RISQ! Renowned Individuals Semantic Quiz – A Jeopardy like Quiz Game for Ranking Facts

Lina Wolf, Magnus Knuth, Johannes Osterhoff, Harald Sack
Hasso-Plattner-Institute Potsdam
Prof.-Dr.-Helmert-Str. 2–3
14482 Potsdam, Germany
{lina.wolf|magnus.knuth|johannes.osterhoff|harald.sack@hpi.uni-potsdam.de}@hpi.uni-potsdam.de

ABSTRACT

In 2011 the IBM Computer Watson was beating its human opponents in the American TV quiz show *Jeopardy!*. However, the questions for the quiz have been developed by a team of human authors. Authoring questions is a difficult task, because in a *Jeopardy!* game the questions should be neither too easy nor too hard and should fit the general scope of knowledge of the audience and players. Linked Open Data (LOD) provides huge amounts of information that is growing daily. Yet, there is no ranking that determines the importance of LOD facts, as e. g. by querying LOD for movies starring a distinct actor provides numerous answers, whereas it cannot be answered, which of the movies was the most important for this actor. To rank search results for semantic search various heuristics have been developed to cope with the problem of missing rank in the semantic web. This paper proposes a *Jeopardy!* like quiz game with questions automatically generated from LOD facts to gather ranking information for persons to provide a basis for the evaluation of semantic ranking heuristics.

Categories and Subject Descriptors

E.0 [**Data**]: General; K.8.0 [**Personal Computing**]: General—*Games*

General Terms

Human factors, Experimentation

Keywords

Games with a purpose, linked open data, data cleansing

1. INTRODUCTION

In February 2011 the Watson Question Answering system [2] built by the IBM Research team challenged two human champions in the American TV quiz show *Jeopardy!* and bested them. However, the questions being played for the quiz had been created by human authors. The information used to solve these questions came from the Linked Open Data (LOD) and analysis of large amount of documents like newspaper articles. Even though Watson has won the quiz show, the knowledge that can be drawn automatically from the LOD cloud is far from being perfect. The problems arising are unclean data derived from heterogeneous sources as well as unclear and sometimes ambiguous semantics. E. g., the DBpedia [1] property `dbp:title` is used for the title of a person as well as for the title of a musical piece.

```
@prefix : <http://dbpedia.org/resource/>.
@prefix dbp: <http://dbpedia.org/property/>.
@prefix dbo: <http://dbpedia.org/ontology/>.
@prefix foaf: <http://xmlns.com/foaf/0.1/>.
:Jimmy_Carter
 dbp:title "Former President of the United States".
:George_Frideric_Handel
 dbp:title "Flute Sonata in E minor".
```

Another problem with LOD is that facts being represented as RDF triples do not have a given ranking according to their content. Looking at the RDF graph all RDF properties seem to have the same importance for a given RDF subject. This problem was already addressed by Waitelonis & Sack [13] by developing property ranking heuristics to be applied for a semantic video search engine. They used simple statistical measures and linguistic features to determine a property ranking. To perform a valid evaluation for the property ranking heuristics an objective ground truth has to be provided, where facts are assigned their appropriate rank of importance. However, the importance of a fact often is a personal point of view, since interests and knowledge affect the individual appraisal. Properties that might be a good measure for identifying an individual, e. g. the date of birth, might not be known or opted as important by a large amount of people. To find a sound baseline of property ranking, a large amount of answers has to be provided by a variety of users and thus, to make use of the so-called wisdom of the crowds.

Since ranking properties for most users is a rather boring task, it is necessary to provide incentives that motivate people to do all the work. Games With a Purpose [9] have been used in recent years to build ontologies or tag images and videos as shown in Section 2. In this paper we present an online *Jeopardy!* like quiz game called *RISQ!* (Renowned Individuals Semantic Quiz) that generates questions from

I-SEMANTICS 2011, 7th Int. Conf. on Semantic Systems, Sept. 7-9, 2011, Graz, Austria

the DBpedia and which is able to generate a ranking for famous persons and their properties. The game is integrated into the social network Facebook[1] and can as well be played standalone[2].

As an interesting side effect not only a solid ground truth for evaluating property ranking heuristics could be generated, but the game has been proven to serve also as a mean to detect semantic inconsistencies and flaws of the underlying data.

