

# Aemoo: Linked Data Exploration based on Knowledge Patterns

**Editor(s):** Aba-Sah Dadzie, The Open University, UK; Emmanuel Pietriga, INRIA (France) & INRIA Chile

**Solicited review(s):** Aba-Sah Dadzie, The Open University, UK; Jan Polowinski, Technische Universität Dresden, Germany; Mariano Rico, Universidad Politécnica de Madrid, Spain

Andrea Giovanni Nuzzolese <sup>a,\*</sup>, Valentina Presutti <sup>a</sup>, Aldo Gangemi <sup>a,c</sup>,  
Silvio Peroni <sup>a,b</sup>, and Paolo Ciancarini <sup>a,b</sup>

<sup>a</sup> *STLab-ISTC, Consiglio Nazionale delle Ricerche, Rome, Italy*

<sup>b</sup> *Department of Computer Science and Engineering, University of Bologna, Bologna, Italy*

<sup>c</sup> *LIPN, Université Paris 13, Sorbone 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 ReIFinder.

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

## 1. Introduction

In the Semantic Web vision [7] agents were supposed to leverage the Web knowledge in order to help humans in solving knowledge-intensive tasks. Nowadays, Linked Data is feeding the Semantic Web by publishing datasets that rely 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 (a.k.a. knowledge boundary problem [22]) for providing tailored data.

The heterogeneity problem is due to 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).

On the other hand, the knowledge boundary problem consists in the difficulty of identifying what configuration of data is 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. If the topic of the

---

\*Corresponding author. E-mail: andrea.nuzzolese@istc.cnr.it.

summary is the philosopher Immanuel Kant, this application should provide users with tailored information concerning facts about Kant’s major works and ideas, but should skip facts that are too peculiar or curious, such as the nationality of his grandfather.

Elsewhere [22] we introduced a vision for the Semantic Web based on *Knowledge Patterns* (KP) as basic units of meaning for addressing both the heterogeneity and the knowledge boundary problems. More recently, we introduced Encyclopedic Knowledge Patterns [37] (EKP) as a particular kind of KP that are empirically discovered by mining the linking structure of Wikipedia pages. EKPs provide knowledge units that can answer the following competency question:

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

For example, the EKP for describing philosophers should possibly include types such as book, university, and writer because 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 exploit 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 [30], 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 provide a unifying view as well as 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 the following:

- a detailed description of the **Aemoo**<sup>1</sup> visual exploratory search system, based on the extraction of EKPs from Wikipedia and their evaluation (formerly explained in [37]). We shortly introduced Aemoo in [38], whereas in this paper we provide an exhaustive description of the system, and the method used for applying EKPs in order to provide entity summarisation and knowledge aggregation;
- an original 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 [54,26,27]. 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 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 [19], artificial intelligence [32,3], cognitive sciences [4,21] and more recently in the Semantic Web [22]. As discussed in [22], it is possible to identify a shared meaning for KPs across those different theories, as “a structure that is used to organize our knowledge, as well as for interpreting, processing or anticipating information”.

---

<sup>1</sup><http://www.aemoo.org>

In linguistics, *frames* are a form of KPs that were introduced by Fillmore in 1968 [19], 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 theory of case frames, and in 1976 he introduced *frame semantics* [20]. According to Fillmore a frame is

“...any system of concepts related in such a way that to understand any one of them you have to understand the whole structure in which it fits; when one of the things in such a structure is introduced into a text, or into a conversation, all of the others are automatically made available.” [20]

A frame is comparable to a cognitive schema, another form of KPs, since it has a prototypical form that can be applied to a variety of concrete cases that fit that form. According to cognitive science theories, e.g. [4], humans are able to recognize frames, to repeatedly apply them in so-called *manifestations* of a frame, and to learn new frames that can become part of their competence. Frames are considered 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 [32], who recognized that frames convey both cognitive and computational value in representing and organizing knowledge. His definition was:

“...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.” [32]

In knowledge engineering, the term Knowledge Pattern was used by Clark [11]. The notion of KP introduced by Clark is slightly different from frames as introduced by both 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. Clark recognizes KPs as general templates denoting recurring theory schemata, and his approach is similar to the use of theories and morphisms in the formal specification of software.

More recently, Knowledge Patterns have been revamped in the context of the Semantic Web by Gangemi

and Presutti [22]. Their notion of KPs encompasses those proposed by Fillmore, Minsky, and Clark, and goes further envisioning KPs as research objects of knowledge engineering and the Semantic Web from the viewpoint of empirical science.

In [37] we introduced the *Encyclopedic Knowledge Patterns* (EKPs). EKPs were discovered by mining the structure of Wikipedia articles. They are a special kind 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 *defining* or *describing* the entity referenced by the article. DBpedia, accordingly with this hypothesis, has RDF-ized a) the entities referenced by articles as *resources*, b) the wikilinks as *relations* between those resources, and c) the types of the resources as OWL *classes*. EKPs are grounded in the assumption that wikilink relations in DBpedia, i.e. instances of the `dbpo:wikiPageWikiLink` property<sup>2</sup>, convey a rich encyclopedic knowledge that can be formalized as knowledge patterns.

An EKP is a small ontology that defines a class  $S$ , and the typical relations  $R_i$  between individuals from  $S$ , and other individuals from the most relevant classes  $C_j$  that are used to describe the individuals from  $S$ . In [37] we defined a method for extracting EKP 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<sup>3</sup>. For this purpose, we used an OWL vocabulary, called *knowledge architecture*<sup>4</sup> [44], 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 prop-

<sup>2</sup>Prefixes `dbpo:`, `dbpedia:`, and `ka:` stand for `http:dbpedia.org/ontology/`, `http:dbpedia.org/resource/` and `http://www.ontologydesignpatterns.org/ont/lod-analysis-path.owl`, respectively.

<sup>3</sup>The dataset is named `dbpedia_page_links_en`

<sup>4</sup>`http://www.ontologydesignpatterns.org/ont/lod-analysis-path.owl`



Fig. 1. The UML class diagram for the EKP `ekp:Philosopher`. The arrows among classes represent universal restrictions.

erty 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-up fashion. For this purpose we used a function, called  $pathPopularity(P_{i,j}, S_i)$ , 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  $pathPopularity(P_{i,j}, S_i)$  values are above a certain threshold  $t$  ranging from 0 to 1. In order to decide a value for  $t$ , we built a prototypical ranking of the  $pathPopularity(P_{i,j}, S_i)$  scores, called  $pathPopularity_{DBpedia}$ . Then, we computed a K-Means clustering on  $pathPopularity_{DBpedia}$  scores to hypothesise a threshold value for  $t$ . The clustering generated one large cluster (85% of the 40 ranks) with  $pathPopularity_{DBpedia}$  scores below 0.18% and three small clusters with  $pathPopularity_{DBpedia}$  scores above 0.23%. Hence, we set  $t = 0.18\%$ . We recommend to read our previous work [37] for further details about the identification of the threshold  $t$ ;

- OWL2 formalisation 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`<sup>5</sup> resulting from our method. It reports the types of entities that are most frequently (according to  $pathPopularity$  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<sup>6</sup> and published them into the ODP repository<sup>7</sup>.

### 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)

<sup>5</sup>The prefix `ekp:` stands for the namespace <http://www.ontologydesignpatterns.org/ekp/>.

<sup>6</sup>As at DBpedia version 3.7.

<sup>7</sup><http://ontologydesignpatterns.org/aemoo/ekp/>

sources. We presented a preliminary version of Aemoo in [38].

We assume that the human action of linking entities on the Web reflects the way humans 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* and *filtering* of information. Enrichment and filtering are the two actions performed in order to address the knowledge heterogeneity and boundary problems. Users are guided through their navigation by both reducing and focusing the amount of available data : given an entity, instead of being presented with a bunch of triples, or a big unfocused graph, users navigate through *units of knowledge*, and move between them without losing the overview of the entity. In practice, 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:

- a semi-automatic approach that leverages a DBpedia index based on the Entityhub<sup>8</sup> component of Apache Stanbol<sup>10</sup>. 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;
- a completely automatic approach based on Entity Linking<sup>11</sup>. Aemoo uses the Enhancer component of Apache Stanbol for entity linking. In this case, the identity of the subject of a user query is resolved against DBpedia, by choosing the entity with the highest linking 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  $(T_1, \dots, T_n)$  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<sup>12</sup>, 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 following:

- 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`<sup>13</sup>, 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 [49]).

<sup>8</sup>The Entityhub is a component of the Apache Stanbol project, which relies on Apache Solr<sup>9</sup> for building a customised entity-centric index of a linked dataset.

<sup>10</sup><http://stanbol.apache.org>

<sup>11</sup>Entity linking is the ability to resolve the referent of a term in a text, against an entity from a known knowledge base.

<sup>12</sup>The subject of a user query.

