Aemoo: Linked Data exploration based on Knowledge Patterns

Andrea Giovanni Nuzzolese\textsuperscript{a}*, Valentina Presutti\textsuperscript{b}, Aldo Gangemi\textsuperscript{c}, Silvio Peroni\textsuperscript{b}, and Paolo Ciancarini\textsuperscript{a},\textsuperscript{b}

\textsuperscript{a}STLab-ISTC, Consiglio Nazionale delle Ricerche, Rome, Italy
\textsuperscript{b}Department of Computer Science and Engineering, University of Bologna, Bologna, Italy
\textsuperscript{c}LIPN, Université Paris 13, Sorbonne Cité, UMR CNRS, Paris, France

Abstract. This paper presents a novel approach to Linked Data exploration that uses Encyclopedic Knowledge Patterns (EKPs) as relevance criteria for selecting, organising, and visualising knowledge. EKP are discovered by mining the linking structure of Wikipedia and evaluated by means of a user-based study, which shows that they are cognitively sound as models for building entity summarisations. We implemented a tool named Aemoo that supports EKP-driven knowledge exploration and integrates data coming from heterogeneous resources, namely static and dynamic knowledge as well as text and Linked Data. Aemoo is evaluated by means of controlled, task-driven user experiments in order to assess its usability, and ability to provide relevant and serendipitous information as compared to two existing tools: Google and RelFinder.

Keywords: Exploratory search, Knowledge exploration, Visual Exploration, Knowledge Patterns, Analysis of Linked Data

1. Introduction

In the Semantic Web vision\textsuperscript{[7]} agents were supposed to leverage the Web knowledge in order to help humans in solving knowledge-intensive tasks. Nowadays, Linked Data is feeding up the Semantic Web by publishing datasets by relying on URIs and RDF. However, it is still difficult to enable homogeneous and contextualised access to Web knowledge, for both humans and machines, because of the heterogeneity of Linked Data and the lack of relevance criteria (aka knowledge boundary\textsuperscript{[20]}) for providing tailored data.

The heterogeneity consists of different data semantics, ontologies, and vocabularies used in linked datasets. In fact, Linked Data is composed of datasets from different domains (e.g., life science, geography, government). Moreover, some of them classify data according to a reference ontology (e.g., DBpedia) and others just provide access to raw RDF data. For example, if we consider the case of aggregating data from different linked datasets we need a shared intensional meaning over the things described in these datasets in order to properly mash up facts about those things. The scenario is even more complex if we also take into account dynamic data coming from a variety of sources like social streams (e.g., Twitter) and news (e.g., Google News).

Instead, the knowledge boundary problem consists of identifying what data are really meaningful with respect to specific application tasks. Identifying meaningful data involves the need of establishing a clear relevance criteria to be applied as a filter on data. As an example, we can consider an application that leverages Linked Data to provide a summary on some topic. For example, if the topic of the summary is the philosopher Immanuel Kant, this application should provide users with tailored information concerning, for example, facts about Kant’s major works and thoughts, and should skip more peculiar or curious facts, such as the nationality of his grandfather.

Elsewhere\textsuperscript{[20]} we introduced a vision for the Semantic Web based on Knowledge Patterns (KPs) as basic unit of meaning for addressing both the hetero-
geneity and the knowledge boundary problems. More recently we introduced the Encyclopedic Knowledge Patterns [36] (EKPs), a particular kind of KPs, that are empirically discovered by mining the linking structure of Wikipedia articles. EKPs provide knowledge unit for answering to the following competency question:

What are the most relevant entity types \( (T_1, \ldots, T_n) \) that are involved in the description of an entity of type \( C \)?

For example, the EKP for describing philosophers should include types such as book, university, and writer because in Wikipedia a philosopher is typically linked to these entity types.

We assume that EKPs are cognitively sound because they emerge from the largest existing multi-domain knowledge source, collaboratively built by humans with an encyclopedic task in mind.

In this paper we want to exploit the EKPs for designing a system that helps humans to address summarisation and discovery tasks. These tasks can be classified as exploratory search tasks as they involve different phases, i.e., look-up, learning and investigation [28], that characterise the strategies that humans adopt while exploring the Web. For example, consider a student who is asked to build a concept map about a topic for her homework. She starts by looking-up specific terms in a keyword-based search engine (e.g., Google), then she moves through search results and hyperlinks in order to investigate the available information about the topic for finally learning new knowledge she can use to address her task.

Hypotheses. The work presented in this paper grounds on two working hypotheses: (i) EKPs are cognitively sound and provide a unifying view and a relevance criterion for building entity-centric summaries and (ii) they can be exploited effectively for helping humans in exploratory search tasks.

Contribution. The contributions of this paper are:

- a method for extracting EKPs by mining the structure of Linked Data;
- a method for applying EKPs in order to provide entity summarisation and knowledge aggregation;
- a visual exploratory search system, called Aemoo [1] that uses EKPs as criteria for enriching and filtering data;
- an evaluation based on user-study of the cognitive soundness of the EKPs;
- an evaluation of Aemoo by means of controlled, task-driven user experiments in order to assess its usability, and ability to provide relevant and serendipitous information.

It is worth mentioning that there are state of the art systems that provide semantic mash-up and browsing capabilities, such as [53,24,25]. However, they mostly focus on presenting linked data coming from different sources, and visualising it in interfaces that mirror the linked data structure. Instead, Aemoo organises and filters the retrieved knowledge in order to show only relevant information to users, and providing the motivation of why a certain piece of information is included. The rest of the paper is organised as follows: Section 2 presents a method for extracting EKPs from the Wikipedia; Section 3 presents Aemoo, a system based on EKPs for knowledge exploration; Section 4 describes the experiments we conducted for evaluating the EKPs and the system; Section 5 discusses evaluation results, limits and possible solutions to improve the system; Section 6 presents the related work; finally, Section 7 summarises the contribution and illustrates future work.

2. Encyclopedic Knowledge Patterns

A general formal theory for Knowledge Patterns (KPs) does not exist yet. Different independent theories have been developed so far and KPs have been proposed with different names and flavours across different research areas, such as linguistics [17], artificial intelligence [30], cognitive sciences [4,19] and more recently in the Semantic Web [20]. According to [20] it is possible to identify a shared meaning for KPs across these different theories, that can be informally summarized as “a structure that is used to organize our knowledge, as well as for interpreting, processing or anticipating information”.

In linguistics KPs were introduced as frames by Fillmore in 1968 [17] in his work about case grammar. In a case grammar, each verb selects a number of deep cases which form its case frame. A case frame describes important aspects of semantic valency, verbs, adjectives and nouns. Fillmore elaborated further the initial theory of case frames and in 1976 he introduced frame semantics [18]. According to Fillmore a frame is
A frame is comparable to a cognitive schema. It has a prototypical form that can be applied to a variety of concrete cases that fit this prototypical form. According to cognitive science theories [4], humans are able to recognize frames, to apply them several times, in what are called manifestations of a frame, and to learn new frames that became part of their background. Hence, frames (aka KPs) are cognitively relevant, since they are used by humans to successfully interact with their environment, when some information structuring is needed.

In computer science frames were introduced by Minsky, who recognized that frames convey both cognitive and computational value in representing and organizing knowledge. The notion of frame, was formalized by Minsky [30] as:

“... a remembered framework to be adapted to fit reality by changing details as necessary. A frame is a data-structure for representing a stereotyped situation, like being in a certain kind of living room, or going to a child’s birthday party.” [30]

In knowledge engineering the term Knowledge Pattern was used by Clark [10]. However, the notion of KP Clark introduced is slightly different from frames as introduced by Fillmore and Minsky. In fact, according to Clark, KPs are first order theories which provide a general schema able to provide terminological grounding and morphisms for enabling mappings among knowledge bases that use different terms for representing the same theory. Though Clark recognizes KPs as general templates denoting recurring theory schemata, his approach is similar to the use of theories and morphisms in the formal specification of software. Moreover, Clark’s KPs lack the cognitive value as it is for frames. This makes difficult to use this formalization of KPs for representing contextual relevant knowledge.

More recently Knowledge Patterns have been proposed in the context of the Semantic Web by Gangemi and Presutti in [20]. Their notion of KPs encompasses those proposed by Fillmore and Minsky and goes further envisioning KPs as the research objects of the Semantic Web as an empirical science.

In [36] we introduced the Encyclopedic Knowledge Patterns (EKPs). EKPs were discovered by mining the structure of Wikipedia articles. EKPs are a special type of knowledge patterns: they express the core elements that are used for describing entities of a certain type with an encyclopedic task in mind. The cognitive soundness of EKPs is bound to an important working hypothesis about the process of knowledge construction realized by the Wikipedia crowds: each article is linked to other articles when explaining or describing the entity referred to by the article. DBpedia, accordingly with this hypothesis, has RDF-ized a) the subjects referred to by articles as resources, b) the wikilinks as relations between those resources, and c) the types of the resources as OWL classes. EKPs grounds on the assumption that wikilink relations in DBpedia, i.e. instances of the dbpo:wikiPageWikiLink property, convey a rich encyclopedic knowledge that can be formalized as knowledge patterns.