This paper is structured as follows: Section 3 describes how the gameplay could be adopted from its role model from TV, and since creating an enjoyable game is not a trivial task, Section 4 illustrates the construction of its parts in detail. Section 5 assesses the results that could be drawn from the user generated data so far. Finally, we give an outlook about the further development of the game and the utilization of the achieved results.

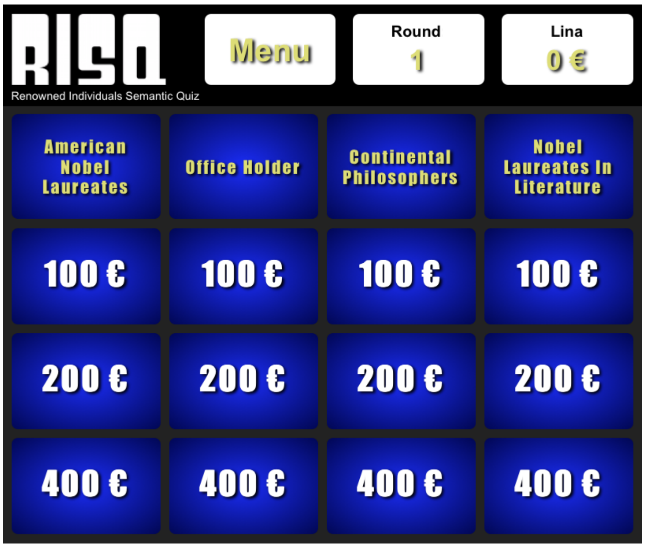
2. RELATED WORK

Linked Data has moved into the focus of information providers and is by far not solely an academic topic any more. Various web applications consume Linked Data to enrich their own data and to enable semantic browsing on the web, as e. g. the BBC music platform[3]. But, these new amounts of data do not only bring advantages, since there are flaws and inconsistencies resulting from the automatic generation. Accompanied by the lack of a valid ranking of the contained facts only some of these inconsistencies can be detected and resolved in an automated way. Many inconsistencies stem from semantic or content-related mismatches that can only be detected or resolved manually by the user. To approach these problems it seems reasonable to tap the wisdom of the crowds, which adopts the collective opinion of a group of individuals. The relevance and correctness of a fact is derived from the answers of many, who are all provided with the same information. At the same time an important problem is to attract the people's attention to this issue and to motivate them to contribute their knowledge in order to get a sufficient number of answers. Here, the development of games with a purpose has emerged as an approved method.

Additionally, as the available amount of linked data is constantly growing the importance based ranking of datasets, entities, and properties becomes more and more important. Toupikov et al. propose an analytic approach called DING! [8] to rank whole datasets (dataset being defined as "a collection of data, published and maintained by a single provider, availible as RDF [. . .]"). The dataset metadata provided in VoID (Vocabulary of Interlinked Datasets[4]), inter-dataset properties are ranked by analytical methods, however the approach does not consider intra-dataset properties.

Mirizzi et al. propose a hybrid method of RDF link-analysis and non semantic textual and search engine result analysis to rank entities with the relation between them [5]. However, they neither rank properties nor make any statements about the relationship between an entity and its literals.

Early games supporting knowledge acquisition tasks include the *ESP Game* [10], and *Phetch* [11] for annotating images and *Peekaboom* [12] for localizing objects in images. In the context of ontology building *Guess What?!* [4] and the *Virtual Pet Game* [3] have to be considered. The *OntoGame platform* [7] provides a generic infrastructure to build up games in connection with the Semantic Web.



American Nobel Laureates	Office Holder	Continental Philosophers	Nobel Laureates In Literature
100 €	100 €	100 €	100 €
200 €	200 €	200 €	200 €
400 €	400 €	400 €	400 €

Figure 1: *RISQ!* screen with topics and questions

The quiz game *SpotTheLink* [6] tries to align concepts from DBpedia to the Proton[5] upper ontology. Two players have to choose a concept from the Proton ontology that fits a random concept taken from the DBpedia and specify the relation between both. The players earn credits if both answers do correspond. According to consensus and majority of the given answers SKOS mappings between the two ontologies are generated.

All referenced games depend on a multi-player mode, which must be considered to be a potential difficulty. Such games can only be played, if at least two people are willing to play at the same time. Since we also want to reach people, who are playing single or just have a few spare minutes, we decided to design *RISQ!* as a single-player game, though a multi-player mode can be added in future.

