Published as: Waitelonis, J.; Ludwig, N.; Knuth, M. & Sack, H. WhoKnows? - Evaluating Linked Data Heuristics with a Quiz that Cleans Up DBpedia. International Journal of Interactive Technology and Smart Education (ITSE), Emerald, 2011, 8.

WhoKnows? - Evaluating Linked Data Heuristics with a Quiz that Cleans Up DBpedia

Jörg Waitelonis, Nadine Ludwig, Magnus Knuth, and Harald Sack

Hasso-Plattner-Institut für Softwaresystemtechnik GmbH,
Prof.-Dr.-Helmert-Str. 2–3, 14482 Potsdam, Germany
{joerg.waitelonis,nadine.ludwig,
magnus.knuth,harald.sack}@hpi.uni-potsdam.de
http://www.hpi.uni-potsdam.de

Abstract. Semantic technologies enable sophisticated search scenarios on educational video content. Linking Open Data (LOD) provides a vast amount of well structured semantic information in heterogenous domains. But, despite of the syntactically well expressed RDF facts, when authoring and publishing LOD, many inconsistencies may occur, especially if the data is generated with the help of automated methods. Data cleansing approaches enable to detect inconsistencies and to overhaul affected data sets, but they are difficult to apply automatically. Also manual reviewing is hardly possible because of the tremendous size of already existing LOD resources. To overcome this, we propose a collaborative reviewing approach, which motivates the reviewer in a playful manner. This paper presents *WhoKnows?* , an online quiz that generates different kinds of questionnaires from DBpedia data sets. Besides its playfulness, *WhoKnows?* has been developed for the evaluation of property relevance ranking heuristics on DBpedia data with the convenient side effect of detecting inconsistencies and doubtful facts.

Key words: DBpedia, data cleansing, serious games, evaluation

1 Introduction

Lecture video portals like Yovisto[1] offer students the possibility to catch up on missed lectures or enhance their knowledge by watching lectures from other universities or scientific conference presentations. In order to become a fully-fledged and efficient eLearning tool sophisticated search methods for the video archive are essential. Semantic search and in particular exploratory search on educational videos enables students to find related videos regarding the content [1]. By this means the study of a research field could be complemented with related video material that can be found with the help of semantic annotations. E. g., a video about the physicist Alfred Kleiner might also be relevant for students, who are interested in Albert Einstein's relativity theory, because the research fields of both physicists are similar. Related entities can be determined by shared

[1] http://www.yovisto.com

categories and linking properties, e. g. Alfred Kleiner and Albert Einstein are related to each other because they both were *Swiss physicists* and Albert Einstein was a *doctoral student* of Alfred Kleiner. These semantic facts can be gained from DBpedia[2] - a semantic extraction of the online encyclopedia Wikipedia[3]. Currently, DBpedia consists of 286 million facts for about 3.5 million entities extracted from the English Wikipedia version. This results in a huge amount of facts for every entity. For a given entity some of these facts are more relevant than others. Besides scientific facts for Albert Einstein like academic advisors or alma maters, DBpedia also contains the birth place and residence, ethnicity or children. These facts (respectively the properties in question) have to be ordered according to their importance to find most relevant related and therefore interesting entities.

In general, properties of a given resource are all equally important, i. e. there is no given ranking that defines the relevance of a certain fact. Anyway, it would be important to separate significant facts from circumstantial facts. We have already addressed this problem in previous work by developing property ranking heuristics that enable an exploratory search experience.

One of the major problems that we had experienced was the lack of a sound evaluation for these property ranking heuristics. There is no objective truth for the importance of a fact. Context, pragmatics, as well as personal point of view determine what a single user considers to be of importance. Thus, a qualitative evaluation has to be implemented. Of course, the obvious solution is to determine a number of test users and to present them a set of preselected resources and all of their properties with the task to rank the properties according to their importance. Unfortunately, this is a rather tedious task and usually the test user gets bored after a few selections. However, for a proper evaluation we require a sufficient large number of users to perform this relevance ranking for a number of resources as large as possible.

