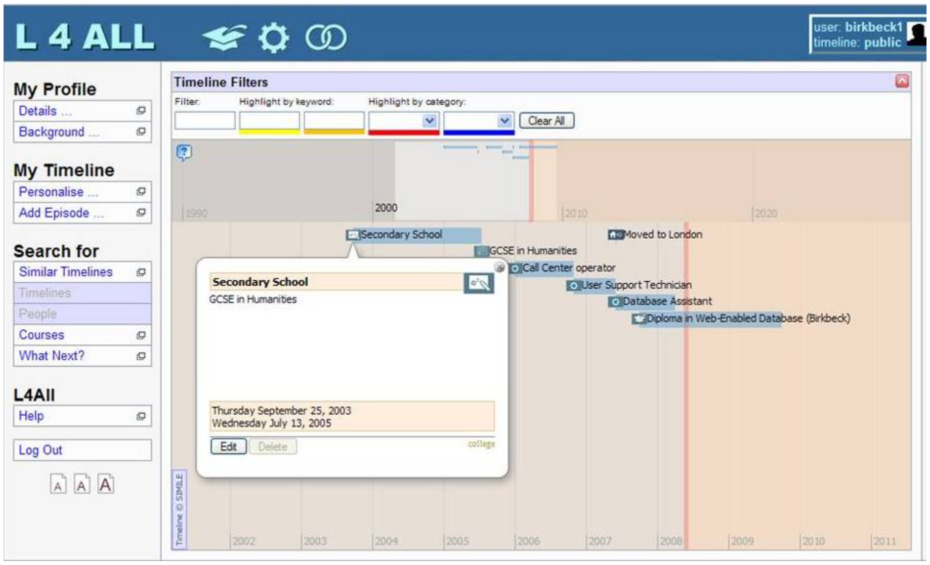
Early work on flexible querying for semi-structured data was undertaken by Kanza and Sagiv [13], who proposed query answering that returns paths in the data whose set of edge labels contain the labels appearing in the query. More generally, Grahne and Thomo [9] used weighted regular transducers to approximate *regular path queries* (RPQs) over semi-structured data. There have been several proposals for flexibly querying Semantic Web data using similarity measures, e.g. [14, 11, 18, 6]. Relaxation of conjunctive queries over RDF data is discussed in [7, 12]. Much work has also been done on relaxing tree-pattern queries over XML data, e.g. [3, 10].

In contrast to the above work, [17] combines query approximation and query relaxation within one framework for querying graph-structured data and applies it to the more general query language of conjunctive RPQs. The prototype implementation described in [16, 20] builds on the theoretical foundations of [17]. More recent work has applied similar query approximation and relaxation techniques to SPARQL 1.1, proposing a language called SPARQL^{AR} that incorporates two query approximation and relaxation operators, APPROX and RELAX, which can be applied to triple patterns within a SPARQL 1.1 query [4, 8].

The closest work to identifying knowledge anchors in knowledge graphs are approaches for ontology summarisation, for example using centrality measures [26]. Measures for ranking using centrality, distance, similarity and coherence have been used to generate explanations of linked data [25]. The notion of relevance based on the relative cardinality and the in/out degree centrality of a graph node has been used in [23] to produce graph summaries. Beyond our work in [1], the closest work to the context of this paper is the summarisation approach presented in [15], which highlighted the value of cognitive science (natural categories) for identifying key concepts in an ontology to aid ontology engineers. However, ontology summarisation aims at identifying key concepts from an ontology in order to help experts to understand and re-engineer the ontology. In contrast, we apply the notion of basic-level objects to identify concepts in a knowledge graph which are likely to be familiar to users who are not domain experts.

3 The Case Study: lifelong learning and career guidance

Our case study is drawn from the domain of lifelong learning and career guidance, specifically, from the L4All project [5, 16]. The L4All project aimed to provide lifelong learners with access to information and resources that would support them in exploring learning and career opportunities and in planning and reflecting on their learning, bringing together experts from lifelong learning and careers guidance, content providers, and groups of students and tutors. The L4All pilot system allowed users to record their past learning, work and life experiences within a ‘timeline’. Figure 1 (from [24]) illustrates the main screen of the L4All user interface. At its centre is a visual representation of the user’s timeline, and the system functionalities are organised around this. Each episode of learning or work is displayed in chronological order, depicted by an icon specific to its type and a horizontal block representing its duration. Details of an episode can be viewed by clicking on the block representing it, which pops-up more detailed information about the episode (dates, description), as well as access to edit and deletion functions.