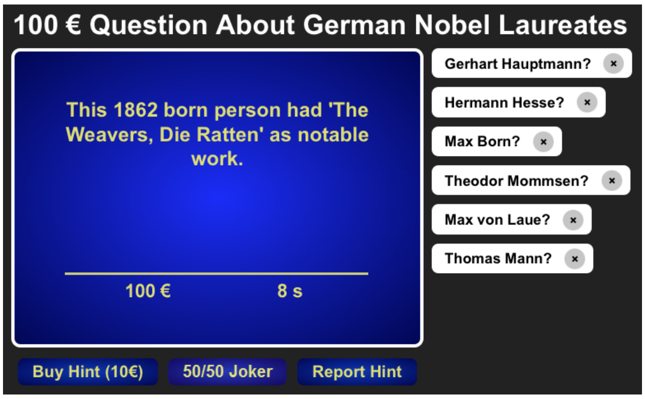
As the original game show on TV, *RISQ!* is simultaneously educational and entertaining, which encourages people to keep on playing. Since *RISQ!* is integrated in a social network, whose members differ in age, gender, social background, and origin, we expect a highly diverse collection of opinions. The data originating from the players' interaction is stored in a database for subsequent analysis.

3. GAMEPLAY AND RULES

3.1 Rules of the TV Show

In the *Jeopardy!* TV show[6] three contestants are playing against each other. The winner is the one, who earned the most money. Money can be earned by correctly answering a question and can also be lost by giving a wrong answer. Actually the clues to solve a question are given in a longer text form and the contestant has to provide the answer as a

[1] http://tinyurl.com/facebook-risq
[2] http://tinyurl.com/risqgamefb
[3] http://bbc.co.uk/music
[4] http://www.w3.org/TR/void/
[5] http://proton.semanticweb.org/
[6] http://www.jeopardy.com/



100 € Question About German Nobel Laureates

This 1862 born person had 'The Weavers, Die Ratten' as notable work.

Figure 2: "Who was G. Hauptmann?" Display of a clue and game-controls

question. A typical clue would be ”On TV, this actor played a father, who knew best & the kindly Dr. Marcus Welby”. The correct answer to this clue would be ”Who is Robert Young?”. Almost all *Jeopardy!* questions ever played have been collected by fans and can be accessed and analysed in the *J! Archive*[7].

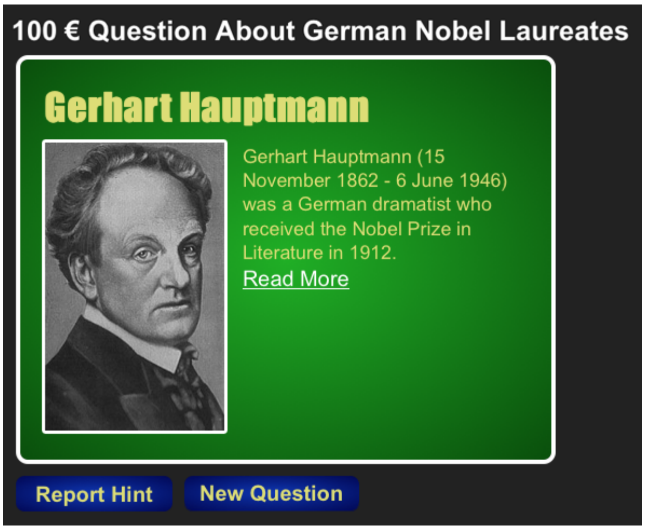
Overall, the game consists of three rounds. In the first two rounds a matrix of six topic categories and five prize categories ($100–$500) that results in 30 questions are played. In the second round the prizes are doubled. In the third round only one single question is played. The contestants have to decide on how much they want to bet on the question after only having seen the category label.

When a question is revealed the moderator reads the clue to the contestants. The moment he finishes reading the contestants can buzz in (by pressing a button). The first contestant to buzz in may answer the question. If she fails, the next contestant in row gets the opportunity to take a guess. After a question has been answered, the contestant who guessed correctlymay choose the next question to be played.

There are a couple of extra questions. Each game hides one *Double Jeopardy!* question. Here, the contestant may choose about how much money to bet and only she is allowed to answer the question. There are also types of questions including pictures, sound, or video.