For this reason, we decided to develop our little quiz game *WhoKnows?* to accomplish this task. Facts are transformed into questionaires and the user can score points according to her performance in the quiz as explained in Section 3. In the scope of semantic web technologies and linked data several other approaches of "games with a purpose" try to tackle similar problems as shown in Section 2. In order to use the viral effects of games in social networks, we integrated the game as Facebook application[4]. But, *WhoKnows?* can also be played as standalone game[5].

While playing the quiz and communicating with the players about their complaints, we have realized soon that our quiz had a very remarkable side effect. People have been complaining about inconsistent, ambiguous, confusing, or conflicting questions among the quiz. By further investigation, we have been able to determine a few mistakes caused by our implementation. But, the majority

[2] `http://dbpedia.org/`
[3] `http://www.wikipedia.org`
[4] `http://apps.facebook.com/whoknows_/`
[5] `http://www.yovisto.com/whoknows`

of problematic cases originated from flaws and inconsistencies of the underlying Linking Open Data (LOD) resources[6] themselves as will be demonstrated in Section 4. A brief outline of possible solutions how to cleanup the detected flaws concludes the paper. Consequently, our little quiz game has proven useful in two ways: First, to evaluate semantic property heuristics, and second, to detect flaws and inconsistencies in LOD resources.

2 Related Work

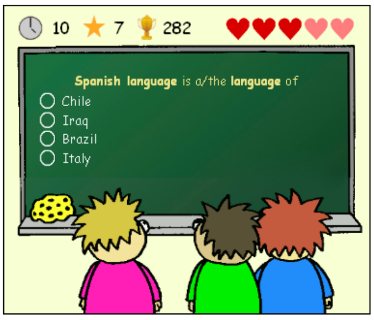
Linked Data [2] has moved over to the focus of various web applications [3] to enrich data and enable a semantic view on the web, as e. g., the prominent BBC music platform[7]. But, despite the non-disputed advantages of this huge amount of data available, the inconsistencies resulting from automatic data generation and the lack of a valid evaluation can not be solved manually in an efficient way. An obvious approach would be to draw the wisdom of the crowds [4] on this problem and thereby use the human intelligence to overcome a computer-generated inconvenience. This approach assumes that an objective evaluation of the relevance of an object is best performed by gathering the opinion of many independently interviewed people, which are provided with the same information and facts. The most challenging problem to achieve this evaluation is to draw people's attention on this research issue and to motivate them to contribute their knowledge. Obviously the development of games with a purpose [5] seems essential for solving this problem. As pioneer games for this approach the *ESP Game* [6], and *Phetch* [7] for annotating images, *Peekaboom* [8] for localizing objects in images, and *Verbosity* [9] for gathering common sense facts have to be referenced.

Following this idea and putting it in the context of the Semantic Web the *OntoGame platform* provides a generic game infrastructure to develop various types of multi-user games with only minimal modifications. A team consisting of two players has to reach a consensus in order to earn credits for the task. The users' input and results are stored as RDF data for a later analysis and reuse [10].

Also designed as multi-player game, the *Listen Game* aims at measuring the semantic relationship between music songs and words. To describe a given song, the game provides a "normal mode", where the user has to pick the best and worst characterizing word from a predefined 174-word vocabulary, and a "freestyle mode", where the user is asked to contribute a new descriptive term. In both cases the user receives immediate feedback on what all other users have chosen. The score is then calculated from the amount of agreement between the user's and all other players' choices [11].

All referred games are following a multi-player approach. This requires two players to be online and willing to play against or with each other at the same time. *WhoKnows?* avoids this potential difficulty by enabling users to play the

[6] `http://linkeddata.org`
[7] `http://www.bbc.co.uk/music`



Subject	Property	Object
Chile	**language**	**Spanish language**
Iraq	language	Arabic language
Brazil	language	Portuguese language
Italy	language	Italian language

Fig. 1. Screenshot and triples used to generate a One-To-One question.

game on their own whenever they want to. Thereby, we assume to increase the motivation of a user and thus, to obtain a larger amount of user-generated data.