Secondary School
GCSE in Humanities
Thursday September 25, 2003
Wednesday July 13, 2005

Fig. 1. L4All Main User Interface Screen

Users' timelines are encoded in the form of RDF/S. Some types of episode can be annotated by the user with a primary and possibly a secondary classification. These classifications are drawn from standard occupational and educational taxonomies of the UK Office for National Statistics[3]. In particular, all educational episodes are classified by a subject from the Subject of Degree classification and a qualification level from the National Qualifications Framework; and all occupational episodes are classified by an occupation from the Standard Occupational Classification and an industry activity sector from the Standard Industrial Classification.

Users are able to search over the timelines of other learners and alumni (for those timelines that have been made public by their owner), giving them a repertoire of learning and work possibilities that they may not have otherwise considered, allowing sharing of successful learning pathways, and presenting suc-

[3] See Labour Force Survey User Guide, Vol 5, http://www.ons.gov.uk/ons/guide-method/method-quality/specific/labour-market/labour-market-statistics/index.html

cessful learners as role models to inspire confidence and a sense of opportunity. In the original pilot system, similarity measures were used for comparing users' timelines, based on converting timelines into strings of comparable tokens and using string metrics for ranking them. This allows a user to search for 'people like me' by specifying the desired matching criteria with respect to their own user profile (e.g. age within a certain range, gender, etc.) and timeline (e.g. considering all episodes, considering only educational or professional episodes, etc.). The system returns a list of matching timelines, ranked by similarity. The user can select one of these timelines to visualise in more detail, including clicking on specific episodes to expose their details (similarly to the display in Figure 1).

However, final evaluation of the L4All system (see [24]) identified several problems with the way in which users specify their search queries and with the ranking of the search results, most notably the fact that the top-ranked timelines returned will be timelines that are *most similar* to the user's own timeline hence offering, in practice, few suggestions for the user's future development. This led to further work, reported in [16], that investigated the use of flexible query processing techniques in order to support users' search over the timeline data, including the development of a prototype graphical tool for specifying search queries. Evaluation of this tool with two lifelong learning practitioners found that its ability to specify finer-grained flexible search queries allows a greater number of relevant episodes to be returned to the user. They reported that they found it "much more useful" to be able to explicitly construct a flexible search query rather than using built-in similarity matching based on the user's timeline. They also found it helpful that users can specify what kind of target episode they are looking for, for inspiration. We refer the reader to [16] for full details.

4 The L4All Ontology and Dataset

Our case study in this paper uses the ontology developed by the L4All project, as well as users' data collected during the project (simplified and anonymised, for reasons of privacy). As described above, the L4All system allows users to create and maintain a chronological record — a timeline — of their learning and work episodes. This data and metadata are encoded as RDF/S, as illustrated in Figure 2. In particular, each instance of the Episode class is: linked to a subclass of Episode by an edge labelled rdf:type; linked to other episode instances by edges labelled 'next' or 'prereq' (indicating whether the earlier episode simply preceded, or was necessary in order to be able to proceed to, the later episode); linked either to an occupation or to an educational qualification by means of an edge labelled 'job' or 'qualif'. Each occupation is linked to a subclass of the Occupation class by an edge labelled rdf:type, and to an instance of the Industry Activity Sector class by an edge labelled 'sector'. Each qualification is linked to a subclass of the Subject class by an edge labelled rdf:type and to an instance of the National Qualification Framework class by an edge labelled 'level'. Figure 3 summarises the 5 class hierarchies within the L4All ontology. The first of these was designed by the L4All project while the other four are drawn from standard taxonomies of the UK Office for National Statistics, as described earlier.