Informally, an EKP is a small ontology that contains a concept $S$ and its relations to the most relevant concepts $C_j$ that can be used to describe $S$. In [36] we defined a method for extracting EKPs by analysing the structure of wikilinks. This method is based on three main steps, i.e.:

- gathering the knowledge architecture of a dataset. The goal of this step is to create a model able to provide an overview over the structure of wikilinks by exploiting the dataset of wikipedia page links available in DBpedia. For this purpose, we used an OWL vocabulary, called knowledge architecture[33], which allows to gather a landscape view on a dataset even with no prior knowledge of its vocabulary. In our case the overviews are focused on the identification of type paths in the dataset of wikilinks. A type path is a property path (limited to length 1 in this work, i.e. a triple pattern), whose occurrences have (i) the same rdf:type for their subject nodes, and (ii) the same rdf:type for their object nodes.

- EKP discovery. This step is focused on the discovery of EKPs emerging from data in a bottom-

---


3The dataset is named dbpedia_page_links_en at http://www.ontologydesignpatterns.org/ont/lod-analysis-path.owl
up fashion. For this purpose we used a function, called $\text{pathPopularity}(P_{i,j}, S_i)$, which records the ratio of how many distinct resources of a certain type participate as subject in a path to the total number of resources of that type. Intuitively, it indicates the popularity of a path for a certain subject type within a dataset. An EKP for a certain type $S$ is identified by the set type paths having $S$ as subject type whose $\text{pathPopularity}(P_{i,j}, S_i)$ values are above a certain threshold $t$ ranging from 0 to 1. In our study we fixed $t = 0.18$.

- OWL2 formalization of EKPs. In this step we apply a refactoring procedure to the dataset resulting from the previous steps in order to formalise EKPs as OWL2 ontologies.

Figure 1 shows the EKP ekp:Philosopher resulting from our method. It reports the types of entities that are most frequently (according to $\text{pathPopularity}$ values) associated with entities typed as philosophers in Wikipedia by means of wikilink relations. According to the method described we generated 231 EKPs out of 272 DBPO classes and published them into the ODP repository. We recommend further reading where more details about the extraction method and the rationale behind the identification of the threshold $t$ are provided.

3. Encyclopedic Knowledge Patterns as relevance criteria for exploratory search

Aemoo is a tool that exploits relevance strategies based on EKPs for supporting exploratory search tasks. It uses EKPs as a unifying view for aggregating knowledge from static (i.e., DBpedia and Wikipedia) as well as dynamic (i.e., Twitter and Google News) sources. A preliminary version of Aemoo was presented in 37.

We assume that the action performed by humans of linking entities on the Web reflects the way they organise their knowledge. EKPs reflect the most frequent links between entity types, hence our hypothesis is that EKPs can be used for selecting the most relevant entities to be included in an entity-centric summary that can support users in knowledge exploration. In fact, Aemoo novelty is its ability to build entity-centric summaries by applying EKPs as lenses over data. In this way, Aemoo performs both enrichment

\[ \text{http://ontologydesignpatterns.org/aemoo/} \]
and filtering of information. Enrichment and filtering are the two actions performed in order to address the knowledge heterogeneity and boundary problems. In fact, users are guided through their navigation: instead of being presented with a bunch of triples or a big unorganised graph they navigate through units of knowledge and move from one to the other without losing the overview of an entity. An EKP determines a topic context according to its entity type. All relations between resources that emerge from the selected EKP are used as the basis for (i) selecting the information to be aggregated and (ii) visualising it in a concept map fashion. We discuss the first point in next section and the visualisation in Section 3.2.

3.1. Knowledge enrichment and filtering

Knowledge enrichment and filtering is obtained by performing the following steps:

– identity resolution of a subject (provided by a user query) against Linked Data entities;
– selection of the EKP corresponding to the subject type;
– filtering and enrichment of static data about the subject, according to the model provided by the selected EKP;
– filtering and enrichment of dynamic data about the subject, according to the model provided by the selected EKP;
– aggregation of peculiar knowledge about the subject.

In next paragraphs we detail these steps.

Identity resolution. Identity resolution is performed in two possible ways:

– with a semi-automatic approach that leverages a DBpedia index based on the Entityhub component of Apache Stanbol. This approach is semi-automatic because the system returns a list of possible entities that match the subject of a user query, based on the value of their rdfs:label annotation. The user selects an entity within the list.
– with a completely automatic approach based on a Named Entity Resolution (NER) system. Aemoo uses the Enhancer component of Apache Stanbol as NER system. In this case the identity of the subject of a user query is resolved against DBpedia, by choosing the entity with the highest NER confidence.

EKP selection. An EKP provides a unit of knowledge that can be used for answering the question: “What are the most relevant entity types (T₁, ..., Tₙ) that are involved in the description of an entity of type C?”. Thus, given a certain entity, Aemoo needs to (i) identify its type and (ii) select its corresponding EKP, in order to build its entity-centric summary. Given a subject entity, Aemoo replies on an index that associates EKPs with DBPO types. This index is built during the extraction of EKPs from Wikipedia. Selecting an EKP from the index involves the followings:

– identification of the most specific DBPO type t for a subject entity. This allows to avoid multi-typing and to be compliant with the method used for generating type paths (cf. Section 2);
– retrieving the EKP associated with t from the index. If no association is available, Aemoo traverses the DBPO taxonomy of superclasses iteratively until an association is found. If no association is found again the EKP for owl:Thing, i.e., ekp:Thing, is selected. This is the most generic type and its EKP was generated by analysing wikilinks having as subjects those entities with the only type owl:Thing. We remind that in this work we ignore types defined in other ontologies than DBPO (e.g., YAGO).

As an example, the type dbpo:Philosopher would be selected for the entity dbpedia:Immanuel_Kant and the EKP ekp:Philosopher retrieved from the index, accordingly. Fig. 1 shows the UML class diagram for this EKP in which it is possible to see what are the typical relations emerging between entities of this type and other entity types.

Filtering and enrichment of static data. Aemoo uses EKPs as lenses over data for organising and filtering the knowledge to be presented about a specific

---

8The Entityhub is a component of Stanbol that relies on Apache Solr for building a customised entity-centric index of a linked dataset.
9http://ontologydesignpatterns.org/ekp/owl/Thing.owl
10http://ontologydesignpatterns.org/ekp/owl/Philosopher.owl
subject entity. EKPs are used for automatically generate SPARQL CONSTRUCT queries to be executed against DBpedia for retrieving the relevant data. We remark that EKPs evolves over time, and that the results returned by Aemoo will be affected, accordingly. Thus, the filtering and the organisation of data are pattern based and dynamic. For example, the following SPARQL query is generated by using the `ekp:Philosopher`:

```
CONSTRUCT {
    ?entity a ?type .
    ...
}
WHERE{
    GRAPH <dbpedia_page_links_en> {
    }
    GRAPH <dbpedia_instance_types_en> {
        ?entity a ?type .
    }
    GRAPH <ekp_philosopher> {
        ?ekp_property rdfs:range ?type .
    }
}
```

Where, `dbpedia_page_links_en`, `dbpedia_instance_types_en`, and `ekp_philosopher` are three datasets in our SPARQL endpoint[1], i.e., the DBpedia dataset of Wikipedia links, the DBpedia dataset of instance types and the dataset identifying the `ekp:Philosopher` respectively. The query above implements the concept of knowledge boundary in Aemoo that means to filter only the wikilinks that are relevant to a specific subject, using its reference EKP as relevance criterion. The query can be explained as follows:

- the subject entity is associated with its page links at lines 9 and 10;
- each linked entity is bound to its type at line 13;
- linked entities are filtered based on their types, according to the EKP. Based on the definition of type paths [36], we have:

```
* S_i = dbpo:Philosopher
```

where, `?type` is a variable bound to the object type and `?ekp_property` is a variable bound to the EKP local property for a wikilink type (e.g., `ekpS:linksToCity`). This property is generate during the formalisation step of the method described in Section 2. By constraining the results with domain and range of `?ekp_property` (lines 17-20), the result of the query is all wikilinks that match the `ekp:Philosopher` EKP.