WhoKnows? is designed as game for in-between times integrated in a social network. Members of such networks differ in many characteristical attributes like age, gender, origin, social background, which is a highly appreciated precondition for a huge diversity of opinion. Also, since each player is interested in achieving high scores, players don't share correct answers, and thus, knowledge about facts in the game will not be distributed. This ensures an independence of opinions. By facing the challenges of consistently new and more difficult questions the users are motivated to develop special local knowledge - a decentralization of knowledge evolves. At the end all resulting data is stored in a database for further analysis and turned into a collective decision. All these criteria are requirements to ensure an optimal output using the wisdom of the crowds [4]. This makes *WhoKnows?* a significant approach to evaluate property relevance ranking heuristics, and as a side effect to detect inconsistencies and doubtful facts in LOD resources.

3 *WhoKnows?* - Concept and Realization

This section deals with the concept and implementation of the *WhoKnows?* quiz game. First, the game concept and graphical realization followed by the system's architecture are presented. Afterwards, preprocessing of the data as well as the generation of the questionnaires are described in detail. Finally, the evaluation of game rounds and how inconsistencies in DBpedia can be revealed are discussed.

3.1 Game Concept and Graphical Interface

The game is based on the principle to present questions to the user that have been generated out of facts stemming from DBpedia RDF triples. As an example, Fig. 1 shows the question *'Spanish language is the language of ...?'* with correct answer *'Chile'*. The question originates from the triple

```
dbp:Chile dbpprop:language dbp:Spanish_language .
```

and is composed by turning the order upside down:

`Object is the property of: subject1, subject2, subject3...`

Fig. 1 also shows the triples for the remaining choices. In addition, false answers 'Iraq', 'Brazil', and 'Italy' are randomly selected from other triples meeting the requirement that the triples' subjects belong to the same or a similar category. It is important to ensure that the selected false answers are really wrong (c. f. Sect. 3.3).

To add variety and to increase the user's motivation, the game is designed with different game variants:

- **One-To-One:** only one answer is correct,
- **One-To-N:** one or more answers are correct,
- the **Hangman** game asks to fill the correct answer in a cloze.

While playing the game, the variants are used alternately. Different background and game specific button-shapes enable a preattentive recognition of the particular game type. After selecting the answer, the user immediately receives feedback about the correctness of her choice by using an appropriate color scheme (green for correct, red for wrong) and showing happy resp. unhappy emotions of the game mascots. If the user has the opinion a question is somehow odd or strange a 'Dislike'-button enables to mark this question as an potentially inconsistent, ambiguous, confusing, or conflicting question.

3.2 Preprocessing of DBpedia Data

As of January 2011, the DBpedia data set describes 3.5 million 'entites' with over half a billion 'facts'[8]. But, not all information is useful in the context of the *WhoKnows?* game. The relevant data sets for the game are chosen from the English language version of DBpedia and include:

- *Ontology Infobox Properties*, contains properties of the entities.
- *Titles*, contains labels of resources.
- *Article Categories*, contains resources and their direct categories.
- *Categories (SKOS* [9]*)*, contains categories and category hierarchies.
- *Pagelinks*, contains referencing links between the underlying Wikipedia-articles.

The game was primarily designed with the purpose of evaluating property rankings for DBpedia resources in the context of exploratory video search with Yovisto[10] as described in [1]. Therefore, the huge set of available entities has been restricted only to those, which are relevant for the underlying video search engine Yovisto. Based on this data set, further processing filters irrelevant information and deduces implicit knowledge to speed up the question generation. Because of the enormous amount of data, the processing has been parallelized

[8] `http://wiki.dbpedia.org/Datasets`
[9] `http://www.w3.org/2004/02/skos/`
[10] `http://www.yovisto.com/`

on a hadoop[11] cluster to achieve results in a reasonable time frame. This preprocessing comprises the following steps:

Filtering of irrelevant triples The *Ontology Infobox Properties* contains the essential triples for question generation. Some triples contain very long literals, which would result in an improper presentation. If the triples are too specific (as e. g., the resolution of a movie or the exact number of population of a city), they can cause a misleading comprehension of questions. Furthermore, only those triples are to be processed, whose object and subject are part of the list of preselected Yovisto entities. Other facts are not relevant for our purpose, because the Yovisto search engine does not provide results related to them. In this way the number of triples can be reduced from originally 11.1 million to only 2.1 million.