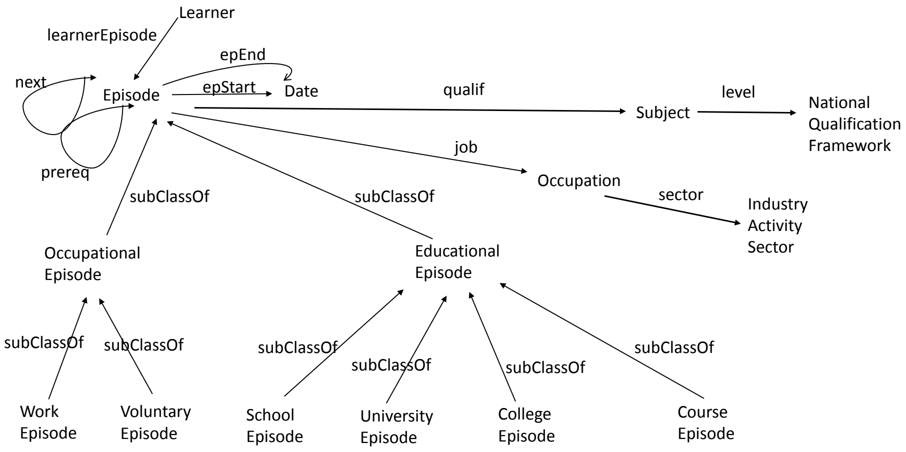


Fig. 2. Fragment of the L4All Ontology

Class hierarchy	**Depth**	**No. of classes**
Episode	3	9
Subject	3	161
Occupation	4	465
National Qualification Framework	3	36
Industry Activity Sector	3	22

Fig. 3. Characteristics of the L4All ontology.

Our initial ‘seed’ data for the case study of this paper comprised 17 timelines sourced from the L4All project — 16 from real users, and 1 demonstration timeline. Each timeline consists of a mixture of educational and occupational episodes, and they vary in terms of the number of episodes contained within them, as well as the classification of each episode. The average number of episodes per timeline is 5.3 and the total number of epsides is 91.

We scaled up this data by creating duplicate versions of the 17 timelines, using a similar technique to that used in [20]: each duplicated timeline is identical to the original in terms of the number of episodes, whether the type of the episode is educational or occupational, and the way in which episodes are linked to each other; the ontology is used to alter the classification of each episode to be a 'sibling' class of its original class, thereby maintaining realistic timelines. In this way, we created successively larger datasets: L4All1 - seed data replicated 10 times, L4All2 - seed data x 100, L4All3 - seed data x 1000 etc. For example, a dataset of the size of L4All2 could arise from the student/alumni network of a single Higher Education institution; while a dataset of the size of L4All3 could arise from a federation of institutions. For the remainder of this paper, we focus on dataset L4All2.

5 Knowledge Anchors in the L4All Ontology and Dataset

Our work in [1] utilized the definition of Basic Level Objects introduced in the seminal work by Rosch et al. [19], to develop algorithms for identifying knowledge anchors in data graphs. Knowledge anchors represent *familiar and highly inclusive entities* in the graph which can be used as knowledge bridges to direct users to familiar entities from where links to new knowledge can be made. This new knowledge can take meaning by becoming anchored with basic concepts in the user's cognitive structures. The performance of our algorithms for identifying knowledge anchors was evaluated through an experimental study in the Music domain involving free naming tasks by humans [1].

Identifying anchoring entities is not a trivial task because the graph may include thousands of entities at different levels of abstraction. We consider two types of relationships: (i) *hierarchical relationships* denoting membership between the subject and object of the corresponding RDF triples (for our case study here, we use two hierarchical relationships: rdfs:subClassOf and rdf:type); (ii) *domain-specific relationships* are properties other than the hierarchical relationships, representing shared attributes between entities. We adopt two complementary groups of metrics from [19, 1] to identify knowledge anchors (the reader is directed to [1] for a detailed description of the various metrics and algorithms):

(i) *Distinctiveness metrics* identify the most differentiated categories, whose attributes are associated with the category members but not with members of other categories. We use three distinctiveness metrics: *attribute validity*, which is proportional to the number of relationships of a specific type involving the category's sub-classes; *category-attribute collocation*, which takes into account the frequency of an attribute within the members of a category and gives preference to categories that have many attributes shared by their members; *category validity*, which takes into account whether a category has many attributes shared by its members but at the same time has attributes that are not related to many other categories.

(ii) *Homogeneity metrics* identifie categories whose members share many attributes. We use three set-based similarity metrics: *Common Neighbors*, *Jaccard*, and *Cosine*.