<sup>13</sup><http://ontologydesignpatterns.org/ekp/owl/Thing.owl>

As an example, the type `dbpo:Philosopher` would be selected for the entity `dbpedia:Immanuel_Kant` and the EKP `ekp:Philosopher`<sup>14</sup> retrieved from the index accordingly. Figure 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 subject entity. EKPs are used for automatically generating 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` EKP:

```

1.  CONSTRUCT {
2.    ...
3.    ?entity a ?type .
4.    dbpedia:Immanuel_Kant
5.      ?ekp_property ?entity
6.    ...
7.  }
8.  WHERE{
9.    GRAPH <dbpedia_page_links_en> {
10.     dbpedia:Immanuel_Kant
11.       dbpo:wikiPageWikiLink ?entity .
12.    }
13.    GRAPH <dbpedia_instance_types_en> {
14.     ?entity a ?type .
15.    }
16.    GRAPH <ekp_philosopher> {
17.     ...
18.     ?ekp_property
19.       rdfs:domain dbpo:Philosopher .
20.     ?ekp_property
21.       rdfs:range ?type
22.     ...
23.    }
24.  }
```

Where, `dbpedia_page_links_en`, `dbpedia_instance_types_en`, and `ekp_philosopher` are three datasets in our SPARQL endpoint<sup>15</sup>, 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, since it filters out the wikilinks that are not typical 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 10 and 11;
- each linked entity is bound to its type at line 14;
- linked entities are filtered based on their types, according to the EKP. Based on the definition of type paths [37], we have:
  - \*  $S_i = \text{dbpo:Philosopher}$ ;
  - \*  $q = ?\text{ekp\_property}$ ;
  - \*  $O_j = ?\text{type}$ ;

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 generated during the formalisation step of the method described in Section 2. By constraining the results with domain and range of `?ekp_property` (lines 18-21), the result of the query consists of 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:

```

1.  dbpedia:Immanuel_Kant
2.    a dbpo:Philosopher ;
3.    rdfs:label "Immanuel Kant"@en ;
4.    ekpS:linksToCity
5.      dbpo:Königsberg ;
6.    ...
7.    ekpS:linksToLanguage
8.      dbpo:Latin ;
9.    ...
10.   ekpS:linksToBook
11.     dbpo:Critique_of_Pure_Reason .
12.
13.  dbpo:Königsberg a dbpo:City ;
14.    rdfs:label "Königsberg"@en .
15.
16.  dbpo:Latin a dbpo:Language ;
17.    rdfs:label "Latin"@en .
18.
19.  dbpo:Critique_of_Pure_Reason
20.    a dbpo:Book ;
21.    rdfs:label
22.      "Critique of Pure Reason"@en .
```

Aemoo further enriches the resulting model, by identifying possible semantic relations that provide

<sup>14</sup><http://ontologydesignpatterns.org/ekp/owl/Philosopher.owl>

<sup>15</sup><http://wit.istc.cnr.it:8894/sparql>

a label to 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`<sup>16</sup>. These semantic relations are added to the previous RDF model by means of `skos:relatedMatch` axioms. For example:

```
1.  ekp_phil:linksToCity
2.    a owl:ObjectProperty ;
2.    skos:relatedMatch
3.      dbpo:birtPlace, dbpprop:placeOfBirth .
```

A wikilink is a hyperlink in a Wikipedia page that is (almost<sup>17</sup>) always surrounded by a piece of text. This piece of text is very important because it can be used as linguistic evidence for explaining the nature of the relation between a subject entity and the object entity linked by a wikilink. Thus, Aemoo extracts this piece of text (with a maximum of 50 words) and enriches the model with an OWL2 annotation representing this linguistic evidence. For example, given the triple:

```
dbpedia:Immanuel_Kant
  ekpS:linksToCity
    dbpedia:Königsberg
```

its annotation will be the following:

```
1.  [] a owl:Axiom ;
2.    owl:annotatedSource
3.      dbpedia:Immanuel_Kant ;
4.    owl:annotatedProperty
5.      ekpS:linksToCity ;
6.    owl:annotatedTarget
7.      dbpedia:Königsberg ;
8.    grounding:hasLinguisticEvidence:
9.      aemoo:wiki-sentence .
10.
11.  aemoo:wiki-sentence a doco:Sentence ;
12.    frbr:partOf wikipedia:Immanuel_Kant
13.    c4o:hasContent
14.      "Immanuel Kant was born in 1724
15.      in Königsberg, Prussia (since 1946
16.      the city of Kaliningrad,
17.      Kaliningrad Oblast, Russia), as
18.      the fourth of nine children."@en .
```

<sup>16</sup>This information is limited by the coverage of DBpedia

<sup>17</sup>An exception is represented, for example, by the infoboxes having wikilinks in table cells without any surrounding text.

```
19.
20.  wikipedia:Immanuel_Kant
21.    a fabio:WikipediaEntry .
```

Where the triple is annotated with the property `grounding:hasLinguisticEvidence`<sup>18</sup>, while the linguistic evidence is identified by `aemoo:wiki-sentence`<sup>19</sup>. The linguistic evidence is (i) typed as a `doco:Sentence`<sup>20</sup> [12], is (ii) declared to be extracted from Wikipedia with the property `frbr:partOf`<sup>21</sup> and (iii) associated with the text by means of the property `c4o:hasContent`<sup>22</sup> [15]

**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 as follows:

```
1.  [] a owl:Axiom ;
2.    owl:annotatedSource
3.      dbpedia:Immanuel_Kant ;
4.    owl:annotatedProperty
5.      ekpS:linksToCity ;
6.    owl:annotatedTarget
7.      dbpedia:Rome ;
8.    grounding:hasLinguisticEvidence
```

<sup>18</sup>The prefix `grounding:` identifies the ontology <http://ontologydesignpatterns.org/cp/owl/grounding.owl>.

<sup>19</sup>The prefix `aemoo:` identifies the local namespace used by Aemoo.

<sup>20</sup>The prefix `doco:` identifies the ontology <http://purl.org/spar/doco>.

<sup>21</sup>The prefix `frbr:` identifies the ontology <http://purl.org/vocab/frbr/core>.