3.2 Adaption of the Rules for *RISQ!*

Deciding for a *Jeopardy!* style quiz game to evaluate the importance of Linked Data facts has several advantages. Questions on the *Jeopardy!* TV show are usually quite hard to answer, the clues given are sometimes quite far away, so *Jeopardy!* fans will probably not mind hard questions. Many standard quiz games start at a low level with simple questions, while higher levels can only be achieved by answering the low-level questions. By answering a certain amount of questions wrongly the contestants will drop out. In data retrieval such a game style would lead to simple questions being played over and over again, while higher level questions can not be evaluated, because it is less likely that they will be played at all. If the questions are not properly ordered according to their difficulty the contestants may get easily frustrated. Compared with this, in *RISQ!* the questions are ordered by their difficulty level (see Section 4.1 for its determination) as well but each difficulty level will be played in each round. The higher prize offered for difficult questions offers incentives to play the harder question. Figure1 shows the screen for selecting the difficulty level and question category.

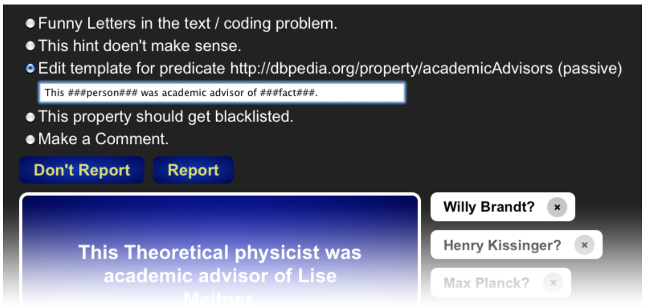


100 € Question About German Nobel Laureates

Gerhart Hauptmann

Gerhart Hauptmann (15 November 1862 - 6 June 1946) was a German dramatist who received the Nobel Prize in Literature in 1912.

Figure 3: A solution card



- Funny Letters in the text / coding problem.
- This hint doen't make sense.
- Edit template for predicate http://dbpedia.org/property/academicAdvisors (passive)

This ###person### was academic advisor of ###fact###.

- This property should get blacklisted.
- Make a Comment.

This Theoretical physicist was academic advisor of Lise Meitner

Figure 4: Reporting errors in grammar or semantics of a property

In order to evaluate whether a certain property is important, we are forced to also show hints to the users which they probably can not answer. Since we do not want to frustrate the contestants and to collect as much data as possible we decided to change the game such as that additional hints can be bought (with game money) when the contestant can not answer the question by the given hint. Figure 2 shows a clue and the game controls for buying a new hint and applying the 50/50 joker.

RISQ! is supposed to be not only entertaining but also educating. Therefore, when a question has been guessed correctly or wrongly, information about the correct solution is displayed together with a link to the corresponding Wikipedia page. A green background of this card indicates that the guess has been correct, while a red background indicates the guess has been wrong. Figure 3 shows the screen displayed after correctly guessing a person.

Since DBpedia data as being the foundation of all questions and clues for *RISQ!* does contain errors and incon-

[7] http://www.j-archive.com/

sistencies, we offer a reporting mechanism to the users to provide feedback, if there is something wrong about a clue (pointing to an inconsistency in the RDF triple being used for question construction). The reporting mechanism can also be used to improve the quality of the clues by improving the grammar-template or by blacklisting properties. Figure 4 shows an example for improving the clue grammatically.

4. CONSTRUCTING CLUES

While the original TV show *Jeopardy!* is played with six categories each having five different prizes, *RISQ!* is played in four categories having three different prizes ranging from 100 € to 400 €. The higher the prize, the harder is the question to answer. All questions have to be played.

4.1 Preranking of Persons

In the current version of *RISQ!* we have limited the topic range to persons, because if we include all categories available within DBpedia, the user would soon get bored or frustrated for not knowing anything about rather uncommon questions. Constructing ideal questions for *RISQ!* demands that the answer (the person to be found) is probably known to the contestants. Therefore, the answers to the lower prized questions should be known by most contestants while the answers to questions in the higher prized stages may only be known to a few.

Which persons are known to the contestants cannot be regarded utterly culture independent. Since the language of the game is English based on the DBpedia facts, we decided to aim at an anglophone audience from the Americas and Western Europe. The process of choosing which renowned persons should be played in each level of the game corresponds to a recommendation system. By using a collaborate recommendation approach we would be faced by the cold-start phenomenon. Furthermore, DBpedia lists a high number of persons that are so little known that they would only create frustration in a quiz game. We decided never to play these scarcely known persons except in the bonus question of round 3. This also leads to the fact that these questions are played rather rarely.