These queries generate entity-centric RDF models, used by Aemoo for building a summary description for a specific subject entity. For example, considering `dbpedia:Immanuel_Kant` as subject entity, its resulting entity-centric model is the following:

```
dbpedia:Immanuel_Kant a dbpo:Philosopher ;
  rdfs:label "Immanuel Kant"@en ;
  ekpS:linksToCity dbpo:Königsberg ;
  ...
  ekpS:linksToLanguage dbpo:Latin ;
  ...
  ekpS:linksToBook dbpo:Kritik der reinen Vernunft .
```

Aemoo further enriches the resulting model, by identifying possible semantic relations that could specialise wikilinks. It associates each wikilink type with a list of popular DBpedia semantic relations holding between the types involved in that wikilink type. For example, considering a wikilink type `ekpS:linksToCity`, which links entities of types `dbpo:Person` (subject) and `dbpo:Place` (object), we can extract from DBpedia that the most popular semantic relations linking these type of entities are `dbpo:birthPlace` and `dbpprop:placeOfBirth`. These semantic relations are added to the previous RDF model by means of `skos:relatedMatch` axioms. For example:

```
ekp_phil:linksToCity a owl:ObjectProperty ;
  skos:relatedMatch dbpo:birthPlace, dbpprop:placeOfBirth .
```

15This information is limited by the coverage of DBpedia.

A wikilink is (almost) always embedded in a piece of text, and most of the times this text provides an explanation why the link is there, which in turn provides a linguistic evidence of a semantic relation between the subject entity and the object of the wikilink. Aemoo uses reflects intuition by further enriching the RDF model of a specific subject entity. It extracts, from the Wikipedia page of a subject entity, the text surrounding a specific wikilink (typically a maximum of 50 words), and adds to the model a dedicated OWL2 triple annotations. For example, let us consider the triple

dbpedia:Immanuel_Kant
  ekpS:linksToCity
dbpedia:Königsberg

Its annotation will be the following:

1. [] a owl:Axiom ;
2. owl:annotatedSource
dbpedia:Immanuel_Kant ;
3. owl:annotatedProperty
  ekpS:linksToCity ;
4. owl:annotatedTarget
dbpedia:Königsberg ;
5. grounding:hasLinguisticEvidence: aemoo:wiki-sentence .
6. aemoo:wiki-sentence a doco:Sentence ;
7. frbr:partOf wikipedia:Immanuel_Kant
  c4o:hasContent
  "Immanuel Kant was born in 1724 in Königsberg, Prussia (since 1946 the city of Kaliningrad, Kaliningrad Oblast, Russia), as the fourth of nine children."@en .

Where the triple is annotated with the property
grounding:hasLinguisticEvidence: and the linguistic evidence is identified by aemoo:wiki-sentence.

The linguistic evidence is (i) typed as a doco:Sentence, is (ii) declared to be extracted from Wikipedia with the property frbr:partOf, and (iii) associated with the text by means of the property c4o:hasContent.

Filtering and enrichment of dynamic data. Besides static data sources, Aemoo uses two dynamic data sources (that can be easily extended): Twitter and Google News. In fact, the entity-centric model built by Aemoo can be extended in order to include (i) the current stream of Twitter messages and (ii) the available articles provided by Google News that mention the subject entities co-occurring by mentions of entities that can be resolved on DBpedia. For example, let us consider the following tweet:

"Lots of people love to read Kant here in Rome."

In this tweet, the named entities “Kant” and “Rome” are resolved to the entities dbpedia:Immanuel_Kant and dbpedia:Rome. These, in turn, have types ekp:Philosopher and ekp:City, respectively. Aemoo uses this mechanism for extending an entity-centric model by aggregating new (heterogeneous) data. According to the representation schema previously described, the entity-centric model is then extended in the following way:

1. [] a owl:Axiom ;
2. owl:annotatedSource
dbpedia:Immanuel_Kant ;
3. owl:annotatedProperty
  ekpS:linksToCity ;
4. owl:annotatedTarget
dbpedia:Rome ;
5. grounding:hasLinguisticEvidence
  twitter:tweet_id .
6. twitter:tweet_id a fabio:Tweet
  c4o:hasContent
  "Lots of people love to read Kant here in Rome."@en .

Where the triple is annotated with the property
grounding:hasLinguisticEvidence: and the linguistic evidence, in this case, is identified by twitter:tweet_id.

The tweet is (i) typed as a fabio:Tweet and (ii) associated with the text by means of the property c4o:hasContent.

It is worth clarifying that not all co-occurrences are selected, but only those whose types are compliant with the intensional schema provided by the reference EKP for the subject entity. In fact, let us consider the tweet

"Just bought another Kant’s book on Amazon."

The tweet_id has to be replaced with an actual tweet identifier available from Twitter.

The prefix grounding: identifies the ontology http://ontologydesignpatterns.org/cp/owl/grounding.owl.

The prefix aemoo: identifies the local namespace used by Aemoo.

The prefix doco: identifies the ontology http://purl.org/spar/doco.

The prefix frbr: identifies the ontology http://purl.org/vocab/frbr/core.

The prefix c4o: identifies the ontology http://purl.org/spar/c4o.

16The prefix grounding: identifies the ontology http://ontologydesignpatterns.org/cp/owl/grounding.owl.
17The prefix aemoo: identifies the local namespace used by Aemoo.
18The prefix doco: identifies the ontology http://purl.org/spar/doco.
19The prefix frbr: identifies the ontology http://purl.org/vocab/frbr/core.
20The prefix c4o: identifies the ontology http://purl.org/spar/c4o.
21The prefix grounding: identifies the ontology http://ontologydesignpatterns.org/cp/owl/grounding.owl.
22The prefix aemoo: identifies the local namespace used by Aemoo.
23The prefix doco: identifies the ontology http://purl.org/spar/doco.
24The prefix frbr: identifies the ontology http://purl.org/vocab/frbr/core.
25The prefix c4o: identifies the ontology http://purl.org/spar/c4o.
The entity `dbpedia:Amazon` typed as `dbpo:Organisation` would not be added to the entity-centric model associated with `dbpedia:Immanuel_Kant`, because, according to the relevance criterion provided by the EKP `ekp:Philosopher`, the relation between `dbpo:Philosopher` entities and `dbpo:Organisation` entities are not to be included.

**Aggregation of peculiar knowledge.** However, the link between `dbpedia:Immanuel_Kant` and `dbpedia:Amazon` is kept and stored as a peculiar information for this specific subject. We refer to this kind of data as “curiosities” about a subject, and make the user able to visualise them in a complementary entity-centric model (cf. Section 3.2). Curiosities are captured by the wikilinks of a subject whose path type falls in the long-tail, i.e. its `pathPopularity` value is below the EKP threshold (cf. Definition 2). While EKPs allow to capture the core relevant knowledge about a subject based on the relations that are popularly used for all (or most) entities of its same type, curiosities show features that are peculiar to that subject, hence providing a different relevance criterion for selection, that is what makes it a distinguished entity as compared to the others of its same type. Let us consider as an example two subjects: “Barack Obama” and “Arnold Schwarzenegger”. Both are office holders according to DBpedia, hence their entity-centric model based on the relevance criterion of identifying the core knowledge about them will be built according to the `dbpo:OfficeHolder` EKP, which will allow Aemoo to show information about legislatures, elections, other office holders, etc. If for the same subjects we want to identify information that distinguish them from each other, the retrieved data that complement the core knowledge, will include useful insights. Aemoo reflects this intuition by generating a complementary entity-centric model for the subjects. This model will include the information that Barack Obama was awarded with a nobel prize, and a list of movies that starred Arnold Schwarzenegger. However, differently from the core knowledge filtered by EKP, these data are noisy, hence further studies include the investigation of this complementary models to the aim of conceiving strategies for refining this “peculiarity-based” relevance criterion.

### 3.2. Knowledge visualisation

In this section we motivate our design choices concerning the visualisation and the presentation of data in Aemoo.

Dadzie and Rowe [13], starting from the conclusion of Shneiderman [46], debate about the requirements that visualisation systems should fulfill for enabling the consumption of Linked Data depending on the target user, i.e., a **tech-user** (a user with a good knowledge of Linked Data and Semantic Web technologies) or a **lay-user** (a user with little knowledge of Linked Data and Semantic Web technologies). We focused on the design of Aemoo user interface (UI) for providing a visual presentation of data in order to enable its usage to lay-users. Hence, we took into account the following requirements:

- **(R1)** data extraction and aggregation in order to provide overviews of the underlying data;
- **(R2)** intuitive navigation;
- **(R3)** support for filtering to users with no knowledge of formal query syntax, data structure and specific vocabularies used in the dataset;
- **(R4)** exploratory knowledge discovery;
- **(R5)** history mechanism for enabling undo or re-tracing of navigation steps;
- **(R6)** support to detail data on demand.