<sup>22</sup>The prefix `c4o:` identifies the ontology <http://purl.org/spar/c4o> [15].

```

9.         twitter:tweet_id .
10.
11.  twitter:tweet_id a fabio:Tweet ;
12.     c4o:hasContent
13.         "Lots of people love to
14.         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`<sup>23</sup>. The tweet is (i) typed as `fabio:Tweet`<sup>24</sup> [40] 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 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 central for philosophers.

**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. Section 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 two subjects as an example: “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 is more unusual to find for office holders, then Aemoo generates a complementary entity-centric model for the subjects based on curiosities. This model will include, for example, 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 [14], starting from the conclusion of Shneiderman [47], debate about the requirements that visualisation systems should fulfil 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.

The method described in Section 3.1 was designed to address requirements R1 and R3. 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

<sup>23</sup>The `tweet_id` has to be replaced with an actual tweet identifier available from Twitter.

<sup>24</sup>The prefix `fabio:` identifies the ontology <http://purl.org/spar/fabio> [40].



> Immanuel Kant

(a) Initial summary page for the query: *Immanuel Kant*.

> Immanuel Kant

(b) Exploring knowledge between *Immanuel Kant* and *Königsberg*. On the left side there is a list of explanation snippets that provide evidences of the relation with respect to the enabled sources.

Fig. 2. Aemoo user interface.

search means to reduce the cognitive load required for activities like look-up, learning and investigation [30]. Novak and Gowin [36] demonstrated that *concept maps* are an effective means for representing and communicating knowledge for supporting humans in un-

dertaking, understanding and learning tasks. Concept maps, introduced by [34], are 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

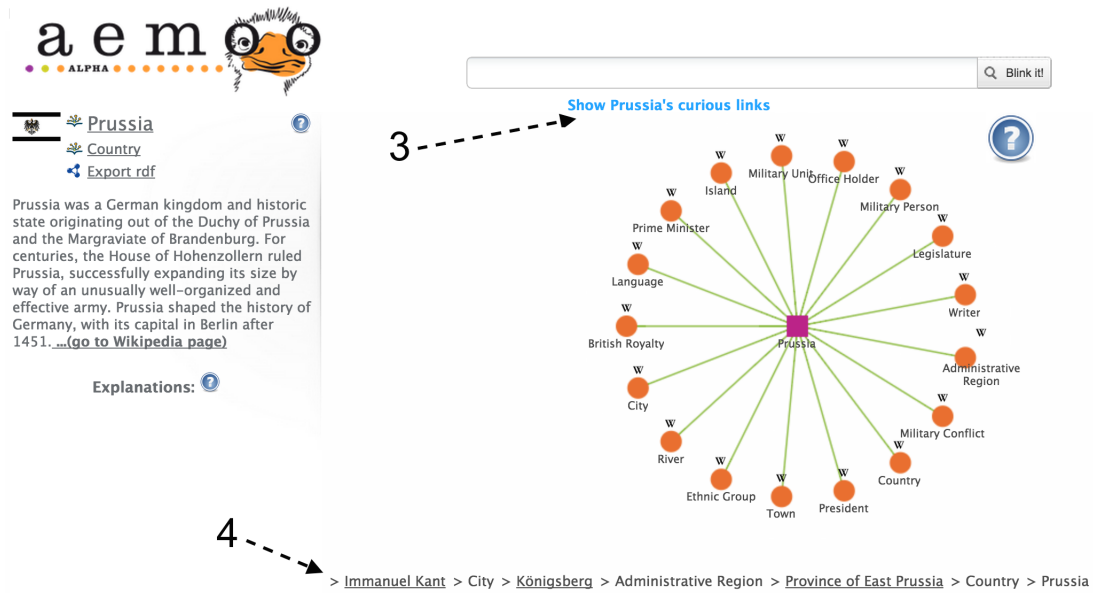


Fig. 3. Aemoo: breadcrumb

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* [13] for addressing exploratory search tasks with Aemoo.

In fact, the concept map related to a subject entity is visualised by Aemoo using a radial graph with one-degree connections 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, referred as a *bona fide* metaphor for presenting data by some authors [17], was preferred to other visualisation solutions (e.g., tabular, tree, etc.) because it directly mirrors the structure of an EKP (cf. Figure 1) and provides a “compass” deriving from the radial layout that is meant to be intuitive for the user navigation (cf. requirement R2). The compass allows

to summarise and organise knowledge and can be used by the users (i) to enable exploratory features and (ii) to provide a mechanism to detail data on demand (cf. requirement R6). In fact, the user can hover with the pointer on the circular nodes of the graph in order to expand its knowledge and gather more and more detailed information. Additionally, the radial graph is interactive: all its elements can be clicked, including the new interface objects that are triggered by navigating it, causing the user to change and enrich the focus of the exploration. The interaction aspect is designed to address requirement R4, i.e. exploratory knowledge discovery. It is worth noting that, in case of much information, radial graphs can easily display messy and overcrowded layouts. However, in Aemoo the radial graph with one-degree connections prevents such situations as the graph goes to only one level below the root and the relationships are unweighted. Additionally, EKPs capture only 10 type paths on average (cf. our paper on EKPs [37]). Thus, in the most frequent case, Aemoo presents a radial graph having a subject entity surrounded by 10 node sets at most. Moreover, radial layouts allow more easily to keep the focus on the central area of the graph [9] that, in the case of Aemoo, includes both the subject entity and the node sets. Hence, the radial graph of Aemoo<sup>25</sup> allows to organise the entity-centric knowledge in a way that is com-

<sup>25</sup>Based on a graph with one-degree connections.

compact and focused to the visualisation of the relations between a subject entity and its surrounding node sets. For example, a tree layout would scale less easily. In case of many node sets, it would require more horizontal space in order to display the subject entity to the top, while all node sets at the same level below the subject entity. Similarly, a tabular layout would organise the subject entity and its related node sets in a more scattered way, if compared to the simple relations that Aemoo presents between a subject entity and its node sets. We remark that this rationale was intended for the Aemoo case. It cannot be applied to radial graphs with multiple-degree connections.

Figure 2(a) shows an example of a concept map having Immanuel Kant as subject entity. Aemoo splits the interface into two parts: (i) on the center-right side it visualises the concept map and (ii) on 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 are designed to address requirements R3, R4 and R6:

- a square box (cf. the arrow with id 1 in Figure 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 in-

terface prototype. Most of the feedback suggested minor changes (e.g., on the positioning of the abstract and explanations), which we implemented. Nevertheless, a major request emerged from most users to keep the graph visualising 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 an unlabeled edge representing a set of wikilink relations. Aemoo also shows a list of semantic relations extracted from DBpedia that might express the intension of the unlabeled relation. This list is displayed in a tooltip appearing by mouse hovering the pointer on any edge (cf. Section 3.1). Such relations are (i) represented as `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, Figure 2(a) shows a tooltip for the edge expressing a set of wikilink relations between Immanuel Kant and the node set City. Such a tooltip provides `dbpo:birthPlace` and `dbpprop:placeOfBirth`<sup>26</sup> as possible semantic relations for that type of link;
- a list of explanations appears in a widget on 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

<sup>26</sup>The prefix `dbpprop` stands for the namespace `http://dbpedia.org/property/`.

The screenshot shows the Aemoo interface for the entity "Immanuel Kant". On the left, there is a profile card with a portrait, the name "Immanuel Kant", and the role "Philosopher". Below this is a short biography and an "Export rdf" link. To the right, a radial graph titled "Show Immanuel Kant's core links" has "Immanuel Kant" at the center. It is connected to six nodes: "Settlement", "Musical", "Nerve", "Publisher", "Artist", and "Person". A callout box highlights the "Nerve" node with the text "Nerve" and "Optic nerve W". A search bar at the top right contains the text "Blink it!".

> [Immanuel Kant](#) > Immanuel Kant (curiosity)

(a) The radial graph displayed by Aemoo contains curiosities about Immanuel Kant.

The screenshot shows the Aemoo interface for the entity "Prussia". On the left, there is a profile card with the name "Prussia" and the role "Country". Below this is a short biography and an "Export rdf" link. To the right, a radial graph titled "Show Prussia's core links" has "Prussia" at the center. It is connected to ten nodes: "Noble", "Building", "Election", "Bird", "Politician", "Political Party", "Person", "Chancellor", "Settlement", and "Body Of Water". A callout box highlights the "Body Of Water" node with the text "Body Of Water" and "Baltic Sea W". A search bar at the top right contains the text "Blink it!".