The above six metrics are calculated for each class entity in the graph, considering both its hierarchical and its domain-specific relationships. Hence, for each class we obtain a set of scores that rate that entity's suitability as a knowledge anchor. We combine these scores using majority voting, selecting entities that have at least 50% non-zero scores, subject to the constraint that a knowledge anchor should have at least one non-zero score from the subset of hierarchical relationships and at least one from the subset of domain-specific relationships. For example, the knowledge anchors identified within Occupation hierarchy of the L4All2 dataset include: Administrative_Occupations, Associate_Professional_and_Technical_Occupations, Managers_and_Senior_Officials, Professional_Occupations, Teaching_and_Research_Professionals.

6 Extending Flexible Querying with Knowledge Anchors

We now demonstrate how the combination of flexible querying and knowledge anchors can assist users' querying of the L4All dataset. We focus on querying the L4All2 dataset (relating to 1700 learners and 9100 episodes).

Example 1. Suppose the user is currently studying for a BSc in Information Systems and wishes to find out what possible future job choices there are by seeing what other people with qualifications in Information Systems, or similar, have gone on to do. This can be undertaken by evaluating the following SPARQL^{AR} query [4, 8] over the L4All data (of course, this is not what an end-user would enter, but it could be the SPARQL^{AR} query generated from the user's interaction with a graphical user interface, such as that described in [16]):

```
SELECT ?WorkEp ?Occ
WHERE {?EdEp  rdf:type  <http://www.L4All.com/University_Episode>.
       RELAX ( ?EdEp <http://www.L4All.com/qualif>/rdf:type
                     <http://www.L4All.com/Information_Systems> ).
       APPROX( ?EdEp <http://www.L4All.com/prereq> ?WorkEp ).
       ?WorkEp  rdf:type  <http://www.L4All.com/Work_Episode>.
       ?WorkEp  <http://www.L4All.com/job>/rdf:type ?Occ}
```

Briefly, this query returns a list of triples (w, o, d) such that:

- w is a Work Episode (an instantiation of the variable `?WorkEp` in the fourth triple pattern);
- o is the Occupation associated with w (instantiation of the variable `?Occ` in the fifth triple pattern);
- there is a University Episode e (instatiation of the variable `?EdEp` in the first triple pattern) whose subject is `Information_Systems` (second triple pattern), or similar (operator RELAX); and that is connected by a `prereq` edge to w (third triple pattern), or by a similar path (operator APPROX);
- d is the 'distance' of the answer (w, o) from the exact form of the query (i.e. the query without any APPROX or RELAX operators applied); this distance is the sum of the costs of all the edit and relaxation operations that have been applied to the exact form of the query in order to obtain the answer (w, o); for the purposes of this paper, we assume that all edit

and relaxation operations have a cost of 1 — but in general these can be application-defined or user-defined.

Suppose that, before running the query, the user elects to apply two of the edit operations that are available as part of the APPROX operator: Insertion of an edge label, and Substitution of an edge label. The user also selects one relaxation operation from those available as part of the RELAX operator: Replacement of a subclass by its immediate superclass.

Suppose the user asks first for the exact answers to the query (i.e. answers at distance 0, without any approximation or relaxation being applied to the query). There are 86 such answers and the top 20 of them are listed in Figure 4 (the namespace prefixes have been removed). If the user is interested in one of the suggested occupations (listed in the second column), he/she can click (through an appropriate GUI) on the URL appearing in the first column in order to retrieve the entire timeline that this episode belongs to, and can interact with a visualisation of the timeline similar to that in Figure 1.

?WorkEp	**?Occ**
A_4_E_14_22	Research_Professionals
A_1_E_8_98	IT_User_Support_Technicians
A_3_E_5_37	Software_Professionals
A_7_E_7_92	Engineering_Technicians
A_7_E_7_4	Quality_Assurance_Technicians
A_7_E_7_60	Quality_Assurance_Technicians
A_2_E_6_11	Purchasing_Managers
A_8_E_5_88	Pensions_and_Insurance_Clerks
A_8_E_6_88	Physicists,_Geologists_and_Meterologists
A_1_E_8_58	IT_User_Support_Technicians
A_4_E_14_14	Registrars_and_Senior_Administrators_of_Educational_Establishments
A_7_E_7_52	Architectural_Technologists_and_Town_Planning_Technicians
A_4_E_14_78	Primary_and_Nursery_Education_Teaching_Professionals
A_7_E_7_20	Laboratory_Technicians
A_4_E_14_46	Scientific_Researchers
A_7_E_7_84	IT_Service_Delivery_Occupations
A_8_E_5_24	Stock_Control_Clerks
A_8_E_6_24	Electronics_Engineers
A_3_E_5_21	Software_Professionals
A_2_E_6_3	Advertising_and_Public_Relations_Managers