Distinct sets of resources The output of the previous step in combination with the *Article Categories* data set enables to restrict processing to a distinct set of entities, properties, and categories, only comprising the relevant resources. This eases the database import and speeds up further processing. The process results in 0.7 million entities, 516 different properties, and 240,000 categories.

Assignment of entities to superordinate categories To assign categories to the entities that are necessary to generate *wrong* answers, the data sets and *Categories (SKOS)* are combined. The categorization is based on SKOS, because the DBpedia categorization schema is incomplete and to some extend even wrong. In *Article Categories* the entities are directly assigned to categories, which sometimes are too specific. The problem is that almost all entities of this specific category can be considered as a correct answer, and hence, no false answer can be chosen. For example, all persons from the category 'English Entertainers' speak English language. Therefore, no person can be selected, who does not speak English. A more general superordinate category 'European Actors' comprises also non-English speaking persons and therefore is a better choice to select false answers. Another problem lies in the fact that we have to find the appropriate superordinate category. For this purpose the transitive closure of categories has to be computed. Thereby, entities can be directly assigned to their corresponding superordinate categories. To speed up the process too general categories, such as *Person*, *Place*, *Organization*, or *Plant*, and too specific categories, such as *Space Shuttle*, or *Inline Hockey League*, are excluded. Only those categories are taken into account that comprise at least 500 but not more than 15,000 entities. This filter process reduces the number of relevant categories from originally 240,000 to only 1,200.

Weighting of Entities To ensure that the degree of difficulty raises with progressing game level, entities are weighted according to a degree of difficulty. To

[11] `http://hadoop.apache.org/`

obtain the weight of an entity we estimate its degree of popularity. We assume that the more popular an entity is, the easier a question can be answered. As a basis for this approach, we use the Wikipedia link graph derived from the *Pagelinks* data set. To calculate the popularity, we utilize the Google PageRank algorithm [12]: The more links exist, the more popular is the linked entity. In contrast to the original PageRank algorithm, we do not take the weight of the linking resource into account.

Restricting the candidates for questions only to those facts that achieve a certain level of link popularity is contradicting our original goal of evaluating property heuristics. But, while playing the game, we realized that it was important to narrow the choice for questions, because otherwise the user was confronted with a huge number of barely solvable questions.

3.3 Generating the Questionnaire

For reasons of efficiency, the generation of questions is performed only once offline. The actual selection of questions is executed online on request. The process of query generation comprises three steps:

- weighting of triples and properties,
- assigning the degree of difficulty to triples, and
- generating false answers.

The weight of a triple t is computed from the weights of its subjects $t.s$ and objects $t.o$ to

$$weight(t) = \frac{44 \cdot weight(t.s) + 1 \cdot weight(t.o)}{45}. \tag{1}$$

For generating a question, the original RDF triple representing a fact is turned upside-down, i. e. the question asks for a subject to complement the given property and object. Because the subject is chosen by the player, its impact on $weight(t)$ has been raised with the empirically determined factor 44. For weighting the properties we assume that the more popular the subjects and objects of a RDF triple are, the more important is the property itself. The weight of a property p is computed to

$$weight(p) = \frac{winsor_5(weight(s_i)) + winsor_5(weight(o_j))}{2}, \tag{2}$$

with s_i the subjects appearing with p, o_j the objects appearing with p, and $winsor_5()$ the winsorized mean to exclude outlier.

Assigning the degree of difficulty to a triple differs for distinct game types. For example, for the One-To-One games the degree is directly derived from the weight of triples. Given a list of triples ordered by their weight, we define the first n entries to be on level 1 difficulty, the next n entries to be on level 2, and so on. We have assigned $n = 75$.