> [Albert of Brandenburg Prussia](#) > [Alan Turing](#) > [Prussia](#) > Prussia (curiosity)

(b) The radial graph displayed by Aemoo contains curiosities about Prussia.

Fig. 4. Aemoo displaying curiosities about Alan Turing and Prussia.

the concept map. For example, the arrow with id 2 in Figure 2(b), points to the explanations of 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. Ex-

planations 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 fulfils 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 link located in top-center of the interface under the search bar (cf. the arrow with id 3 in Figure 2(b)) 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 presentation of the core knowledge about a subject entity in order to keep the interface coherent, and to ensure a smooth interaction between the two views. Figure 4(a) and Figure 4(b) show the radial graphs containing the curiosities about Alan Turing and Prussia respectively. It is possible to note how these graphs report unusual or less common relations between the subject entities and the entity types identified by the node sets, e.g., the relation between Alan Turing and Optic nerve (cf. 4(a)) and that between Prussia and Baltic Sea (cf. 4(b)).

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

### 3.3. Implementation details

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.

Table 1  
UI components with respect to design requirements.

UI component	R1	R2	R3	R4	R5	R6
Concept map	✓	✓	✓			✓
Entity abstract	✓					
Entity box		✓				✓
Explanation widget						✓
Breadcrumb				✓	✓	

The overview of the architecture at the server side, including also the components for EKP extraction, is depicted in Figure 5.

The architecture is designed by using the Component-based [23] and the REST [18] architectural styles.

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

- KP extraction coordinator which takes care of the coordination of the overall extraction process;
- Property path identifier that is responsible for the identification of type paths;
- 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.

Aemoo is composed of the following components:

- Aemoo coordinator which coordinates all the activities;
- Identity resolver which is in charge of resolving an user query with respect to entity in Linked Data;
- KP selector that selects an appropriate KP according to the entity identified;
- Knowledge filter which takes care of applying a KP on raw RDF data;
- Knowledge aggregator that aggregates knowledge from other sources with respect to the KP selected.

All components are implemented as Java OSGi [53] bundles, components and services, and some of them can be accessed through the RESTful interfaces exposed by the Aemoo REST provider (i.e., Aemoo coordinator and KP selector) and the KP extractor REST provider (i.e., KP extraction coordinator, KP repository manager).

The client side interacts with the other components via REST interfaces through AJAX. Additionally, it handles the visualisation of Aemoo through the

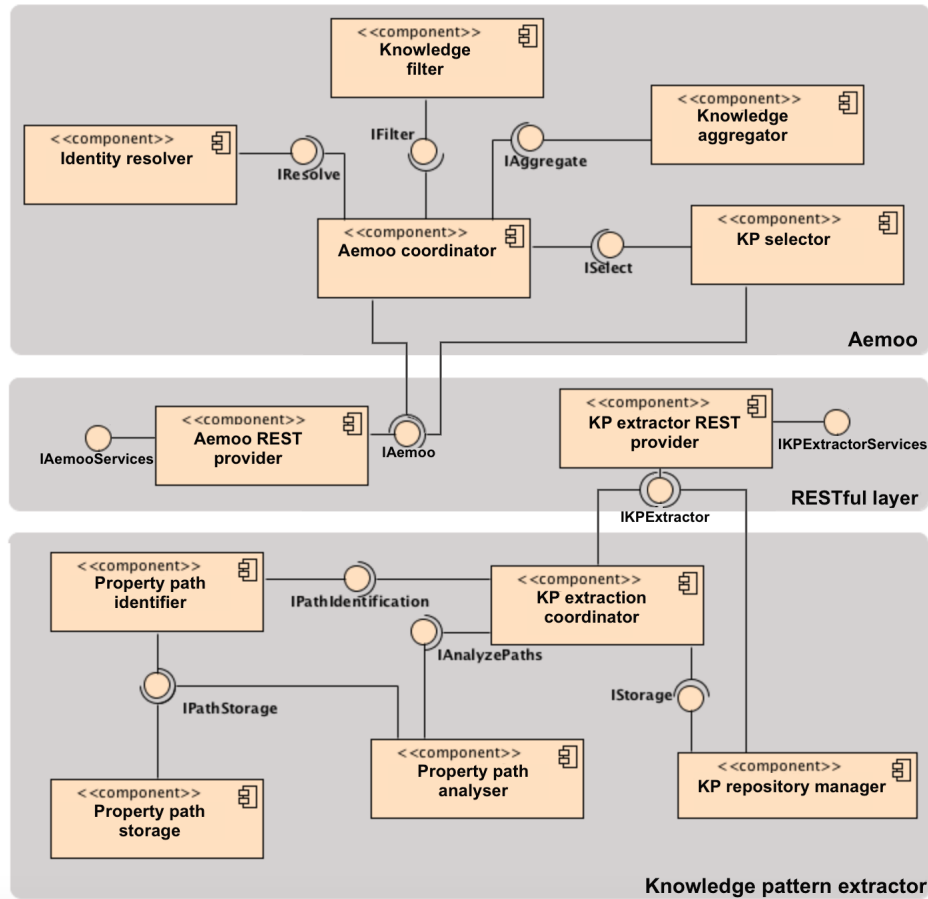


Fig. 5. Overview of the architecture of Aemoo and the EKP extractor.

JavaScript InfoVis Toolkit<sup>27</sup>, a Javascript library that supports the creation of interactive data visualisations for the Web.

#### 4. Evaluation

In Section 1 we hypothesised that EKPs provide intuitive entity-centric summaries. We also hypothesised that EKPs can be exploited for visualising Linked Data in order to help humans in exploratory search tasks. In Section 4.1 we summarise the experimental setup we defined in [37] and used for assessing the cognitive soundness of EKPs. While, in Section 4.2 we describe the experimental setup used for assessing our working hypothesis.

##### 4.1. Cognitive soundness of EKPs

In [37] we carried out a user-based study to assess the cognitive soundness of EKPs. We intended to make EKPs emerge from human consensus, and to compare them to those extracted automatically from Wikipedia. In that study, 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

<sup>27</sup><http://thejit.org/>

are perceived as sound units of meaning independently from one specific language or culture, at least for those represented in our study. Although the multi-cultural and multi-language character of EKPs cannot be assessed with proof<sup>28</sup>, the good inter-rater agreement (Kendall’s  $W = 0.68$ ) and the high reliability of the raters (Cronbach’s  $\alpha = 0.93$ ), provide an encouragement towards further dedicated study.

In order to compare the EKPs annotated by humans to the EKPs empirically extracted from Wikipedia, we computed the correlation between the scores assigned by participants, and a ranking function named  $pathPopularity_{DBpedia}$ , representing the trend of  $pathPopularity$  values for Wikipedia EKPs. We recorded high correlation on average (Spearman’s  $\rho = 0.75$ ) meaning that the identified value for  $t$  (cf. Section 2) is a stable criterion for determining knowledge boundaries for EKPs. More details can be found in our previous work [37].

#### 4.2. Experimental setup for evaluating Aemoo

In order to assess the validity of our hypothesis (i.e., EKPs provide intuitive entity-centric summaries that can support humans during exploratory search tasks) we carried out a user-based study, whose aim is 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 [30], 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 Google does not provide an interface specially designed for exploratory search, it is currently the most

used exploratory tool 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 provides a reference baseline; (ii) RelFinder [14] is a tool supporting visual exploratory search on linked data, 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 noted 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. In order to make the results comparable, we constrained the tools background knowledge to Wikipedia (for Google) and to DBpedia (for RelFinder), which constitute the data intersection of the three tools.

The three tasks of the controlled experiment are the following:

**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 be recorded as a set of triples <subject, objects, description>. For example, the fact that “Alan Turing worked at the University of Manchester” is a valid information to include in a summary about Alan Turing. Let us suppose that a participant is able to find this information with Aemoo (e.g., by navigating the node sets of the radial graph having

<sup>28</sup>It has to be remarked that we had a small number of participants and they were all highly educated, fluent in English and mainly from European countries. We assume that participants from European countries have many cultural and linguistic aspects (as speakers of Indo-European languages) in common.

“Alan Turing” as subject entity), then she has to record a triple as follows:

- subject=*Alan Turing*
- object=*University of Manchester*
- description=*Alan Turing worked at the University of Manchester*

Where the subject and the object identify the two elements of the relation found, and the description provides an explanation about the nature of such a relation, as it is understood by the participants (e.g., by reading the explanations provided by Aemoo).