Requirements R1 and R3 are fulfilled by applying the method described in Section 3.1. Then, the main issue is to provide an intuitive visualisation (R2) that benefits from the way Aemoo extracts, aggregates and filters data for enabling exploratory search (R4). Designing an UI for supporting humans in exploratory search means to reduce the cognitive load required for activities like look-up, learning and investigation [28]. Novak and Gowin [35] demonstrated that **concept maps** are an effective means for representing and communicating knowledge for supporting humans in undertaking understanding and learning tasks. Concept maps, introduced by [33], are graphical diagrams that are typically used to depict relationships between concepts and they can be easily adopted in our case for visualising knowledge organised by means of EKPs. In fact EKPs, similarly to concept maps, include concepts (i.e., DBPO classes) and relations among them. Thus, we claim that visualising knowledge in a concept map is a fair solution to help humans to construct their **mental models** [12] for addressing exploratory search tasks with Aemoo. The usage of concept maps in Aemoo addresses requirement R2, i.e enabling an intuitive navigation mechanism in the UI.

The concept map related to a subject entity is visualised by Aemoo in a radial way having:

- the subject entity in the center represented as squared node;
- circular nodes around the subject entity that represent sets of resources of a certain type. Namely, they are the types of the resources linked to the subject entity that Aemoo aggregated and enriched from the various sources (cf. requirement R1). We refer to them as node sets. As previously described, such types are the ones that a user would intuitively expect to see in a summary description of an entity according to its type, as we described in Section 2.
The radial graph with one-degree connections was preferred to other solutions (e.g., tabular, tree, etc.) because it mirrors the structure of an EKP (cf. Figure 1). Additionally, the radial graph, referred as a *bona fide* metaphor for presenting data by some authors [15], is meant to be used by users as a “compass” for exploring the knowledge about a subject entity, which is always displayed in its center. The usage of the radial concept map as a navigation compass is designed to address requirement R6, i.e. to detail data on demand. In fact the user can over with the pointer on the circular nodes of the graph in order to expand its knowledge and gather more and more detailed information. The radial graph is interactive: all its elements can be clicked, including the new interface objects that are trigger by navigating it, causing the user to change and enrich the focus of the exploration. The interaction aspect allows us to address requirement R4, i.e. exploratory knowledge discovery.

Figure 2(a) shows an example of a concept map having Immanuel Kant as subject entity. Aemoo splits the interface into two parts: (i) in the center-right side it visualises the concept map and (ii) in the left side it visualises a widget named *entity abstract*. An entity abstract provides a high-level overview on selected entities fulfilling requirement R1, along with the concept map. This overview consists of an entity label with its thumbnail, DBPO type and abstract. A user can click on an entity label or on its DBPO type in order to be redirected to their corresponding pages in DBpedia. Similarly, a user can open the Wikipedia page associated with the subject entity by clicking on the link “(go to Wikipedia page)” that Aemoo shows at the end of the abstract (cf. Figure 2(a)).

The concept map is interactive and serves, as previously argued, as an entity-centric navigation tool for gathering more detailed information and browsing among DBpedia entities. More in detail, the following elements of the interactive interface address requirements R3, R4 and R6:

- a square box (cf. the arrow with id 1 in Fig. 2(b)) is visualised by hovering the pointer on any node set. Such a box contains the list of the entities belonging to a node set. The list is paginated in order to present 10 entities per page. We refer to these boxes as *entity boxes*. Each entity in an entity box is depicted along with an icon that indicates the provenance (i.e., Wikipedia, Twitter or Google News). These icons are also shown for node sets in order to summarise the provenance about their contained entities. In an early prototype of Aemoo, the content of the entity boxes was visualised by exploding the node sets as new circular nodes. We ran a first set of preliminary tests by asking volunteer users to play with Aemoo and provide open feedback on its early interface prototype. Most of the feedback suggested
minor changes (e.g. on the positioning of the abstract and explanations), which we implemented. Nevertheless, a main major request emerged from most users that was to keep the graph to visualise only one-degree relations and to move elsewhere the content of the sets in order to keep the interface content readable and easier to navigate. In addition to this empirical observation we claim that this choice is also motivated by the inherent difference between the relation between the subject entity and the node sets, and the relation between the node sets and the entities that they contain. This is why Aemoo now separates entity boxes from the concept map, which keeps the interface easier to navigate and reflects the different semantics of the visualised relations.

- the relations between the subject entities and the surrounding node sets are always displayed by using the label “links to” representing a set of wikilink relations. Aemoo also shows a list of semantic relations extracted from DBpedia that might express the intension of the “links to” relation. This list is displayed in a tooltip appearing by mouse hovering the pointer on any “links to” label (cf. Section 3.1). Such relations are (i) represented as \texttt{skos:relatedMatch} axioms in the entity-centric model generated by Aemoo for the subject entity (cf. Section 3.1), and (ii) are extracted according to their frequencies for the specific path type. The tooltip is used in order to enable the detail-on-demand functionality (cf. requirement R6). For example, Fig. 2(a) shows a tooltip for the “links to” relation between Immanuel Kant and the node set City. Such a tooltip provides dbpo:birthPlace and dbpprop:placeOfBirth as possible semantic relations for that type of link;

- a list of explanations appears in a widget at the left-bottom of the interface (cf. the arrow with id 2 in Figure 2(b)) by hovering on any entity in an entity box. These explanations provide details-on-demand (cf. requirement R6) and come from the Wikipedia text surrounding the wikilink, as described in Section 3.1. The explanations can be used by humans for understanding the semantics of the relations between the entities visualised by the concept map. For example, in Figure 2(b) are shown all the linguistic evidences that explain the relations between Immanuel Kant and Königsberg. A user can easily get that the link represents two relations: Immanuel Kant was born in Königsberg, and there is a statue of Kant in that city. Explanations have also associated icons, which indicate their provenance;

- any entity in an entity box is clickable. This enables navigation and detail-on-demand capabilities (cf. requirements R2 and R6). In fact, a user can change the focus (i.e., the subject entity) of the concept map at any time by selecting with a click any possible entity in an entity box. When the focus changes the concept map is rearranged according to the new subject entity and its type by applying the appropriate EKP. Figure 3 shows the situation after some exploration steps that changed the focus to Prussia as subject entity. At the center-bottom of the interface there is the exploratory history (cf. the arrow with id 4 in Figure 3), named breadcrumb. The breadcrumb fulfills both requirements R4 and R5 as it allows a user to retrace his exploratory steps at any time and provides her with updated information about her exploratory path.

The sources to be used for populating the concept map can be chosen by users through a set of checkboxes that appear at the top-right corner of the interface (cf. Figure 2(b)). A blue link located in top-center of the interface under the search bar allows users to switch to the “curiosities” about a subject entity. When clicking on this link the knowledge is again arranged in a concept map fashion, and enriched with news and tweets just as it happens for the previous summary, but this time the node sets are selected with a different criterion: they are types of resources that are unusual to be included in the description of a country, hence possibly providing insights of what distinguishes e.g. Prussia, from other countries (cf. Section 3.1). We use the same visualisation metaphor as for the presenting the core knowledge about a subject entities in order to keep the interface coherent and ensure a smooth interaction between the two views.

Table 1 provides a summary of Aemoo UI components indicating the requirements they contribute to address.

\footnotetext{The prefix \texttt{dbpprop} stands for the namespace \url{http://dbpedia.org/property/}}
Aemoo is released as a RESTful architecture: it consists of a server side component implemented as a Java-based REST service, and a client side component based on HTML and JavaScript.

The overview of the architecture of the server side, including also the components for EKP extraction, is depicted in Figure 4.

The architecture is designed by using the Component-based [21] and the REST [16] architectural styles.

3.3. Implementation details

Aemoo is composed of the following components:

- KP Extraction Coordinator which takes care about the coordination of the overall extraction process;
- Property Path Identifier that is responsible for the identification of type paths (cf. Definition 1);
- Property Path Storage that manages the storage of identified paths;
- Property Path Analyzer that draws boundaries around paths in order to formalise KPs;
- KP Repository Manager which is responsible for the storage, indexing and fetching of KPs.

Table 1
UI components with respect to design requirements.

<table>
<thead>
<tr>
<th>UI component</th>
<th>R1</th>
<th>R2</th>
<th>R3</th>
<th>R4</th>
<th>R5</th>
<th>R6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Concept map</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Entity abstract</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Entity box</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Explanation widget</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Breadcrumb</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

The Knowledge Pattern (KP) extractor is composed of the following components:

- KP Extraction Coordinator which takes care about the coordination of the overall extraction process;
4. Evaluation

In Section 4.1 we hypothesised the (i) cognitive soundness of EKPs for providing intuitive entity-centric summaries. We also hypothesised that (ii) EKPs can be exploited for visualising Linked Data in order to help humans in exploratory search tasks. In this section we describe the experimental setup used for assessing the validity of our first hypothesis, while in Section 4.2 we describe the experimental setup used for assessing the validity of the second one.

4.1. Experimental setup for evaluating EKPs

In order to assess our first hypothesis we carried out a user-based study aimed at making emerge EKPs from human consensus and to compare them with those empirically emerging from DBpedia. We asked 17 participants to indicate the core relevant types of things (object types) that could be used to describe a certain type of things (subject types). For example, for the subject “Country” (such as Germany), core object types can be “Language” (e.g. German), “Country” i.e. other countries with which it borders (e.g. Denmark, Poland, Austria, Czech Republic, Switzerland, France, Luxembourg, Belgium, and Netherlands), etc.