Fig. 4. Top 20 exact query answers

After doing this for a while, the user next asks for query answers at distance 1, with the aim of obtaining a greater diversity of timelines and possibly additional suggested occupations. There are 237 answers at distance 1 and the top 20 are listed in Figure 5. The user can again view the suggested occupations in the second column, click on a URL in the first column to retrieve an entire timeline, and interact with a visualisation of the timeline. The user can continue their querying and timeline visualisation in this fashion, successively asking for more answers at increasing distances.

?WorkEp	**?Occ**	**Distance**
A_2_E_6_51	Personnel,_Training_and_Industrial_Relations_Managers	1
A_7_E_7_12	IT_User_Support_Technicians	1
A_8_E_5_56	Market_Research_Interviewers	1
A_7_E_7_44	Science_and_Engineering_Technicians	1
A_8_E_6_72	Civil_Engineers	1
A_8_E_5_64	Library_Assistants/Clerks	1
A_8_E_5_16	Transport_and_Distribution_Clerks	1
A_7_E_7_28	IT_Service_Delivery_Occupations	1
A_8_E_6_8	Mechanical_Engineers	1
A_8_E_5_96	Filing_and_Other_Records_Assistants/Clerks	1
A_2_E_6_35	Research_and_Development_Managers	1
A_8_E_6_40	Planning_and_Quality_Control_Engineers	1
A_4_E_14_6	Social_Science_Researchers	1
A_8_E_5_8	Library_Assistants/Clerks	1
A_1_E_8_50	IT_User_Support_Technicians	1
A_7_E_7_68	IT_User_Support_Technicians	1
A_1_E_8_74	IT_User_Support_Technicians	1
A_1_E_8_34	IT_User_Support_Technicians	1
A_7_E_7_60	Quality_Assurance_Technicians	1
A_3_E_5_69	Software_Professionals	1

Fig. 5. Top 20 query answers at distance 1

It is evident that, although it can return relevant and useful answers for the user, this kind of incremental flexible querying can easily result in information overload. Moreover, the user may unfamiliar with some of the specialist terminology relating to occupations. The user will also gain little insight into the relationships between the different occupations being suggested and how they are categorised within the broader context of the Occupation hierarchy.

Example 2. Repeating the above query and user interactions, let us consider an alternative presentation of the results as *paths* within the Occupation hierachy, *rooted at the nearest Knowledge Anchor*. The user again asks first for the exact answers to the query. Suppose answers are returned incrementally, 10 at a time. The first 10 answers would be displayed as shown in Figure 6. We see again the same top 10 exact results as earlier, except this time the suggested occupation and the URL(s) of the corresponding work episodes are shown as leaves within fragments of the Occupation hierarchy, each fragment rooted at a Knowledge Anchor. The user can again click on a URL representing a work episode to retrieve and interact with an entire timeline. However, the user can now also see one or more ancestor classes of the suggested occupations. Moreover, the user can click on a Knowledge Anchor to explore the Occupation hierarchy further, independently of the results of this particular query.

```
KA:Associate_Professional_and_Technical_Occupations
     Science_and_Technology_Associate_Professional
       IT_Service_Delivery_Occupations
         IT_User_Support_Technicians A_1_E_8_98, A_1_E_8_58 (0)
     Science_and_Engineering_Technicians
       Engineering_Technicians A_7_E_7_92 (0)
       Quality_Assurance_Technicians A_7_E_7_4, A_7_E_7_60 (0)
KA:Managers_and_Senior_Officials
     Corporate_Managers
       Functional_Managers
         Purchasing_Managers A_2_E_6_11 (0)
KA:Professional_Occupations
     Science_and_Technology_Professionals
       Information_and_Communication_Technology_Professionals
         Software_Professionals A_3_E_5_37 (0)
       Science_Professionals
         Physicists,_Geologists_and_Meterologists A_8_E_6_88 (0)
KA:Teaching_and_Research_Professionals
     Research_Professionals A_4_E_14_22 (0)
KA:Administrative_Occupations
     Administrative_Occupations:_Records
       Pensions_and_Insurance_Clerks A_8_E_5_88 (0)
```

Fig. 6. Top 10 query answers at distance 0, under Knowledge Anchors (KA)

If the user asks for 10 more answers, the display of results would expand as shown in Figure 7. We see that for some occupations there are additional suggested episodes, that when clicked on can lead to additional timelines for possible inspiration. Moreover, there are additional suggested occupations that extend the hierarchy fragments under some of the Knowledge Anchors, e.g. Primary_and_Nursery_Education_Teaching_Professionals, Electronics_Engineers.