For each identified triple, the user has 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 assigned a specific tool to use for exploring information with a maximum time of 10 minutes for each task. They have been instructed to include all possible triples they could get from the tool results, being them interesting, wrong, relevant, obvious, etc., and to rank them accordingly. They were invited to act so that 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, which have 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 their 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<sup>29</sup>, 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, this procedure was meant to foster participants to provide feedback based on the self-assessment of their user experience by also comparing different tools on the same tasks. The feedback were collected by means of open questions at the end of the experiment (cf. Table 2). 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 [26]) 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. The ten statements were those of the System Usability Scale (SUS) [8]. The SUS is a well-known metric used for evaluating the usability of a system. It has the advantage of being technology-independent, and reliable even with a very small sample size [46]. It also provides a two-factor orthogonal structure, which can be used to score the scale on independent Usability and Learnability dimensions [46].

Table 2 reports the five open questions aimed at collecting feedback (pros and cons) from the participants about the quality of their experience with Aemoo. The user feedback from this questionnaire has been used to perform a qualitative analysis of Aemoo based on Grounded theory [48]: a method often used in Social Sciences to extract relevant concepts from unstructured corpora of natural language resources (texts, interviews, or questionnaires).

We developed an ad-hoc web application named AemooEval<sup>30</sup> for supporting the experiment. The application 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 6 shows the interface of the tool during the 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

<sup>29</sup>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.

<sup>30</sup><http://wit.istc.cnr.it/sweng/cgi-bin/>



## Task [max 10 mins.]

The goal of this exercise is to make a summary on the topic "Alan Turing".

Show more

The screenshot shows the Aemoo interface for the topic "Alan Turing". At the top left is the Aemoo logo with a cartoon character. A search bar contains "Alan Turing" and a "Blink it!" button. Below the search bar, a sidebar on the left displays "Alan Turing" with a "Scientist" tag, an "Export rdf" button, and a Wikipedia-style bio: "Alan Mathison Turing, OBE, FRS (/ˈtʃʊərɪŋ/ TEWR-ing; 23 June 1912 – 7 June 1954) was a British mathematician, logician, cryptanalyst, computer scientist and philosopher. He was highly influential in the development of computer science, providing a formalisation of the concepts of "algorithm" and "computation" with the Turing machine, which can... (go to Wikipedia page)". Below the bio are "Explanations:" and a section titled "Early life and career" with a truncated bio: "... Orissa Province, in British India.[14][15] Turing's father, Julius Mathison Turing (1873-1947), was the son of a clergyman from a Scottish family of merchants which had been based in the Netherlands and included a baronet. Julius's wife, Alan's mother, was Ethel Sara (née Stoney; 1881-1976), daughter of Edward Waller Stoney, chief engineer of the Madras Railways. The Stoneys were a...". A central area shows "Show Alan Turing's curious links" with a network diagram where "Alan Turing" is the central node, linked to "City", "Town", "Country", "Administrative Region", "Disease", "University", "Award", and "Scientist". A highlighted box labeled "Country" lists "Saint Vincent and t...", "Wales", and "Netherlands". At the bottom, there are input fields for "Subject", "Object", and "Description", and two rows of radio buttons for "Relevancy" and "Unexpectedness", each with options 1, 2, 3, 4, 5.

Fig. 6. The interface of the Web-based tool designed and implemented for carrying on the user-study.

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 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, and equally distributed in terms of gender. All the participants were undergraduate students in computer science coming from 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 about how to use AemooEval, and a brief tutorial on the three tools i.e., Aemoo, RelFinder and Google.

Table 2  
Open questions for grounded theory-based analysis.

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

In order to assess the background and skills of participants, before running the experiments they were asked to rank<sup>31</sup> twelve statements, reported in Table 3, composing a self-assessment questionnaire. Fig-

Table 3  
Self-assessment questionnaire

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

ure 7 shows the averages of scores for each statement and their standard deviations. According to the results of the self-assessment questionnaire, participants confirmed to have little knowledge about 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 knowl-

<sup>31</sup>using a 5-point Likert scale from 1: Strongly Disagree to 5: Strongly Agree

edge 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 experiments.

## 5. Results and discussion

**Results.** 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 8. On average, Aemoo performs better than RelFinder and Google. The better result on average is due to Task 2, for which Aemoo significantly outperforms the other tools, while RelFinder has a slightly 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, and show the results in Figure 9(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 [46], 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 (average > 68), however we were also interested in investigating its usability in comparison to the other tools.

Figure 9 shows the  $p$ -values, computed by using the Tukey's HSD (honest significant difference) method, indicating the statistical significance of the pairwise

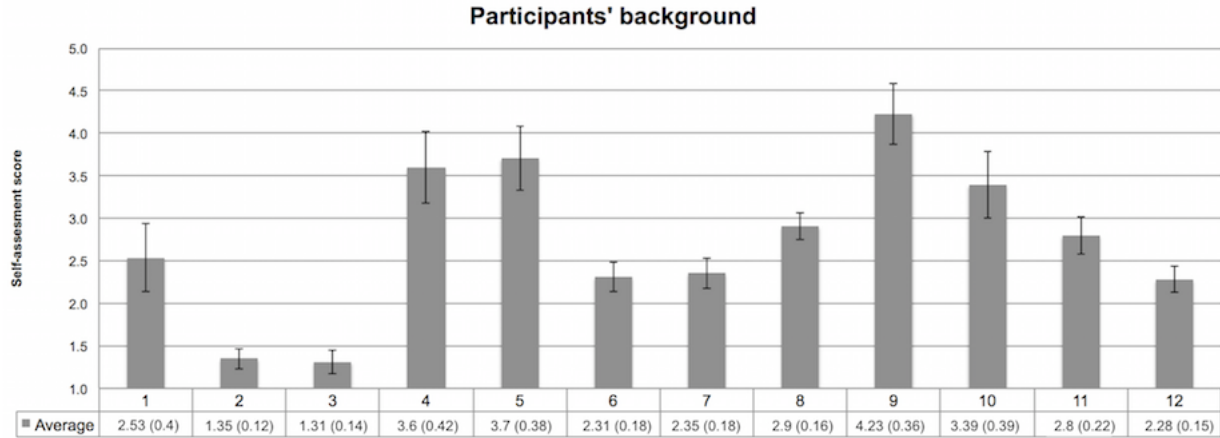


Fig. 7. Answers provided by participants to the questions concerning their background related to the experiment. The answers are recorded on a 5-points Likert scale ranging from 1 to 5. Question labels correspond to the question numbers in Table 3. Standard deviation values are expressed between brackets and shown as black vertical lines in the chart.

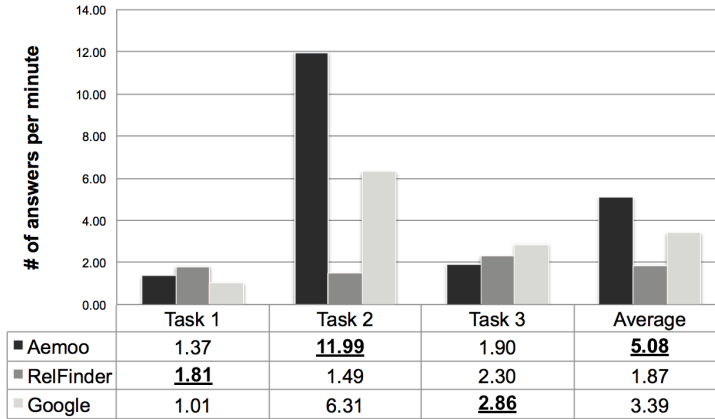
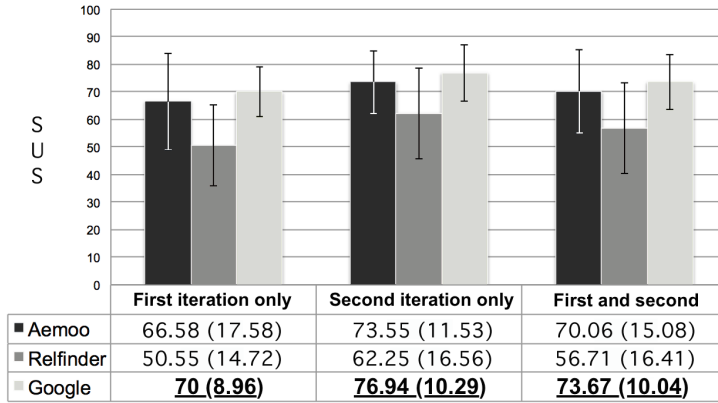


Fig. 8. Number of answers per minute for each task and tool.

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 overall SUS score, we recorded its two sub-parameters, i.e., Learnability and Usability [29], for the three systems. According to [29], the Learnability score is obtained by analysing the answers provided by participants to the SUS statements 1, 2, 3, 5, 6, 7, 8, and 9. Instead, the Usability score is

obtained by analysing the answers provided to the SUS statements 4 and 10. The final value obtained for these parameters is within the range [0 (Hard to learn/use), 1 (Easy to learn/use)]. The results of this analysis are reported in Figure 10. Figure 10(a) shows the learnability scores for the three systems and their standard deviations. According to Figure 10(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 10(c) shows the obtained scores (with their standard deviations). We have strong evidence for claiming that Aemoo is more usable than RelFinder (moderate evidence supports the same claim if we consider the two iterations, separately).



(a) SUS scores and standard deviation values for Aemoo, RelFinder and Google. Standard deviation values are expressed between brackets and shown as black vertical lines in the chart.

Comparison	First iteration only	Second iteration only	First and second iterations
Aemoo/RelFinder	$0.01 < p \leq 0.05$	$0.05 < p \leq 0.1$	$p < 0.01$
Aemoo/Google	$p > 0.1$	$p > 0.1$	$p > 0.1$
Google/RelFinder	$0.01 < p \leq 0.05$	$0.01 < p \leq 0.05$	$p < 0.01$

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

Fig. 9. SUS scores (cf. Figure 9(a)) and  $p$ -values (cf. Figure 9(b)).

The results of the open questionnaire (cf. Table 2) were analysed by using the Grounded Theory [48]. 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 11 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 12 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 4 shows the results by separating the values for the first