The participants had different nationalities (Italy, Germany, France, Japan, Serbia, Sweden, Tunisia, and Netherlands), and different mother tongues, although they were all fluent in English. Having participants from different nationalities and native languages allowed us to observe whether EKPs are perceived as sound units of meaning independently from one specific language or culture, at least for those represented in our study. Although this cannot be assessed with proof, a successful result such as e.g. high inter-rater agreement, would provide an encouragement towards further dedicated study.

<table>
<thead>
<tr>
<th>DBPO class type</th>
<th>nRes(S)</th>
<th>nPath(Si)</th>
<th>AvPathOcc(Si)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Language</td>
<td>3,246</td>
<td>99</td>
<td>29.27</td>
</tr>
<tr>
<td>Philosopher</td>
<td>1,009</td>
<td>112</td>
<td>18.29</td>
</tr>
<tr>
<td>Winter</td>
<td>10,102</td>
<td>172</td>
<td>15.30</td>
</tr>
<tr>
<td>Ambassador</td>
<td>286</td>
<td>85</td>
<td>15.58</td>
</tr>
<tr>
<td>Legislature</td>
<td>485</td>
<td>83</td>
<td>25.11</td>
</tr>
<tr>
<td>Album</td>
<td>99,047</td>
<td>172</td>
<td>17.71</td>
</tr>
<tr>
<td>Radio Station</td>
<td>16,310</td>
<td>151</td>
<td>7.31</td>
</tr>
<tr>
<td>Administrative Region</td>
<td>31,386</td>
<td>185</td>
<td>11.30</td>
</tr>
<tr>
<td>Country</td>
<td>2,234</td>
<td>169</td>
<td>35.16</td>
</tr>
<tr>
<td>Insect</td>
<td>37,742</td>
<td>98</td>
<td>9.16</td>
</tr>
<tr>
<td>Disease</td>
<td>5,215</td>
<td>153</td>
<td>12.10</td>
</tr>
<tr>
<td>Aircraft</td>
<td>6,420</td>
<td>126</td>
<td>10.32</td>
</tr>
</tbody>
</table>

We provided participants with a list of 12 subject types (with examples, textual labels, and abstracts from DBpedia) spanning different domains. These types are presented in Table 2 accompanied with the number of resources of that type in DBpedia (nRes(S)), the number of paths they participate in as subject types (nPath(Si)), and the average number of occurrences of their associated paths (AvPathOcc(Si)). The information about nRes(S), nPath(Si) and AvPathOcc(Si) was not given to participants.

Then, we provided participants with a list of object types (with examples, textual labels and abstracts from DBpedia) for each subject type in the list. This second list was obtained by selecting, for each subject type Si, the object types Oj with pathPopularity(\(P_{i,j}, S_i\) \(>= 18.18\), where 18.18 is the threshold value t (cf. Section 2) and a random sample of object types with pathPopularity(\(P_{i,j}, S_i\) \(< 18.18\).

The participants were asked to assign a score on a 5-point scale (from 1: Irrelevant to 5: Relevant) to each pair subject type/object type, indicating how core and relevant a certain object type was for building a de-
cription of the subject type. We defined a mapping between pathPopularity values (ranging from 0 to 100) and the 5-point scale rates that users provided. The mapping (cf. Table 3) allowed us to compare ranking values provided by humans and pathPopularity_{DBpedia} ranges.

Table 3

<table>
<thead>
<tr>
<th>pathPopularity_{DBpedia} interval</th>
<th>Relevance score</th>
</tr>
</thead>
<tbody>
<tr>
<td>[18, 100]</td>
<td>5</td>
</tr>
<tr>
<td>[11, 18]</td>
<td>4</td>
</tr>
<tr>
<td>[2, 11]</td>
<td>3</td>
</tr>
<tr>
<td>[1, 2]</td>
<td>2</td>
</tr>
<tr>
<td>[0, 1]</td>
<td>1</td>
</tr>
</tbody>
</table>

In Section 5.1 we provide the outcomes of this experimental setup.

4.2. Experimental setup for evaluating Aemoo

In order to assess the validity of our second hypothesis (i.e., the visualisation based on EKPs can support humans during exploratory search tasks) we carried out a user-based study whose aim was twofold:

- evaluating the system usability of Aemoo;
- analysing users’ feedback about their interaction with the UI of Aemoo.

For this purpose, we defined three tasks involving look-up, learning and investigation [28], phases that characterise the strategies that humans adopt while exploring the Web. Each task could be undertaken by using one of three tools: Google, RelFinder, and Aemoo. The tool to be used was automatically selected by a system built for the evaluation, hence it was not a choice of the participants but constraint by the experimental setting. Using three tools allowed us to conduct the evaluation of Aemoo as a comparative analysis. Google and RelFinder provided us two viable and suitable choices to compare with: (i) Although it does not provide a specially designed interface for exploratory search, Google is de facto the most used tool currently used for this task on the Web. Users have developed their own methods for exploring and discovering knowledge by using Google, and they are very familiar with its interface. We expected that comparing with Google would give us insights on how Aemoo is perceived as compared to a popular and well known (exploratory) search interface. For this reason, Google provided us with a reference baseline; (ii) RelFinder [13] is a tool supporting visual exploratory search, it is very popular among Semantic Web experts, less known to the general users. It uses a graph visualisation metaphor and gives users the possibility to filter data according to a number of fixed criteria. Comparing with RelFinder allows us to assess both the usability of our visualisation interface with respect to RelFinder’s, and the effectiveness of the EKP-based relevance criterion for automatic summarisation as compared to manual filtering.

It has to be noticed that the three tools rely on different background data sources (i.e. Google on potentially the whole Web, RelFinder on a number of linked datasets, and Aemoo on DBpedia, Twitter and Google News). This would cause an issue during the analysis of results as the data gathered by users from the different tools would not be straightforwardly comparable: one may not be able to judge whether a difference in the task solutions is due to a more effective exploration support or to a larger/smaller set of available data sources. Based on this rationale, in order to make the results comparable, we constrained the tools background knowledge to Wikipedia (for Google) and to DBpedia (for RelFinder), which represent the intersection of the the three tools data sources.

The three tasks of the controlled experiment are the followings:

Task 1 - Summarisation: the participants were asked to build a summary of the subject “Alan Turing” with the information they could get from the tool at hand. They were asked (i) to collect as quickly as possible all elements and their relations that could be possibly included in such summary and record them, (ii) and to score them based on their relevance and unexpectedness. The information had to recorded as a set of triples <subject, objects, description>. For example, if “Paris” was the subject, the information that Paris is located at the heart of the Île-de-France region would be recorded as follows:

- subject=Paris
- object=Île-de-France
- description=Paris is located at the heart of the Île-de-France region

For each identified triple, the user had to separately rank its relevance and unexpectedness with respect to the subject, using a 5-point scale ranging from 1 (Irrelevant/banal) to 5 (Relevant/unexpected). Participants
were free to start their exploration from any concept, but they were constrained on the tool to use for exploring information, and on the maximum time (10 minutes) they could take for each task. They were instructed to include all possible triples they could get from the tool results, being them interesting, wrong, relevant, obvious, etc., and rank them accordingly. They were invited to act such as the finalisation of the summary would happen at a later stage, based on their ratings.

Task 2 - Related entities: the participants were asked to find as many objects of a certain type as possible, having a relation to the given subject. The actual question was: What are the places related to “Snow White”? Hence, the participants had to provide a list of all places related to “Snow White”, with the help of the provided tool. Examples are: Germany, Albania, Russia, etc.

Task 3 - Relation finding: the participants were asked to find one or more relations between two subjects and to describe its semantics. The actual question was: What is the relation between “Snow White” and “Charlize Theron”? Participants had to report the list of relations they were able to find, with the help of the provided tool. Possible elements of such list could be: acts in the movie, plays role in the movie, etc.

Each group performed the three tasks (on the same subjects) twice using two different tools\(^2\)\(^5\), one of which was always Aemoo. On one hand, performing the same task twice might introduce a bias as during the second iteration the cognitive load could be reduced due to the previous investigation of the same subject. On the other hand, following this procedure allowed us to get feedback on participants’ perception of the tools comparison. In order to mitigate the bias possibly affecting the results, the experiment was designed in order to balance the number of participants using Aemoo as first tool with the number of participants using an alternative system (either Google or RelFinder [24]) as first tool. We recorded the experiment executions by keeping track of the ordering, hence we were able to split the results collected from the first run of the tasks from those collected from the second run.