The preprocessing of false answers is based on the generation of a set of wrong subjects for every triple. When generating a question, a set of false answers is

chosen from its corresponding set by random. To ensure that the false answer fits the specific topic of the question, wrong subjects are selected from the same category as the correct subject. Because every triple (s, p, o) belongs to one or more categories, the best matching subject category C_s needs to be determined. This is achieved by selecting the category comprising the most subjects. If there are categories with the same number of subjects, the most specific category will be selected. Finally, for every triple (s, p, o) the wrong subjects are selected from C_s.

3.4 Evaluating Game Rounds

After the player has finished answering a question the subsequent evaluation comprises three important aspects: recalculating the *game state*, which includes scoring, the number of remaining virtual lives, and the level for the next round, updating *game statistics*, and the possibility to *review the question*.

Game State Scoring allows to measure and to compare the players' achievements. A player can easily judge how well she performed in *WhoKnows?* compared to her friends, which motivates people to play over and over again in order to achieve a better score.

Scorings for a single round can be positive or negative, a player can loose lives, and move up or down one level of the game. To calculate the achieved score, given answers are matched against the facts from DBpedia, which are assumed to be true. Scores are static and not changed afterwards, even though facts in DBpedia may be subsequently changed.

Incorrect answers will be penalized by loosing one life and a predefined number of points. In addition, the player moves one level down, to adapt the degree of difficulty to the user's abilities. In contrast, correct answers raise the level of difficulty. Questions have to be solved within limited time frame, the expiration of the time limit costs 30 points penalty and one life. For correct answers the player earns 20 points, for false answers 20 points will be deducted. The higher penalty for not answering the question should motivate the player to decide for an answer before time is up.

Reviewing the Question The player has the possibility to verify the result of a game round and, if necessary to submit a complaint. These functions had to be provided, because it turned out that sometimes data is incorrect or incomplete. Submitting an answer is followed by presenting the correct solution to the player, enriched with additional background information to verify the correct answer or to obtain further information provided by DBpedia or Wikipedia.

In case the player has the opinion that the requested fact seems to be inconsistent, ambiguous, confusing or even contradicting a reporting option in the form of a 'Dislike'-button has been provided. Whenever the player dislikes a question, all related information is stored to reconstruct the doubtful question.

This allows to detect and possibly to correct singular and structural inconsistencies. An analysis of those doubtful facts/questions is described in the next section.

4 Results

WhoKnows? was originally designed to evaluate the ranking heuristics proposed in [1]. The heuristics are based on the RDF graph structure and use statistical arguments to rank RDF properties according to their relevance. By the addition of the 'Dislike'-button potential inconsistencies in DBpedia can be revealed.

At present, the game has been played 781 times by 165 different users. In total 4,051 distinct triples have been played. Overall 18,488 rounds have been performed of which 13,404 have been answered correctly.

4.1 Evaluation of Property Ranking

For evaluating the property ranking heuristics we need to determine the perceived importance of single properties. Our evaluation underlies the assumption that, the more often a question is answered correctly, the more likely it is that the fact seems to be of importance. Whenever a distinct property for different subjects and objects occurs frequently among the correct answers, it can be regarded as an important property. *WhoKnows?* keeps track for every triple how often it has been played and how often it has been matched correctly.

Table 1. Property Ranking for DBpedia Ontology classes *Organisation* (left) and *Company* (right)

Rank	Property	Correct Answers
1	locationCountry	97.43%
2	league	94.73%
3	meetingCity	94.64%
4	team	91.45%
5	leader	89.47%
6	keyPerson	88.88%
	...	
13	parentCompany	82.35%
	...	
15	location	81.25%
16	country	81.10%
17	product	72.41%
	...	
21	service	45.45%

Rank	Property	Correct Answers
1	locationCountry	93.75%
2	keyPerson	88.88%
3	parentCompany	82.35%
4	location	81.25%
5	product	72.41%
6	country	64.70%
7	service	45.45%

Using this data we are able to rank DBpedia properties according to their ratio, how often a question with this property has been answered correctly. Properties should be ranked within the categories their entities are assigned to in an

ontology. Because a property which is very important for an entity of one category can be very unimportant for an entity of a another category. Table 1 shows the ranking of the properties of the two categories *Company* and *Organisation*. Entities assigned to the category *Organisation* are mostly sports clubs, institutes, universities etc. Apparently even for these very similar categories the property ranking differs. E. g., the keyPerson is ranked more important for a *Company* than for an *Organisation*.