Suppose the user next asks for query answers at distance 1, with the aim of obtaining a greater diversity of timelines and possibly additional suggested occupations. Figure 8 illustrates how the top 20 answers at distance 1 would be presented, added to the already retrieved top 20 answers at distance 0. We again see that for some occupations there are more suggested episodes, which when clicked on lead to more timelines; and that are also additional suggested occupations, extending the hierarchy fragments under some of the Knowledge Anchors, e.g. Research_and_Development_Managers, Social_Science_Researchers. The user can again click on URLs representing work episodes, retrieve and interact with entire timelines, and click on a Knowledge Anchor to explore the Occupation hierarchy further. The user can continue their incremental querying, timeline

```
KA:Associate_Professional_and_Technical_Occupations
     Science_and_Technology_Associate_Professional
       IT_Service_Delivery_Occupations A_7_E_7_84 (0)
         IT_User_Support_Technicians A_1_E_8_98, A_1_E_8_58 (0)
     Science_and_Engineering_Technicians
       Engineering_Technicians A_7_E_7_92 (0)
       Quality_Assurance_Technicians A_7_E_7_4, A_7_E_7_60 (0)
       Laboratory_Technicians A_7_E_7_20 (0)
     Draughtspersons_and_Building_Inspectors
       Architectural_Technologists_and_Town_Planning_Technicians A_7_E_7_52 (0)
KA:Managers_and_Senior_Officials
     Corporate_Managers
       Functional_Managers
         Advertising_and_Public_Relations_Managers A_2_E_6_3 (0)
         Purchasing_Managers A_2_E_6_11 (0)
KA:Professional_Occupations
     Science_and_Technology_Professionals
       Information_and_Communication_Technology_Professionals
         Software_Professionals A_3_E_5_37, A_3_E_5_21 (0)
       Science_Professionals
         Physicists,_Geologists_and_Meterologists A_8_E_6_88 (0)
       Engineering_Professionals
         Electronics_Engineers A_8_E_6_24 (0)
KA:Teaching_and_Research_Professionals
     Research_Professionals A_4_E_14_22 (0)
       Scientific_Researchers A_4_E_14_46 (0)
     Teaching_Professionals
       Registrars_and_Senior_Administrators_of_Educational_Establishments A_4_E_14_14 (0)
       Primary_and_Nursery_Education_Teaching_Professionals A_4_E_14_78 (0)
KA:Administrative_Occupations
     Administrative_Occupations:_Records
       Pensions_and_Insurance_Clerks A_8_E_5_88 (0)
       Stock_Control_Clerks A_8_E_5_24 (0)
```