and the second iterations, respectively<sup>32</sup>, which confirm the previous ones with the only difference that Aemoo and RelFinder have the same results for relevance ratings in the first iteration.

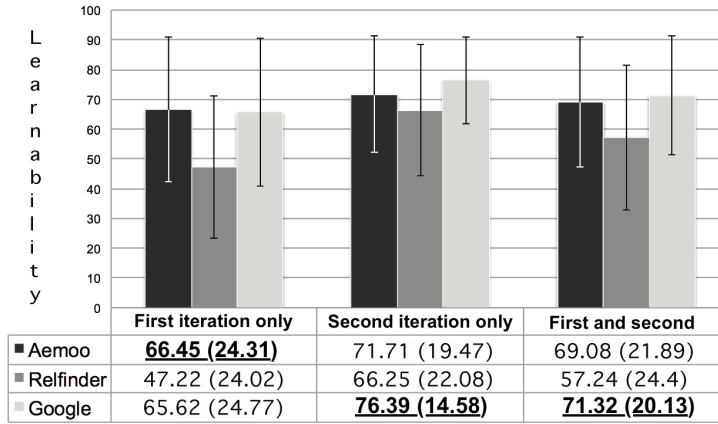
Table 4

Relevance and unexpectedness ratings according to participants' feedback, by taking into account the specific iteration.  $r_1$  and  $r_2$  indicate the relevance values reported during the first and second iteration, respectively. Similarly,  $u_1$  and  $u_2$  indicate unexpectedness ratings.

Tool	$r_1$	$u_1$	$r_2$	$u_2$
Aemoo	3.8	<b>2.82</b>	<b>4.42</b>	<b>3.02</b>
RelFinder	<b>3.81</b>	2.53	4.13	2.75
Google	4	2.18	4.1	2.22

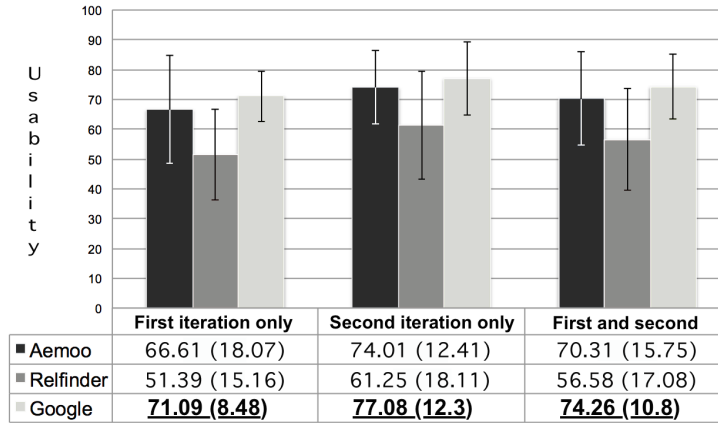
**Discussion.** In information retrieval, accuracy is the typical measure used to evaluate the performance of a system. The higher the precision and recall, the more the system is accurate. Unfortunately, exploratory search tasks are typically associated with undefined and uncertain goals [57]. For example, there are a myriad of correct answers that a user can find for provid-

<sup>32</sup>“r” stands for relevance, “u” stands for unexpectedness. Subscripts indicate the iteration.



(a) Learnability scores.

Comparison	Used as first tool	Used as second tool	Average
Aemoo/RelFinder	$p > 0.1$	$p > 0.1$	$p > 0.1$
Aemoo/Google	$p > 0.1$	$p > 0.1$	$p > 0.1$
Google/RelFinder	$p > 0.1$	$p > 0.1$	$p > 0.1$

(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.

Comparison	First iteration only	Second iteration only	First and second iterations
Aemoo/RelFinder	$0.01 < p \leq 0.05$	$0.05 < p \leq 0.1$	$p < 0.01$
Aemoo/Google	$p > 0.1$	$p > 0.1$	$p > 0.1$
Google/RelFinder	$0.01 < p \leq 0.05$	$0.01 < p \leq 0.05$	$p < 0.01$

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

Fig. 10. Learnability (cf. Figure 10(a)) and Usability (cf. Figure 10(c)) values with corresponding  $p$ -values (cf. Figures 10(b) and 10(d)) measuring significance. Standard deviation values are expressed between brackets and shown as black vertical lines in the chart.

ing a summary about the topic “Alan Turing” (cf. Task 1). It is nearly impossible to classify all correct and wrong answers in order to use precision and recall ap-

propriately. Additionally, we are not interested in the accuracy of results as much as we are in evaluating the

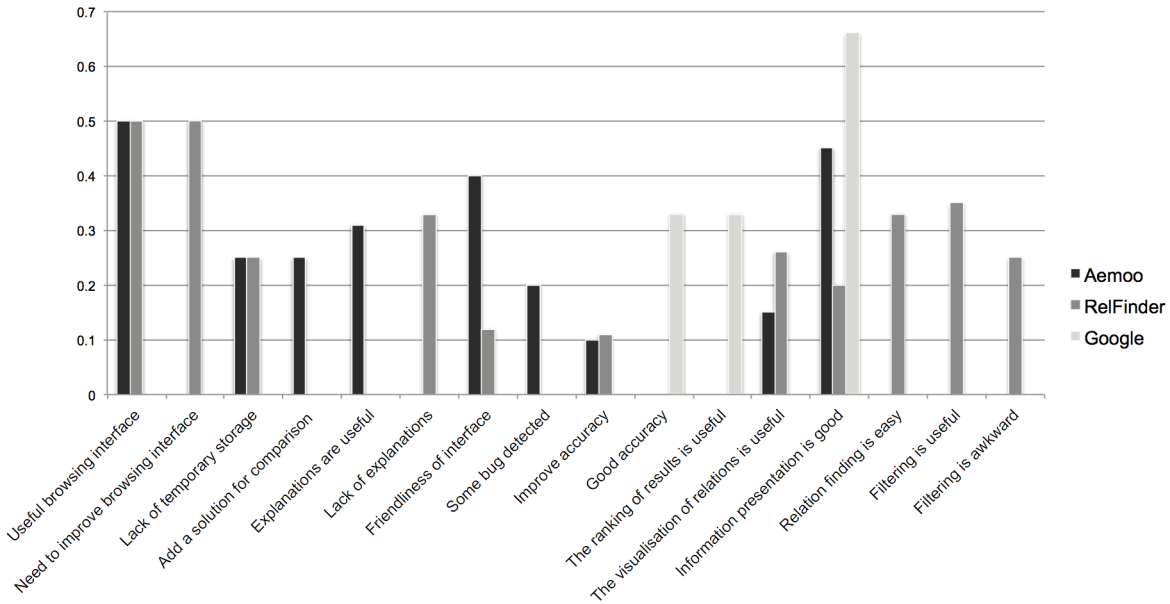


Fig. 11. A chart of the most mentioned pros and cons in the open questionnaires..

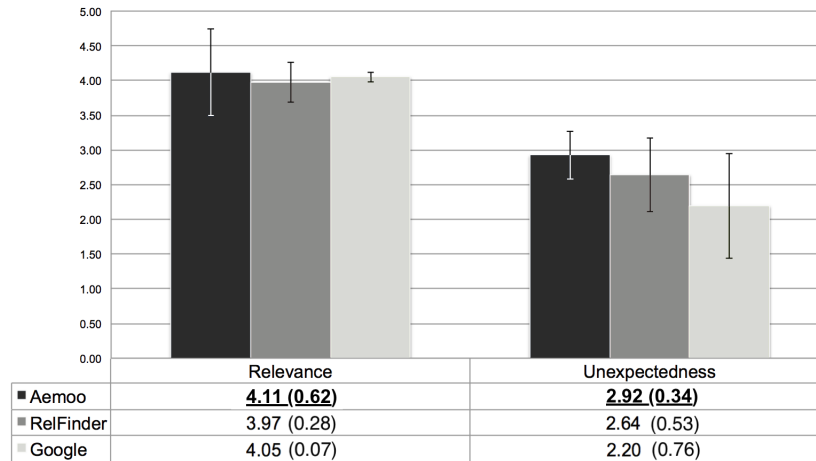


Fig. 12. Relevance and unexpectedness ratings (and their standard deviations between brackets) according to participants' feedback. .

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 (i) to be not Semantic Web experts, (ii) to be unfamiliar with the tools (with the exception of Google), and (iii) to be unfamiliar with the subjects of the tasks (cf. Figure 7). 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 aver-

agely usable, which is a satisfactory result especially considering the comparison with RelFinder usability. 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 detail, the main cons reported by users, which 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. This hypothesis is supported by the fact that we received a considerable number of positive comments from the grounded analysis related to the relation finding mechanism of RelFinder (cf. the entry "Relation finding is easy" in Figure 11). 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 into an added value to 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 7), 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 11). 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* [50]. 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 7) 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 [54,26,27] that leverage the semantic relations asserted in the linked datasets without applying any criterion for defining a boundary, which could be used for tailoring or contextualizing knowledge. For example, RelFinder [26] provides the visualisation of existing relations between two or 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 lack of knowledge boundary, or the heterogeneity problem, for summarising, recommending or browsing Linked Data. Many of them present novel approaches (see below for a quick overview), however to the best of our knowledge, none of them leverages ontology patterns (i.e., EKPs), which is the main contribution that distinguishes our approach from existing ones.

**Entity summarisation.** In [51] a diversity-aware algorithm, called DIVERSUM, is presented. It 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 highest 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 [10] 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 [52] 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** [14] 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 [14], Aemoo can be considered a visualisation-based approach to explore Linked Data. Similarly to existing solutions, such as [5,1,26,24,6,41,55], Aemoo is designed to enable the consumption of Linked Data by lay users, by providing visualisation, filtering, data overview, data merging, and browsing capabilities in order to reduce the cognitive load and support humans in knowledge ex-

ploration and discovery (cf. Section 3). Differently from systems like [5,26,55], 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 reflect the way things are intuitively connected by humans in Wikipedia pages.

Other systems that rely on faceted browsing for filtering results include Yovisto [56], which is a platform that provides exploratory capabilities specialised in academic lecture recordings and conference talks, and Visor [43]. Visor 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 [31], LED [33] 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 is in general a well known visualisation metaphor in literature [17,24,2,25]. Aemoo uses the radial visualisation for rendering concept maps deriving from EKPs that provide a cognitive sound and an intuitive way for representing knowledge [36]. There are systems that allow to build graphical concept maps, such as Cmap<sup>33</sup> [35], nevertheless they are not directly comparable with Aemoo as they aim at supporting users in designing and editing concept maps, which is different from the task addressed by Aemoo (i.e. exploratory search and entity summarisation).

EKPs can be compared to Fresnel lenses [42] that provide a solution for defining implementation-independent templates for data presentation and are used by some state of the art systems, e.g., DBpedia Mobile [5], Marble<sup>34</sup> or IsaViz [41]. However, EKPs were conceived as units of meaning to be used

<sup>33</sup><http://cmap.ihmc.us>

<sup>34</sup><http://mes.github.io/marbles/>



not only for data presentation, but more in general, for data exchange. EKPs also provide a relevance criterion for automatically filtering data, while Fresnel lenses only support 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 that support customised presentation interfaces for EKP-based filtered data.

**Recommending Linked Data.** To some extent, the EKP-based filtering of Aemoo can be abstracted as a recommendation mechanism. Although recommending systems are not directly comparable to Aemoo, for the sake of completeness we report here some relevant work in this area.

MORE [16] leverages DBpedia, Freebase and Linked-MDB in order to recommend movies. It computes similarities between movies thanks to an adaptation of the Vector Space Model (VSM). Seevl [39] is a recommendation system that provides personalised 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 [28] 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 which uses Encyclopedic Knowledge Patterns (EKPs) as relevance criteria for selecting, organising, and visualising knowledge. EKPs were discovered by mining the linking structure of Wikipedia. A system called Aemoo has been 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) EKPs provide a unifying view as well as a relevance criterion for building entity-centric summaries and (ii) they can be exploited effectively for helping humans in exploratory search tasks.