At the end of each iteration, the participants were asked to rate ten statements using a five-point Likert scale (from 1: Strongly Disagree to 5: Strongly Agree) and to answer five open questions. Table 4 shows the ten statements, which are based on the System Usability Scale (SUS) [8]. The SUS is a well-known metrics used for evaluating the usability of a system. It has the advantage of being technology independent and to be reliable even with a very small sample size [45]. It also provides a two-factor orthogonal structure, which can be used to score the scale on independent Usability and Learnability dimensions [45].

Table 4

<table>
<thead>
<tr>
<th>Nr.</th>
<th>Question</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>I think that I would like to use the system frequently</td>
</tr>
<tr>
<td>2</td>
<td>I found the system unnecessarily complex</td>
</tr>
<tr>
<td>3</td>
<td>I thought the system was easy to use</td>
</tr>
<tr>
<td>4</td>
<td>I think that I would need the support of a technical person to be able to use the system</td>
</tr>
<tr>
<td>5</td>
<td>I found the various functions in the system were well integrated</td>
</tr>
<tr>
<td>6</td>
<td>I thought there was too much inconsistency in the system</td>
</tr>
<tr>
<td>7</td>
<td>I would imagine that most people would learn to use the system very quickly</td>
</tr>
<tr>
<td>8</td>
<td>I found the system very awkward to use</td>
</tr>
<tr>
<td>9</td>
<td>I felt very confident using the system</td>
</tr>
<tr>
<td>10</td>
<td>I needed to learn a lot of things before I could get going with the system</td>
</tr>
</tbody>
</table>

Table 5 reports the five open questions aimed at collecting feedback (pros and cons) from the participants about the quality of their experience with Aemoo. We used users feedback to this questionnaire for performing a qualitative analysis of Aemoo based on Grounded theory [47]: a method often used in Social Science to extract relevant concepts from unstructured corpora of natural language resources (e.g., texts, interviews, or questionnaires).

We developed an ad-hoc web tool named AemooEval [50] for supporting the experiment. The tool was designed in order to allow users to perform their tasks with any of the tools (i.e. Aemoo, Google, RelFinder) without the need of switching context and interface for recording their findings and providing their judgments. Figure 5 shows the interface of the tool during the

\(^2\)The first iteration consisting of the three tasks performed with one tool, and the second iteration consisting of the same three tasks performed with a second tool.

\(^5\)http://wit.istc.cnr.it/sweng/cgi-bin/
execution of Task 1 (i.e. Summarisation). The “Show more” button at the top of the page allows to view/hide a text providing guidelines for undertaking the task. The main body of the interface embeds the tool to be used (Aemoo in the Figure), while the bottom of the interface provides text fields and checkboxes for collecting input from users e.g. the triple <subject, object, description> for Task 1.

For Task 2 and Task 3, the bottom part of the interface only includes text fields as no ranking is requested. AemooEval takes care of managing the task iterations, automatically selecting the tool to be used at a certain iteration by guaranteeing the balance of the alternating sequence of tool usage, storing user feedback and metadata (user id, iteration, time to perform the task, etc.), and enforcing the time constraint.

The three tasks were performed by 32 participants aged between 20 and 35 years old and equally distributed in terms of gender. All the participants were undergraduate students in computer science coming from two different universities, i.e., the University of Bologna in Italy and the University of Paris 13 in France. The participants were divided into 5 groups and supervised by an evaluator, who was in charge to support them during the experiments. The evaluator provided participants with an introductory description of the experiment’s goal and tasks, a brief tutorial

Fig. 5. The interface of the Web-based tool designed and implemented for carrying on the user-study.
Table 5
Open questions for grounded theory-based analysis.

<table>
<thead>
<tr>
<th>Nr.</th>
<th>Question</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>How effectively did the system support you in answering to the previous tasks?</td>
</tr>
<tr>
<td>2</td>
<td>What were the most useful features of the system that helped you to perform your tasks?</td>
</tr>
<tr>
<td>3</td>
<td>What were the main weaknesses that the system exhibited in supporting your tasks?</td>
</tr>
<tr>
<td>4</td>
<td>Would you suggest any additional features that would have helped you to accomplish your tasks?</td>
</tr>
<tr>
<td>5</td>
<td>Did you (need to) open and read Wikipedia pages when using the system? If yes, please explain the motivation.</td>
</tr>
</tbody>
</table>

Table 6
Self-assessment questionnaire

<table>
<thead>
<tr>
<th>Nr.</th>
<th>Question</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>I have extensive experience in exploratory search</td>
</tr>
<tr>
<td>2</td>
<td>I am an expert user of Aemoo</td>
</tr>
<tr>
<td>3</td>
<td>I am an expert user of RelFinder</td>
</tr>
<tr>
<td>4</td>
<td>I am an expert user of Google</td>
</tr>
<tr>
<td>5</td>
<td>I frequently use the Web to explore information and to perform tasks such as homework, presentations, working analysis, reports</td>
</tr>
<tr>
<td>6</td>
<td>I have extensive experience in Semantic Web and Linked Data technologies</td>
</tr>
<tr>
<td>7</td>
<td>I have detailed knowledge of Alan Turing</td>
</tr>
<tr>
<td>8</td>
<td>I have detailed knowledge of Snow White</td>
</tr>
<tr>
<td>9</td>
<td>I know what Wikipedia is</td>
</tr>
<tr>
<td>10</td>
<td>I frequently use Wikipedia</td>
</tr>
<tr>
<td>11</td>
<td>I know what DBpedia is</td>
</tr>
<tr>
<td>12</td>
<td>I frequently use DBpedia</td>
</tr>
</tbody>
</table>

In order to assess the background and skills of participants, before running the experiments they were asked to rank 27 twelve statements, reported in Table 6, composing a self-assessment questionnaire. Figure 6 shows the averages of scores for each statement and their standard deviations. According to the results of the self-assessment questionnaire, participants confirmed to be lay-users as for what concerns their previous knowledge of Semantic Web, Linked data technologies, and DBpedia (statement 6, 11, 12). They declared to have small experience of exploratory search (statement 1), and to have no knowledge about RelFinder and Aemoo (statements 2, 3), though as expected, they were familiar with Google (statement 4).

Furthermore, the knowledge about the two subjects explored during the execution of the tasks was comparable among the participants (statement 7, 8) as it can be observed from the small standard deviation value (0.18 and 0.16, respectively).

The next section shows and discusses the results of the illustrated experiments.

5. Results and discussion

5.1. Cognitive soundness of EKPs

Table 7 reports the results computed for the inter-rater agreement and the reliability score referred to the first experiment, i.e. emerging EKPs from human consensus (cf. Section 4.1). For computing the inter-rater agreement we used Kendall’s coefficient of concordance (W). Kendall’s W computes the agreement among raters on a scale ranging from 0 (Agreement) to 1 (Disagreement). For assessing the reliability of the raters, we computed the Cronbach’s coefficient of internal consistency (α). Cronbach’s α is considered unacceptable for α < 0.5 and excellent for α ≥ 0.9. On average the inter-rater agreement resulted very good, with W = 0.68, and the reliability of the raters was excellent, with an average α = 0.93.

In order to compare the EKPs annotated by humans with the EKPs empirically extracted from Wikipedia,
we computed the correlation between the scores assigned by participants, and a ranking function named $path\text{Popularity}_{DBpedia}$, representing the trend of $path\text{Popularity}$ values for Wikipedia EKPs. Details on how $path\text{Popularity}_{DBpedia}$ was obtained can be found in [36]. We used Spearman’s $\rho$ for measuring correlation, which can have values in a range [-1 (Disagreement), 1 (Agreement)]. Table 8 shows that we obtained high correlation values (average $\rho = 0.75$).

Table 8

<table>
<thead>
<tr>
<th>DBPO class</th>
<th>Correl. users / DBpedia</th>
</tr>
</thead>
<tbody>
<tr>
<td>Language</td>
<td>0.893</td>
</tr>
<tr>
<td>Philosopher</td>
<td>0.661</td>
</tr>
<tr>
<td>Writer</td>
<td>0.748</td>
</tr>
<tr>
<td>Ambassador</td>
<td>0.655</td>
</tr>
<tr>
<td>Legislature</td>
<td>0.716</td>
</tr>
<tr>
<td>Album</td>
<td>0.871</td>
</tr>
<tr>
<td>Radio Station</td>
<td>0.772</td>
</tr>
<tr>
<td>Administrative Region</td>
<td>0.874</td>
</tr>
<tr>
<td>Country</td>
<td>0.665</td>
</tr>
<tr>
<td>Insect</td>
<td>0.624</td>
</tr>
<tr>
<td>Disease</td>
<td>0.824</td>
</tr>
<tr>
<td>Aircraft</td>
<td>0.664</td>
</tr>
</tbody>
</table>

**Discussion**  
The good inter-rater agreement and the high reliability of participants allows us to claim that the 12 analysed EKPs constitute a reliable sample for our investigation. The high correlation between participants’ scores and $path\text{Popularity}_{DBpedia}$ shows that (i) the identified value for $t$ [36] (cf. Section 2) is a stable criterion for determining knowledge boundary for EKPs; and that (ii) the extracted EKPs can serve as criterion for building summaries about an entity, based on its type. The multi-cultural and multi-language character of the participants, together with the high values of agreement, encourage a line of thought that hypothesises EKPs to be cognitively sound units of meaning, regardless the specificity of language and culture. However, it has to be remarked that participants were all from western countries, highly educated and fluent in English, hence this hypothesis needs dedicated and further investigation to be validated.