According to this heuristic we can recommend a selection and ordering of likely relevant properties for an entity. As, e. g., for the company *Ford*[12] the following properties are considered to be important as shown in Table 2. Since we observed only object properties and filtered the data set according to our purpose in advance (c. f. Sect. 3.2), there may be some relevant properties missing (e. g. dbpprop:foundationPerson, dbpprop:industry).

Table 2. Recommended property ordering for dbp:Ford_Motor_Company.

Property (Rank)	Object Value
locationCountry	United States
keyPerson	William Clay Ford, Jr. Alan Mulally
product	Automobile
service	Vehicle leasing Service (motor vehicle) Finance

Possible error sources are widespread: our initial selection of triples can be seen as quite restrictive, since we observed only triples concerning Yovisto entities. By assigning triples to different levels of difficulty a number of triples have not been played yet, because the level was not reached[13]. Up to now only 188 distinct properties (from 508 properties in the data set) have been played at least once. Some properties have been played very often, e. g. governmentType has been played 3472 times. Thus, players might have learned the correct answer already, which influences the rating.

Some properties are prone for trivial questions, because subject and object often contain parts of words of each other, which oversimplifies the task.

4.2 Detection of Inconsistencies

For the detection of inconsistencies we had to analyze all games that have been marked by the 'Dislike'-button. The 'Dislike'-button was used 470 times (One-To-One, 274; Hangman, 125; One-To-N, 71). The following issues have been identified:

[12] `http://dbpedia.org/resource/Ford_Motor_Company`
[13] so far the highest reached level is 31

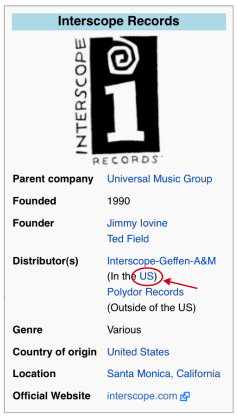


Fig. 2. Screenshot of an info box in Wikipedia with inconsistent linking.

The question was too simple or too difficult. In some cases users have had the opinion that the question was too trivial or too simple. This can happen, if the correct answer and the question itself contain almost identical terms, as e. g., (Austrians, isEthnicGroupOf, Austria).

Game Bugs. The 'Dislike'-button has been used 77 times to mark programming errors, which have been fixed in subsequent releases.

Inconsistencies. The remaining dislikes refer to real inconsistencies in DBpedia. These problems are caused either during automated data extraction from Wikipedia, or by false statements within Wikipedia articles. Most DBpedia data has been extracted from the Wikipedia info boxes. Typically, the inconsistencies occur, if Wikipedia info boxes are not well structured. The info boxes should contain only key-value pairs, which can be extracted in a reliable way.

Fig. 2 shows an example of an info box with an inappropriate linking (links are underlined for better reading). The 'Distributor(s)' values contain a link to 'United States', which produces an obviously incorrect triple (Interscope Records, distributingCompany, United States).

With the 4,051 triples played 342 triples have been identified as potentially inconsistent. After removing the programming errors, within the resulting 265 triples 121 have been verified as true inconsistencies, i. e. with *WhoKnows?* we have been able to create a reduced data set with a ratio of around 46% wrong triples. This is a reasonable number to be reviewed manually in order to correct the affected triples.

5 Summary and Outlook on Future Work

Based on these improved semantic metadata, we are able to evaluate our property ranking heuristics and thereby improve semantic recommendations. Also, eLearning platforms, such as our video lecture portal Yovisto, can achieve higher quality and efficiency.