Both hypotheses were validated by means of controlled, task-driven user experiments aimed at assess-

ing the usability of Aemoo, and its 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 very good results by designing a novel Open Knowledge Extraction (OKE) approach and its implementation, called Legalo<sup>35</sup> [45], 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 which allows users to save the summary data of their exploration in RDF;
- integrating the EKP-based approach with user profiles for boundary creation.

## References

- [1] Auer, S., Doehring, R., Dietzold, S., 2010. LESS - Template-Based Syndication and Presentation of Linked Data. In: Aroyo, L., Antoniou, G., Hyvönen, E., ten Teije, A., Stuckenschmidt, H., Cabral, L., Tudorache, T. (Eds.), *Proceedings of the 7th Extended Semantic Web Conference - The Semantic Web: Research and Applications (ESWC 2010)*. Vol. 6089 of *Lecture Notes in Computer Science*. Springer, Berlin, Heidelberg, pp. 211–224, DOI: 10.1007/978-3-642-13489-0\_15.
- [2] Baldassarre, C., Daga, E., Gangemi, A., Gliozzo, A. M., Salvati, A., Troiani, G., 2010. Semantic scout: Making sense of organizational knowledge. In: Cimiano, P., Pinto, H. S. (Eds.), *Proceedings of the 11th International Conference on Knowledge Engineering and Management by the Masses (EKAW 2010)*. Vol. 6317 of *Lecture Notes in Computer Science*. Springer, Berlin, Heidelberg, pp. 272–286, DOI: 10.1007/978-3-642-16438-5\_19.
- [3] Barker, K., Porter, B., Clark, P., 2001. A Library of Generic Concepts for Composing Knowledge Bases. In: Gil, Y., Musen, M., Shavlik, J. (Eds.), *Proceedings of the 1st International Conference on Knowledge Capture (K-Cap 1997)*. ACM, New York, NY, USA, pp. 14–21, DOI: 10.1145/500737.500744.

<sup>35</sup><http://wit.istc.cnr.it/stlab-tools/legalo>

- [4] Barsalou, L. W., 1999. Perceptual symbol systems. *Behavioral and Brain Sciences* 22 (04), 577–660, DOI: 10.1017/S0140525X99002149.
- [5] Becker, C., Bizer, C., 2008. DBpedia Mobile: A Location-Enabled Linked Data Browser. In: Bizer, C., Heath, T., Idehen, K., Berners-Lee, T. (Eds.), *Proceedings of the 1st Workshop on Linked Data on the Web (LDOW 2008)*. Vol. 369 of *CEUR Workshop Proceedings*. CEUR-WS.org, Aachen, Germany, pp. 81–82.
- [6] Berners-Lee, T., Chen, Y., Chilton, L., Connolly, D., Dhanaraj, R., Hollenbach, J., Lerer, A., Sheets, D., 2006. Tabulator: Exploring and Analyzing linked data on the Semantic Web. In: *Proceedings of the 3rd International Semantic Web User Interaction Workshop*. Vol. 2006. pp. 1–16.
- [7] Berners-Lee, T., Hendler, J., Lassila, O., May 2001. The Semantic Web. *Scientific American* 284 (5), 34–43, DOI: 10.1038/scientificamerican0501-34.
- [8] Brooke, J., 1996. SUS - A quick and dirty usability scale. *Usability evaluation in industry* 189 (194), 4–7.
- [9] Burch, M., Konevtsova, N., Heinrich, J., Hoeflerlin, M., Weiskopf, D., Dec 2011. Evaluation of Traditional, Orthogonal, and Radial Tree Diagrams by an Eye Tracking Study. *Visualization and Computer Graphics, IEEE Transactions on* 17 (12), 2440–2448, DOI: 10.1109/TVCG.2011.193.
- [10] Cheng, G., Tran, T., Qu, Y., 2011. RELIN: Relatedness and Informativeness-Based Centrality for Entity Summarization. In: Aroyo, L., Welty, C., Alani, H., Taylor, J., Bernstein, A., Kagal, L., Noy, N., Blomqvist, E. (Eds.), *Proceedings of the 10th International Semantic Web Conference (1)*. Vol. 7031 of *Lecture Notes in Computer Science*. Springer, Berlin, Heidelberg, pp. 114–129, DOI: 10.1007/978-3-642-25073-6\_8.
- [11] Clark, P., Thompson, J., Porter, B., 2000. Knowledge Patterns. In: Cohn, A. G., Giunchiglia, F., Selman, B. (Eds.), *Proceedings of the 7th International Conference on Principles of Knowledge Representation and Reasoning (KR 2000)*. Morgan Kaufmann, San Francisco, CA, USA, pp. 591–600.
- [12] Constantin, A., Peroni, S., Pettifer, S., Shotton, D., Vitali, F., 2015. The Document Components Ontology (DoCO). To appear in *Semantic Web – Interoperability, Usability, Applicability*.
- [13] Craik, K., 1943. *The Nature of Explanation*. Cambridge University Press.
- [14] Dadzie, A.-S., Rowe, M., 2011. Approaches to visualising Linked Data: A survey. *Semantic Web* 2 (2), 89–124, DOI: 10.3233/SW-2011-0037.
- [15] Di Iorio, A., Nuzzolese, A. G., Peroni, S., Shotton, D., Vitali, F., 2014. Describing bibliographic references in RDF. In: Castro, A. G., Lange, C., Lord, P., Stevens, R. (Eds.), *Proceedings of 4th Workshop on Semantic Publishing (SePublica 2014)*. Vol. 1155 of *CEUR Workshop Proceedings*. CEUR-WS.org, pp. 41–56.
- [16] Di Noia, T., Mirizzi, R., Ostuni, V. C., Romito, D., Zanker, M., 2012. Linked open data to support content-based recommender systems. In: Presutti, V., Pinto, H. S. (Eds.), *Proceedings of the 8th International Conference on Semantic Systems (I-SEMANTICS 2012)*. ACM, New York, NY, USA, pp. 1–8, DOI: 10.1145/2362499.2362501.
- [17] Draper, G. M., Livnat, Y., Riesenfeld, R. F., 2009. A survey of radial methods for information visualization. *Visualization and Computer Graphics, IEEE Transactions on* 15 (5), 759–776, DOI: 10.1109/TVCG.2009.23.
- [18] Fielding, R. T., 2000. REST: architectural styles and the design of network-based software architectures. Doctoral dissertation, University of California, Irvine.
- [19] Fillmore, C., 1968. The case for the case. In: Bach, E., Harms, R. (Eds.), *Universals in Linguistic Theory*. Rinehart and Winston, New York, pp. 21–119.
- [20] Fillmore, C. J., Oct. 1976. Frame semantics and the nature of language. *Annals of the New York Academy of Sciences* 280 (1), 20–32, DOI: 10.1111/j.1749-6632.1976.tb25467.x.
- [21] Gallese, V., Metzinger, T., 2003. Motor ontology: the representational reality of goals, actions and selves. *Philosophical Psychology* 16 (3), 365–388, DOI: 10.1080/0951508032000121760.
- [22] Gangemi, A., Presutti, V., 2010. Towards a Pattern Science for the Semantic Web. *Semantic Web* 1 (1-2), 61–68, DOI: 10.3233/SW-2010-0020.
- [23] Garlan, D., Monroe, R. T., Wile, D., 2000. Acme: Architectural description of component-based systems. In: Leavens, G. T., Sitaraman, M. (Eds.), *Foundations of Component-Based Systems*. Cambridge University Press, pp. 47–68.
- [24] Goyal, S., Westenthaler, R., 2004. RDF Gravity (RDF Graph Visualization Tool). Salzburg Research, Austria.
- [25] Hastrup, T., Cyganiak, R., Bojars, U., 2008. Browsing Linked Data with Fenfire. In: Bizer, C., Heath, T., Idehen, K., Berners-Lee, T. (Eds.), *Proceedings of the 1st Workshop on Linked Data on the Web (LDOW 2008)*. Vol. 369 of *CEUR Workshop Proceedings*. CEUR-WS.org, Aachen, Germany, pp. 1–2.
- [26] Heim, P., Hellmann, S., Lehmann, J., Lohmann, S., Stegemann, T., 2009. RelFinder: Revealing Relationships in RDF Knowledge Bases. In: Chua, T.-S., Kompatsiaris, Y., Mérialdo, B., Haas, W., Thallinger, G., Bailer, W. (Eds.), *Proceedings of the 3rd International Conference on Semantic and Media Technologies (SAMT)*. Vol. 5887 of *Lecture Notes in Computer Science*. Springer, Berlin, Heidelberg, pp. 182–187, DOI: 10.1007/978-3-642-10543-2\_21.
- [27] Heim, P., Ziegler, J., Lohmann, S., 2008. gFacet: A Browser for the Web of Data. In: Auer, S., Dietzold, S., Lohmann, S., Ziegler, J. (Eds.), *Proceedings of the International Workshop on Interacting with Multimedia Content in the Social Semantic Web (IMC-SSW 2008)*. Vol. 417 of *CEUR Workshop Proceedings*. CEUR-WS, Aachen, Germany, pp. 49–58.