Fig. 7. Top 20 query answers at distance 0, under Knowledge Anchors (KA)

```
KA:Associate_Professional_and_Technical_Occupations
     Science_and_Technology_Associate_Professional
       IT_Service_Delivery_Occupations A_7_E_7_84 (0); A_7_E_7_28 (1)
         IT_User_Support_Technicians A_1_E_8_98, A_1_E_8_58 (0);
            A_7_E_7_12, A_1_E_8_50, A_7_E_7_68, A_1_E_8_74, A_1_E_8_34 (1)
     Science_and_Engineering_Technicians A_7_E_7_44 (1)
       Engineering_Technicians A_7_E_7_92 (0)
       Quality_Assurance_Technicians A_7_E_7_4, A_7_E_7_60 (0); A_7_E_7_60 (1)
       Laboratory_Technicians A_7_E_7_20 (0)
     Draughtspersons_and_Building_Inspectors
       Architectural_Technologists_and_Town_Planning_Technicians A_7_E_7_52 (0)
KA:Managers_and_Senior_Officials
     Corporate_Managers
       Functional_Managers
         Advertising_and_Public_Relations_Managers A_2_E_6_3 (0)
         Purchasing_Managers A_2_E_6_11 (0)
         Personnel,_Training_and_Industrial_Relations_Managers A_2_E_6_51 (1)
         Research_and_Development_Managers A_2_E_6_35 (1)
KA:Professional_Occupations
     Science_and_Technology_Professionals
       Information_and_Communication_Technology_Professionals
         Software_Professionals A_3_E_5_37, A_3_E_5_21 (0); A_3_E_5_69 (1)
       Science_Professionals
         Physicists,_Geologists_and_Meterologists A_8_E_6_88 (0)
       Engineering_Professionals
         Electronics_Engineers A_8_E_6_24 (0)
         Civil_Engineers A_8_E_6_72 (1)
         Mechanical_Engineers A_8_E_6_8 (1)
         Planning_and_Quality_Control_Engineers A_8_E_6_40 (1)
KA:Teaching_and_Research_Professionals
     Research_Professionals A_4_E_14_22 (0)
       Scientific_Researchers A_4_E_14_46 (0)
       Social_Science_Researchers A_4_E_14_6 (1)
     Teaching_Professionals
       Registrars_and_Senior_Administrators_of_Educational_Establishments A_4_E_14_14 (0)
       Primary_and_Nursery_Education_Teaching_Professionals A_4_E_14_78 (0)
KA:Administrative_Occupations
     Administrative_Occupations:_Records
       Pensions_and_Insurance_Clerks A_8_E_5_88 (0)
       Stock_Control_Clerks A_8_E_5_24 (0)
       Market_Research_Interviewers A_8_E_5_56 (1)
       Library_Assistants/Clerks A_8_E_5_64 (1)
       Transport_and_Distribution_Clerks A_8_E_5_16 (1)
       Filing_and_Other_Records_Assistants/Clerks A_8_E_5_96 (1)
```

Fig. 8. Top 20 query answers at distances 0 and 1, under Knowledge Anchors (KA)

visualisation, and Occupation exploration in this fashion, successively asking for more answers at increasing distances.

This alternative presentation makes more evident the relationships between the occupations returned as query results, and allows in parallel the user to explore increasingly larger fragments of the Occupation hierarchy, each rooted at a Knowledge Anchor that may be more meaningful to the user than a specialist occupation. We argue that this faciliates increasing awareness of possible relevant occupations by the user. In our future work we will undertake trials with groups of students and practitioners from lifelong learning and careers guidance to investigate this hypothesis through user evaluation activities that involve comparison of the two alternative forms of results presentation.

7 Concluding Remarks

The work presented in this paper addresses the challenge of supporting the exploration of large knowledge graphs by users who are not experts in the domain. We have proposed a hybrid approach combining flexible graph querying and knowledge anchors. Flexible queries allow automatic expansion of query results by query approximation and query relaxation. While this facilitates knowledge graph exploration when the user may not be fully aware of the domain entities and relationships, it can still be challenging for the user to make sense of a large number of query answers. Knowledge anchors, representing basic-level entities that are close to the user's cognitive structures, are likely to be familiar to many users and can provide starting points for introducing unfamiliar entities. In our hybrid approach, we introduce knowledge anchors into query results by including paths to the nearest knowledge anchor. Our hybrid approach has been applied to interacting with an existing knowledge graph for exploring future career options. Our immediate plans are to develop interactive visualisations such as those illustrated in Example 2 and Figures 6-8, evaluate the approach with groups of students and practitioners, and also investigate other ways of hybridising flexible queries and knowledge anchors, e.g. for filtering or ranking query results.

References

1. M. Al-Tawil, V. Dimitrova, D. Thakker, and B. Bennett. Identifying knowledge anchors in a data graph. In *27th ACM Conf. on Hypertext and Social Media*, 2016.
2. M. Al-Tawil, D. Thakker, and V. Dimitrova. Nudging to expand user's domain knowledge while exploring linked data. In *3rd Int. Workshop on Intelligent Exploration of Semantic Data*, pages 54–65, 2014.
3. S. Amer-Yahia, L. V. S. Lakshmanan, and S. Pandit. FleXPath: Flexible structure and full-text querying for XML. In *Proc. ACM SIGMOD 2004*, pages 83–94.
4. A. Calì, R. Frosini, A. Poulovassilis, and P. T. Wood. Flexible querying for SPARQL. In *Proc. ODBASE 2014 (OTM Conferences)*, pages 473–490, 2014.
5. S. de Freitas et al. L4All: a web-service based system for lifelong learners. In *The Learning Grid Handbook: Concepts, Technologies and Applications, Volume 2: The Future of Learning.* IOS Press, 2008.