The original purpose we developed *WhoKnows?* for, was to evaluate heuristics to rank LOD properties and thus, obtain a semantic relatedness between entities according to the properties they are linked by. The presented approach is an efficient method to detect popular properties within a limited amount of triples. But, since we had to constrain the number of triples to one-fifth of the original DBpedia triples, the results we achieved can not be transferred to all DBpedia entities and their properties. Therefore, ongoing work continues in the development of sound property ranking heuristics for the purpose of detecting the most relevant characteristics of entities, but also semantically strong (respectively weak) relationships between them.

Nonetheless, *WhoKnows?* had the very important side effect of detecting inconsistencies in DBpedia. Automatic recognition of semantic inconsistencies remains to be a difficult challenge since a valid evaluation is almost impossible in the light of the large amount of data in the LOD resources. The automatic detection, the prevention, and particularly the rectification of LOD inconsistencies is an important research issue and every successful contribution will have a significant impact on the usefulness and availability of semantically enriched data in the web. In this way, a more precise semantic search on this data is achieved, which directly improves all recommendations based on semantic relationships.

6 Acknowledgement

We would like to thank our students Emilia Wittmers, Eyk Kny, Sebastian Kölle, and Gerald Töpper for their outstanding contribution for *WhoKnows?* in the seminar 'Linked Open Data Application Engineering' at Hasso-Plattner-Institute in summer term 2010.

References

1. Waitelonis, J., Sack, H.: Towards Exploratory Video Search Using Linked Data. In: Proc. of the 2009 11th IEEE Int. Symp. on Multimedia (ISM2009), Washington, DC, USA (December 2009)
2. Bizer, C., Heath, T., Idehen, K., Berners-Lee, T.: Linked data on the web. In: Proc. of the 17th Int. Conf. on World Wide Web, ACM (2008) 1265–1266
3. Hees, J., Roth-Berghofer, T., Dengel, A.: Linked data games: Simulating human association with linked data. In Atzmller, M., Benz, D., Hotho, A., Stumme, G., eds.: Proceedings of LWA2010 - Workshop-Woche: Lernen, Wissen & Adaptivitaet, Kassel, Germany (2010)

4. Surowiecki, J.: The Wisdom of Crowds: Why the Many Are Smarter Than the Few and How Collective Wisdom Shapes Business, Economies, Societies and Nations. Doubleday (May 2004)
5. von Ahn, L., Dabbish, L.: Designing games with a purpose. Communications of the ACM **51**(8) (2008) 58–67
6. von Ahn, L., Dabbish, L.: Labeling images with a computer game. In: Proceedings of the 2004 Conference on Human Factors in Computing Systems, CHI 2004, Vienna, Austria, April 24 - 29, 2004. (2004) 319–326
7. von Ahn, L., Ginosar, S., Kedia, M., Liu, R., Blum, M.: Improving accessibility of the web with a computer game. In Grinter, R.E., Rodden, T., Aoki, P.M., Cutrell, E., Jeffries, R., Olson, G.M., eds.: CHI, ACM (2006) 79–82
8. von Ahn, L., Liu, R., Blum, M.: Peekaboom: a game for locating objects in images. In: CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems, New York, NY, USA, ACM Press (2006) 55–64
9. von Ahn, L., Kedia, M., Blum, M.: Verbosity: a game for collecting common-sense facts. In: CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems, New York, NY, USA, ACM Press (2006) 75–78
10. Siorpaes, K., Hepp, M.: Ontogame: Weaving the semantic web by online gaming. In Hauswirth, M., Koubarakis, M., Bechhofer, S., eds.: Proceedings of the 5th European Semantic Web Conference. LNCS, Berlin, Heidelberg, Springer Verlag (June 2008)
11. Turnbull, D., Liu, R., Barrington, L., Lanckriet, G.: A game-based approach for collecting semantic annotations of music. In: Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR 2007). (2007)
12. Brin, S., Page, L.: The anatomy of a large-scale hypertextual web search engine. In: Proceedings of the seventh international conference on World Wide Web 7. WWW7, Amsterdam, The Netherlands, The Netherlands, Elsevier Science Publishers B. V. (1998) 107–117