- [28] Kaminskis, M., Fernández-Tobías, I., Ricci, F., Cantador, I., 2012. Knowledge-based Music Retrieval for Places of Interest. In: Liem, C. C., Müller, M., Tjoa, S. K., Tzanetakis, G. (Eds.), *Proceedings of the 2nd International Workshop on Music Information Retrieval with User-Centered and Multimodal Strategies (MIRUM 2012)*. ACM, New York, NY, USA, pp. 19–24, DOI: 10.1145/2390848.2390854.
- [29] Lewis, J. R., Sauro, J., 2009. The Factor Structure of the System Usability Scale. In: Kurosu, M. (Ed.), *Proceedings of the 1st International Conference Human Centered Design (HCD 2009)*. Vol. 5619 of *Lecture Notes in Computer Science*. Springer, Berlin, Heidelberg, pp. 94–103, DOI: 10.1007/978-3-642-02806-9\_12.
- [30] Marchionini, G., Apr. 2006. Exploratory Search: From Finding to Understanding. *Communications of the ACM* 49 (4), 41–46, DOI: 10.1145/1121949.1121979.
- [31] Marie, N., Gandon, F., Ribière, M., Rodio, F., 2013. Discovery Hub: on-the-fly linked data exploratory search. In: Sabou, M., Blomqvist, E., Di Noia, T., Sack, H., Pellegrini, T. (Eds.), *Proceedings of the 9th International Conference on Semantic*

- Systems (I-SEMANTICS 2013). ACM, New York, NY, USA, pp. 17–24, DOI: 10.1145/2506182.2506185.
- [32] Minsky, M., 1975. A Framework for Representing Knowledge. In: Winston, P. (Ed.), *The Psychology of Computer Vision*. McGraw-Hill, pp. 211–281.
- [33] Mirizzi, R., Di Noia, T., 2010. From exploratory search to web search and back. In: Nica, A., Varde, A. S. (Eds.), *Proceedings of the 3rd workshop on Ph.D. Students in Information and Knowledge Management (PIKM 2010)*. ACM, New York, NY, USA, pp. 39–46, DOI: 10.1145/1871902.1871910.
- [34] Novak, J. D., 1977. *A theory of education*. Cornell University Press, Ithaca, N.Y.
- [35] Novak, J. D., Cañas, A. J., 2006. *The Theory Underlying Concept Maps and How to Construct and Use Them*. Research Report 2006-01 Rev 2008-01, Florida Institute for Human and Machine Cognition.
- [36] Novak, J. D., Gowin, D. B., 1984. *Learning how to learn*. Cambridge University Press.
- [37] Nuzzolese, A. G., Gangemi, A., Presutti, V., Ciancarini, P., 2011. Encyclopedic Knowledge Patterns from Wikipedia Links. In: Aroyo, L., Welty, C., Alani, H., Taylor, J., Bernstein, A., Kagal, L., Noy, N., Blomqvist, E. (Eds.), *Proceedings of the 10th International Semantic Web Conference (ISWC 2011)*. Vol. 7031 of Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, pp. 520–536, DOI: 10.1007/978-3-642-25073-6\_33.
- [38] Nuzzolese, A. G., Presutti, V., Gangemi, A., Musetti, A., Ciancarini, P., 2013. Aemoo: exploring knowledge on the Web. In: Davis, H., Halpin, H., Pentland, A., Bernstein, M., Adamic, L., Alani, H., Monnin, A., Rogers, R. (Eds.), *Proceedings of the 5th Annual ACM Web Science Conference*. ACM, New York, NY, USA, pp. 272–275, DOI: 10.1145/2464464.2464519.
- [39] Passant, A., 2010. dbrec – Music Recommendations Using DBpedia. In: Patel-Schneider, P. F., Pan, Y., Hitzler, P., Mika, P., Zhang, L., Pan, J. Z., Horrocks, I., Glimm, B. (Eds.), *Proceedings of the 9th International Semantic Web Conference (ISWC 2010)*. Vol. 6497 of Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, pp. 209–224, DOI: 10.1007/978-3-642-17749-1\_14.
- [40] Peroni, S., Shotton, D., 2012. FaBiO and CiTO: Ontologies for describing bibliographic resources and citations. *Web Semantics: Science, Services and Agents on the World Wide Web* 17, 33–43, DOI: 10.1016/j.websem.2012.08.001.
- [41] Pietriga, E., Oct. 2003. Isaviz: A visual authoring tool for rdf. W3C technical report, W3C, available at <http://www.w3.org/2001/11/IsaViz>.
- [42] Pietriga, E., Bizer, C., Karger, D. R., Lee, R., 2006. Fresnel: A Browser-Independent Presentation Vocabulary for RDF. In: Cruz, I., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., Aroyo, L. M. (Eds.), *Proceedings of the 5th International Semantic Web Conference (ISWC 2006)*. Vol. 4273 of Lecture Notes in Computer Science. pp. 158–171, DOI: 10.1007/11926078\_12.
- [43] Popov, I. O., Schraefel, M. M. C., Hall, W., Shadbolt, N., 2011. Connecting the Dots: A Multi-pivot Approach to Data Exploration. In: Aroyo, L., Welty, C., Alani, H., Taylor, J., Bernstein, A., Kagal, L., Noy, N., Blomqvist, E. (Eds.), *Proceedings of the 10th International Semantic Web Conference (ISWC 2011)*. Vol. 7031 of Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, pp. 553–568, DOI: 10.1007/978-3-642-25073-6\_35.
- [44] Presutti, V., Aroyo, L., Adamou, A., Schopman, B. A. C., Gangemi, A., Schreiber, G., 2011. Extracting core knowledge from linked data. In: Hartig, O., Harth, A., Sequeda, J. (Eds.), *Proceedings of the 2nd International Workshop on Consuming Linked Data (COLD 2011)*. Vol. 782 of CEUR Workshop Proceedings. CEUR-WS.org, Aachen, Germany, pp. 37–48.
- [45] Presutti, V., Consoli, S., Nuzzolese, A. G., Reforgiato Recupero, D., Gangemi, A., Bannour, I., Zargayouna, H., 2014. Uncovering the Semantics of Wikipedia Pagelinks. In: Janowicz, K., Schlobach, S., Lambrix, P., Hyvönen, E. (Eds.), *Proceedings of the 19th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2014)*. Vol. 8876 of Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, pp. 413–428, DOI: 10.1007/978-3-319-13704-9\_32.
- [46] Sauro, J., 2011. *A practical guide to the system usability scale: Background, benchmarks & best practices*. Measuring Usability LCC.
- [47] Shneiderman, B., 1996. The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In: *Proceedings of the IEEE Symposium on Visual Languages*. IEEE, pp. 336–343, DOI: 10.1109/VL.1996.545307.
- [48] Strauss, A., Corbin, J., 1998. *Basics of Qualitative Research: Techniques and Procedures for developing Grounded Theory*. Sage Publications Inc, DOI: 10.4135/9781452230153.
- [49] Suchanek, F., Kasneci, G., Weikum, G., 2008. Yago - A Large Ontology from Wikipedia and WordNet. *Web Semantics: Science, Services and Agents on the World Wide Web* 6 (3), 203–217, DOI: 10.1016/j.websem.2008.06.001.
- [50] Sun, T., Zhang, M., Mei, Q., 2013. Unexpected Relevance: An Empirical Study of Serendipity in Retweets. In: Kiciman, E., Ellison, N. B., Hogan, B., Resnick, P., Soboroff, I. (Eds.), *Proceedings of the 7th International AAAI Conference on Weblogs and Social Media (ICWSM 2013)*. AAAI Press, Palo Alto, CA, USA, pp. 1–10.
- [51] Sydow, M., Pikula, M., Schenkel, R., 2013. The notion of diversity in graphical entity summarisation on semantic knowledge graphs. *Intelligent Information Systems* 41 (2), 109–149, DOI: 10.1007/s10844-013-0239-6.
- [52] Thalhammer, A., Rettinger, A., 2014. Browsing DBpedia Entities with Summaries. In: Presutti, V., Blomqvist, E., Troncy, R., Sack, H., Papadakis, I., Tordai, A. (Eds.), *Proceedings of the 11th Extended Semantic Web Conference (ESWC 2014 - Satellite Events)*. Vol. 8798 of Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, pp. 511–515, DOI: 10.1007/978-3-319-11955-7\_76.
- [53] The OSGi Alliance, Mar. 2010. OSGi Service Platform Release 4 Version 4.2, Enterprise Specification. Committee specification, Open Services Gateway initiative (OSGi).
- [54] Tummarello, G., Cyganiak, R., Catasta, M., Danielczyk, S., Delbru, R., Decker, S., 2010. Sig.ma: Live views on the web of data. *Web Semantics: Science, Services and Agents on the World Wide Web* 8 (4), 355–364, DOI: 10.1016/j.websem.2010.08.003.
- [55] Tvarozek, M., Bieliková, M., 2010. Factic: Personalized Exploratory Search in the Semantic Web. In: Benatallah, B., Casati, F., Kappel, G., Rossi, G. (Eds.), *Proceedings of the 10th International Conference on Web Engineering (ICWE 2010)*. Vol. 6189 of Lecture Notes in Computer Science. Springer, pp. 527–530, DOI: 10.1007/978-3-642-13911-6\_44.

- [56] Waitelonis, J., Sack, H., 2012. Towards exploratory video search using linked data. *Multimedia Tools and Applications* 59 (2), 645–672, DOI: 10.1007/s11042-011-0733-1.
- [57] White, R. W., Kules, B., Bederson, B. B., 2005. Exploratory search interfaces: categorization, clustering and beyond: report on the XSI 2005 workshop at the Human-Computer Interaction Laboratory, University of Maryland. *SIGIR Forum* 39 (2), 52–56, DOI: 10.1145/1113343.1113356.