5.2. Usability of Aemoo

In order to compare the performance of participants in executing their tasks with the support of the three tools (i.e. Aemoo, RelFinder, and Google), we consider the time spent by participants for providing each answer (e.g. each triple in Task 1). Our intuition is that this measure can give us an insight on how well the tools support users in undertaking their tasks, especially when there is significant difference among the three tools. The observed performance is reported in Figure 7. On average, Aemoo performs better than RelFinder and Google, however this is due to Task 2 for which it significantly outperforms the other tools, while RelFinder has a slight better performance in Task 1, and Google in Task 3.

A main focus of our experiment is to assess how Aemoo is perceived by participants in terms of its usability. To this aim we computed the SUS for the three tools (cf. Table 4), and show the results in Figure 8(b).
We distinguish the results of the first iterations (only considering the questionnaire filled after performing the tasks the first time) from the results of only the second iterations (only considering the questionnaire filled after the second iteration). We also report the aggregated results of the two iterations. Values between brackets provide standard deviations and they are also reported into the chart as vertical black bars. SUS values range between 0 (Unusable) and 100 (Usable). Based on empirical studies [45], a SUS score of 68 represents the average usability value for a system. The same work demonstrates that SUS allows to reliably assess the usability of a system also with a small number of participants. Aemoo usability is satisfactory (av-
Figure 8 shows the p-values, computed by using the Tukey’s HSD (honest significant difference) method, indicating the statistical significance of the pairwise comparison among the three tools. Unfortunately, the evidence is insufficient for claiming the significance of the comparison between Google and Aemoo, however the data are reported for completeness and for possible use in future work. We have strong evidence ($p < 0.01$) supporting the hypothesis that Aemoo is more usable than RelFinder (also RelFinder results less usable than Google), if we consider both iterations. This claim is also supported by moderate evidence if we consider the iterations separately ($0.01 < p \leq 0.05$).

In addition to the SUS score, we analysed two subparameters (Learnability and Usability [27]) for the three systems. The value range for these parameter is [0 (Hard to learn/use), 1 (Easy to learn/use)]. The results of this analysis are reported in Figure 9. Figure 9(a) shows the learnability scores for the three systems and their standard deviations. According to Figure 9(b), the evidence is insufficient for claiming the significance of the comparison, however we report the results for completeness and for possible use in future investigation. As far as Usability is concerned, Figure 9(c) shows the obtained scores (with their standard deviations). We have strong evidence for claiming that Aemoo is more usable than RelFinder (moderate evidence support the same claim if we consider the two iterations, separately).

The results of the open questionnaire (cf. Table 5) were analysed by using the Grounded Theory [47]. We proceeded first with the open coding and then with the axial coding. The open coding aims at extracting actual relevant sentences - called codes - from the answers. The axial coding rephrases the original codes so as to define conceptual clusters capturing semantic connections from codes. Each conceptual cluster was associated with a priority score (the greater the amount of codes feeding a conceptual cluster, the highest its priority) in order to identify the most important issues arising from participants’ feedback. Figure 10 shows the results of the open questionnaire, limited to the most frequently mentioned codes (ordinate values report the number of mentions normalised on a scale between 0 and 1).

Finally, Figure 5 shows the ratings (based on a Likert scale) that participants provided for judging relevance and unexpectedness of the results provided by the three systems during Task 1 (cf. Section 4.2), considering both iterations. Aemoo ($r = 4.11, u = 2.92$) performed slightly better than RelFinder ($r = 3.97, u = 2.64$) and Google ($r = 4.05, u = 2.2$). Table 9 shows the results by separating the values for the first and the second iterations, respectively, which confirm the previous ones with the only difference that Aemoo and RelFinder have the same results for relevance ratings in the first iteration.

Table 9

<table>
<thead>
<tr>
<th>Tool</th>
<th>$r_1$</th>
<th>$u_1$</th>
<th>$r_2$</th>
<th>$u_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aemoo</td>
<td>3.8</td>
<td>2.82</td>
<td>4.42</td>
<td>3.02</td>
</tr>
<tr>
<td>RelFinder</td>
<td>3.81</td>
<td>2.53</td>
<td>4.13</td>
<td>2.75</td>
</tr>
<tr>
<td>Google</td>
<td>4</td>
<td>2.18</td>
<td>4.1</td>
<td>2.22</td>
</tr>
</tbody>
</table>

Discussion. In information retrieval, accuracy is the typical measure used to evaluate the performance of a system. The higher the precision and recall, the more the system is accurate. Unfortunately, exploratory search tasks are typically associated with undefined and uncertain goals [50]. For example, there are a myriad of correct answers that a user can find for providing a summary about the topic “Alan Turing” (cf. Task 1). It is near impossible to classify all correct and wrong answers in order to use precision and recall, appropriately. Additionally, we are not interested in the accuracy of results as much as we are in evaluating the data visualisation (UI usability) and filtering capabilities (EKPs) of Aemoo.

For this reason our evaluation focused on (i) the analysis of the time required by participants for completing their tasks, (ii) the SUS, (iii) a grounded analysis of open user feedback and (iv) user ratings about relevance and unexpectedness of the results provided by the tools.

The time needed by participants for completing the experiment tasks might be biased by a variety of factors that include, for example, users’ expertise about a certain topic or their familiarity with the system. However, in our experiment participants resulted to be lay-users, unfamiliar with the tools (with the exception of

28$r$ is stands for relevance, “$u$” stands for unexpectedness. Subscripts indicate the iteration.
(a) Learnability scores.

(b) \(P\)-values computed with a Tukey’s HSD pairwise comparison of the learnability scores between Aemoo/RelFinder, Aemoo/Google and Google/RelFinder.

(c) Usability scores.

(d) \(P\)-values computed with a Tukey’s HSD pairwise comparison of the usability scores between Aemoo/RelFinder, Aemoo/Google and Google/RelFinder.

Fig. 9. Learnability (cf. Figure 9(a)) and Usability (cf. Figure 9(c)) values with corresponding \(p\)-values (cf. Figures 9(b) and 9(d)) measuring significance. Standard deviation values are expressed between brackets and shown as black vertical lines in the chart.
Google), and unfamiliar with the subjects of the tasks (cf. Figure 6). Hence the observed performances give us a reasonably reliable insight about the effectiveness of the different tools in supporting exploratory search tasks.

As far as UI usability is concerned, the SUS-based analysis provided us with a good overview of the system performance. Aemoo can be considered averagely usable, which is a satisfactory result especially considering the comparison with RelFinder usability. The fact that Aemoo outperformed RelFinder suggests that

![Fig. 10. A chart of the most mentioned pros and cons in the open questionnaires.](image1)

![Fig. 11. Relevance and unexpectedness ratings (and their standard deviations between brackets) according to participants' feedback.](image2)

the design choices implemented in Aemoo successfully address the reference requirements described in Section 3.2 (cf. Table 1) based on [13]. Also, the results of the task-driven experiments enforce the findings related to EKP cognitive soundness, i.e. EKPs provide an effective filtering criteria for building automatic entity summarisations. However, the same results point out the need of further improving Aemoo including smart mechanisms for supporting exploratory browsing, especially for identifying relations between two or more entities.
We acknowledge significant space for improvements and, in this respect, the grounded analysis (based on open questionnaire) provides us with insights on what are the most critical issues to be addressed. In more details, the main “cons” reported by users and that we will consider in future development of the system include the lack of a mechanisms to perform comparison among different entities, and the lack of a mechanisms for supporting temporary storage (e.g., a basket).

In addition to these, we think that the automatic relation finding (provided by RelFinder), if appropriately integrated in the interface, can provide additional value to Aemoo results e.g. by indicating when a specific semantic relation is known between two entities. Some participants appreciated the facet-based filtering provided by RelFinder and a comparable number of them judged it as awkward. This is probably due to the potentially huge number of relations between entities that can be proposed as filtering options to the users. In our opinion, this issue is mainly due to scalability problems that, if addressed appropriately, can turn this functionality to an added value to the data visualisation. Based on this observation, we plan to investigate the trade-off between the automatic (e.g., based on EKPs in Aemoo) and manual filtering, for future integration in Aemoo.