6. R. De Virgilio, A. Maccioni, and R. Torlone. A similarity measure for approximate querying over RDF data. In *Proc. EDBT/ICDT 2013 Workshops*, pages 205–213.
7. P. Dolog, H. Stuckenschmidt, H. Wache, and J. Diederich. Relaxing RDF queries based on user and domain preferences. *J. Intell. Inf. Syst.*, 33(3):239–260, 2009.
8. R. Frosini, A. Calì, A. Poulovassilis, and P. T. Wood. Flexible query processing for SPARQL. *To appear in the Semantic Web Journal*, 2016.
9. G. Grahne and A. Thomo. Regular path queries under approximate semantics. *Ann. Math. Artif. Intell.*, 46(1-2):165–190, 2006.
10. J. Hill, J. Torson, B. Guo, and Z. Chen. Toward ontology-guided knowledge-driven XML query relaxation. In *Proc. 2nd Int. Conf. on Computational Intelligence, Modelling and Simulation (CIMSiM) 2010*, pages 448–453.
11. H. Huang and C. Liu. Query relaxation for star queries on RDF. In *Proc. WISE 2010*, pages 376–389.
12. C. A. Hurtado, A. Poulovassilis, and P. T. Wood. Query relaxation in RDF. *Journal on Data Semantics*, X:31–61, 2008.
13. Y. Kanza and Y. Sagiv. Flexible queries over semistructured data. In *Proc. PODS 2001*, pages 40–51, 2001.
14. F. Mandreoli, R. Martoglia, G. Villani, and W. Penzo. Flexible query answering on graph-modeled data. In *Proc. EDBT 2009*, pages 216–227, 2009.
15. S. Peroni, E. Motta, and M. D'Aquin. Identifying key concepts in an ontology, through the integration of cognitive principles with statistical and topological measures. In *Proc. ASWC 2008*, pages 242–256. Springer-Verlag, 2008.
16. A. Poulovassilis, P. Selmer, and P. T. Wood. Flexible querying of lifelong learner metadata. *IEEE Transactions on Learning Technologies*, 5(2):117–129, 2012.
17. A. Poulovassilis and P. T. Wood. Combining approximation and relaxation in semantic web path queries. In *Proc. ISWC 2010*, pages 631–646, 2010.
18. B. R. K. Reddy and P. S. Kumar. Efficient approximate SPARQL querying of web of linked data. In *Proc. URSW 2010*, pages 37–48.
19. E. Rosch, C. B. Mervisa, W. D. Gray, D. M. Johnson, and P. Boyes-Braem. Basic objects in natural categories. *Cognitive Psychology*, 8(3):382 – 439, 1976.
20. P. Selmer, A. Poulovassilis, and P. T. Wood. Implementing flexible operators for regular path queries. In *Proc. GraphQ 2015 (EDBT/ICDT Workshops)*, pages 149–156, 2015.
21. D. Thakker, V. Dimitrova, L. Lau, F. Yang-Turner, and D. Despotakis. Assisting user browsing over linked data: requirements elicitation with a user study. In *13th Int. Conf. on Web Engineering*, pages 376–383, 2013.
22. D. Thakker, V. Dimitrova, L. Lau, F. Yang-Turner, and D. Despotakis. Making sense of linked data: A semantic exploration approach. In *Mastering Data-Intensive Collaboration and Decision Making*, pages 71–87, 2014.
23. G. Troullinou, H. Kondylakis, E. Daskalaki, and D. Plexousakis. *RDF Digest: Efficient Summarization of RDF/S KBs*, pages 119–134. Springer International Publishing, 2015.
24. N. van Labeke, G. D. Magoulas, and A. Poulovassilis. Personalised search over lifelong learner's timelines using string similarity measures. Technical Report BBKCS-11-01, Birkbeck, 2011. Available at `http://www.dcs.bbk.ac.uk/research/techreps/2011/bbkcs-11-01.pdf`.
25. R. Wille. *Formal Concept Analysis as Mathematical Theory of Concepts and Concept Hierarchies*, pages 1–33. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.
26. X. Zhang, G. Cheng, and Y. Qu. Ontology summarization based on RDF sentence graph. In *Proc. WWW 2007*, pages 707–716, 2007.