As far as information presentation is concerned, Google resulted to be the best system among the three. Although this may be a fair assessment, it is reasonable to think that this judgement is due to the extreme popularity of its interface. If we consider only Aemoo and RelFinder, systems with which the participants were unfamiliar with (cf. Figure 6), Aemoo received a higher number of positive comments concerning information presentation, which further supports the positive outcomes of the EKP assessment analysis and of the SUS analysis. Participants particularly appreciated Aemoo’s browsing interface and the way relations among entities were visually presented (cf. Figure 10). This suggests that the concept map metaphor, implemented as a radial graph with one-degree connections, is a suitable solution for visually exploring Linked Data. Also, users reported positive comments about the visualisation of explanations, which they found useful, indicating that Aemoo succeeds in providing this type of data-on-demand.

A final aspect worth remarking is the feedback about relevance and unexpectedness, which together provide an indication of the capability of the system to produce serendipitous results. Serendipity can be informally defined as beneficial discovery that happens in an unexpected way, it has been recently described as unexpected relevance [49]. The intuition is simple: the more a result is at same time relevant and unexpected, the more it is serendipitous. However, this is a tricky aspect to evaluate, due to its strong subjective character. Furthermore, considering the relatively small population involved in our experiments, we have insufficient evidence for claiming significance. Notwithstanding these limits, the results we obtained are worth reporting and allow us to formulate reasonable speculation on their interpretation. The results of the self-assessment questionnaire (cf. Figure 6) show that all participants declared comparable level of knowledge about the experiment subjects (i.e. low standard deviation values). This in addition to the good ratings provided for relevance and unexpectedness suggests that Aemoo shows a promising behaviour as far as serendipity is concerned, i.e. it was able to provide users with relevant and unexpected results, during the experiments.

6. Related work

Many existing solutions for exploring Linked Data are based on semantic mash-up or browsing applications. Examples are [53,24,25] that leverage the semantic relations asserted in the linked datasets without applying any criterion for defining a boundary to provide users with tailored and contextualized knowledge. For example, RelFinder [24] provides the visualisation of existing relations between two of more DBpedia entities. These relations can be simple or can include more complex paths. The visualisation of relations can be manually filtered by the user according to criteria of relations length, entity types and property names.

An increasing number of research has been done on more sophisticated relevance criteria for addressing the knowledge boundary and the heterogeneity problems for summarising, recommending or browsing Linked Data. Many of them present novel approaches, however to best of our knowledge, none of them is based on a solution that leverages ontology patterns (i.e., EKPs), which is the main contribution that distinguish our approach from existing ones.

Entity summarisation. In [50] is presented a diversity-aware algorithm, called DIVERSUM, that generates graphical entity summaries extracted from Wikipedia.
Differently from Aemoo, the algorithm selects triples related to an entity by both measuring their relevance and diversity and it is aimed at providing the most important triples about an entity with the higher coverage of diverse information. Similarly to Aemoo, DIVERSUM visualises entity summaries by using a radial neighbourhood graph that can be compared to a concept map. Although the algorithm shows good results, the approach is only focused on the selection of triples to be presented in a graph and there is no support to user exploration, e.g., explanations or interactive graphs. The approach proposed by RELIN \[9\] differs from Aemoo because it computes entity summaries by using a variant of the random surfer model, which is based on two kinds of actions, i.e., relational move and informational jump that follow non-uniform probability distributions. It leverages the relatedness and informativeness of description elements for ranking entity triples using patterns. A similar approach is presented by SUMMARUM \[51\] that uses the PageRank algorithm. The PageRank is computed in order to assign relevance scores to the triples having as subject a certain DBpedia entity. Both Relin and SUMMARUM rely on a PageRank like algorithm for building entity summaries, while the summarisation performed by Aemoo is pattern based.

Browsing Linked Data \[13\] presents a detailed discussion about the challenges and the requirements to consume and visualise Linked Data along with an analysis of the state of the art about existing solutions and systems that support the presentation, either textual or visual, of Linked Data. According to the classification provided by \[13\], Aemoo can be considered a visualisation-based approach to explore Linked Data. Similarly to existing solutions, such as \[5,12,24,22,6,40,54\], Aemoo is designed to enable the consumption of Linked Data by lay users, by providing visualization, filtering, data overview, data merging, and browsing capabilities in order to reduce the cognitive load and support humans in knowledge exploration and discovery (cf. Section \[3\]). Differently from systems like \[5,24,54\] Aemoo does not provide users with explicit filtering widgets for enabling, for example, faceted browsing of data, because the filtering mechanism is transparent to the user and relies on EKPs. Additionally, Aemoo filters, aggregates and visualises data from a different perspective, i.e., it focuses on the structure of wikilinks, which are untyped hyperlinks among Wikipedia pages, which reflects the way things are intuitively connected by humans in Wikipedia pages.

Other systems that rely on faceted browsing for filtering results are Yovisto \[55\], which is a platform that provides exploratory capabilities specialised in academic lecture recordings and conference talks and Visor \[42\]. This system facilitates the navigation process by introducing a multi-pivot paradigm, which allows users to identify key elements in the data space, called pivots. Different filtering solutions are proposed by the Discovery Hub \[29\], LED \[31\] that mainly differ from Aemoo because the user’s exploratory path is used for computing results at each exploratory step. Hence, the filtering mechanism does not depend on fixed schemas such as EKPs in Aemoo. In fact, the Discovery Hub uses a spreading activation algorithm for weighting an origin entity and consequently propagating the weights to its neighbours, while LED exploits users’ query in order to create tag clouds aimed at suggesting related knowledge to users during exploration search tasks.

The radial visualisation used by Aemoo for presenting data has been widely adopted so far, e.g. \[22,24\], and it is in general a well known visualisation metaphor in literature \[13\]. Aemoo uses the radial visualisation for rendering concept maps deriving from EKPs that provide a cognitive sound and an intuitive way for representing knowledge \[35\]. There are systems that allow to build graphical concept maps, such as Cmap \[29\] or IsaViz \[40\]. However, EKPs were conceived as units of meaning to be used not only for data presentation, but more in general, for data exchange. Also EKPs provide a relevance criterion for automatically filtering data, while Fresnel lenses only support the presentation design. In fact, Fresnel lenses provide more details about how to actually present data. A possible future development for Aemoo is to design a mapping between EKPs and Fresnel in order to allow the design of customised presentation interfaces for EKP-based filtered data.

---

29 \url{http://cmap.ihmc.us}  
30 \url{http://mes.github.io/marbles/}
Recommending Linked Data. To some extent, the EKP-based filtering of Aemoo can be abstracted to a recommendation mechanism. Although recommending systems are not directly comparable with Aemoo, for the sake of completeness we report here some relevant work in this area.

MORE [32] leverages DBpedia, Freebase and LinkedMDB in order to recommend movies. It computes similarities between movies thanks to an adaptation of the Vector Space Model (VSM). Seevl [35] is a recommendation system that provides personalized access and exploration of a knowledge base about music facts, which is created by exploiting DBpedia. The core of the system is an algorithm, called DBrec, which computes the relatedness among entities of the knowledge base by looking at shared relations, both incoming and outgoing. The authors in [26] present a recommendation system for retrieving music related to a point of interest (POI). The system exploits a spreading activation algorithm in order to weight the relatedness between musicians and POIs in DBpedia.

7. Conclusions and future work

This paper presents a novel approach for Linked Data exploration that uses Encyclopedic Knowledge Patterns (EKPs) as relevance criteria for selecting, organizing, and visualizing knowledge. The EKPs were discovered by means of a method aimed at mining the linking structure of Wikipedia. Based on this approach, a system called Aemoo was implemented for supporting EKP-driven exploration as well as integration of data coming from heterogeneous resources, namely static (i.e., DBpedia and Wikipedia) and dynamic knowledge (i.e., Twitter and Google News).

Our work grounds on two working hypotheses: (i) the cognitive soundness of EKPs and (ii) their effectiveness in summarising and aggregating knowledge to address exploratory search tasks.

The first hypothesis was validated by means of a user-study aimed at making emerge EKPs from human consensus in order to compare these EKPs with those extracted from Wikipedia. Results showed high values of correlation combined with good inter-rater agreement and very high reliability of human raters.

The second hypothesis was validated by means of controlled, task-driven user experiments in order to assess its usability, and ability to provide relevant and serendipitous information as compared to two existing tools: Google and RelFinder.

Currently, we are working on several extensions. Examples include:

- improving the automatic interpretation of hyper-text links by hybridizing NLP with Semantic Web techniques. In this respect we have recently obtained [44] very good results by designing a novel Open Knowledge Extraction (OKE) approach and its implementation, called Legalo [31], that performs unsupervised, open domain, and abstractive knowledge extraction from text for producing directly usable machine readable information;
- providing visual analytics interfaces that compare different entities having the same type;
- providing different views on the same entity by allowing users to change the applied lens, i.e., EKP;
- adding a basket functionality, allowing users to save the summary data of their exploration in RDF;
- integrating the EKP-based approach with user profiles for boundary creation.

References


3http://wit.istc.cnr.it/stlab-tools/legalo