Therefore, we decided for a simple heuristic recommendation approach based on the indegree of a Wikipedia article. If a large number of articles point to the person's article, the article is considered to be more important. Furthermore, a person, whose Wikipedia article has been translated in several languages is important to more then one ethnic group and can therefore be considered more important that an article that has not been translated. Also, a person that is listed in many different databases is probably more important than one only listed in a few. Therefore, we take additionally into account, whether the person got listed in the German person database PND[8], in Freebase[9], YAGO[10], and, in the case of actors and film directors, whether they are listed in the Internet Movie Database[11] as well as the number of films they are connected to.

Contrary to our assumption, Google and Bing search result counts have proven to deteriorate the achieved ranking so far. This is, because Google and Bing only perform a keyword-based search that favours common names and names that equal common words in other languages.

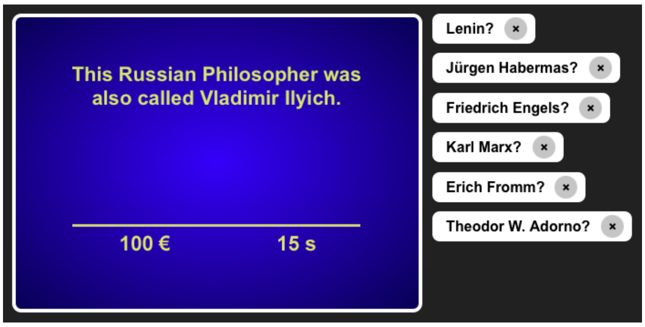

This Russian Philosopher was also called Vladimir Ilyich.

Figure 5: "Who was Lenin?" - active Clue

Afterwards, all values have been normalized and combined regarding their proposed importance. As evaluation of the quality we gave a list of the first 5,000 persons to 10 contestants and asked to mark whom they know well enough to be able to recognize them in a quiz. The contestants marked between 200 and up to 3,000 persons, while the markings have been more frequent at the beginning of the ordered list.

4.2 Choosing Categories

In order to create interesting questions a category should contain at least ten persons, who possess a certain amount of unique properties to be playable (at least 7) and are sufficiently known according to our heuristic to be guessed successfully. Each category complying with the above criterias gets played with the same probability. The level to the threshold for a person to be known is lowered as the above criterias apply to more categories.

4.3 Generation of Clues

In the original *Jeopardy!* quiz show only one clue is shown to the contestants and this clue usually targets one exact answer. However, for automatically created questions we cannot ensure all clues to be indisputably solvable by the contestants without a ranking of the properties of the persons record. Ranking the properties is a goal of this paper. If the contestant lose the question each time a clue is ambiguous or unknown, it would swiftly discourage her and deliver only poor data about whether the person would have been known. Therefore, we decided to enable the contestants to ask for more clues, each clue costing some of the contestant's prize.

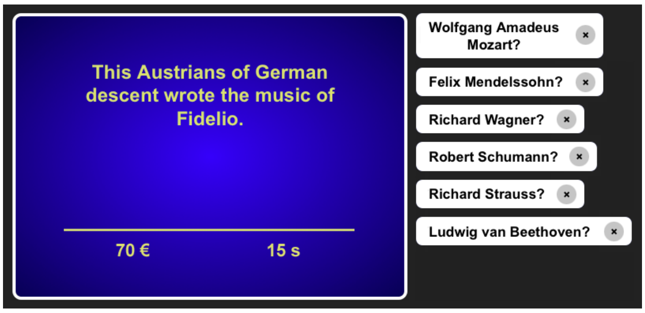
For now there are three different types of questions:

Active Clues

The program constructs active questions by creating a naturally formed sentence using one of the subjects categories and a triple having the person as the subject, as seen in Figure 5. The label of the category gets mapped to its singular by using a manually created dictonary. Removing the plural 's' ist not sufficient as we cannot distiguish between plural forms of words and words or names ending on 's'. The properties' label will be looked up in a manually created template table for constructing naturally sounding sentences.

Constructing the template table of person properties is not

[8] http://www.d-nb.de/standardisierung/normdateien/pnd.htm
[9] http://www.freebase.com/
[10] http://www.mpi-inf.mpg.de/yago-naga/yago/
[11] http://www.imdb.com/



This Austrians of German descent wrote the music of Fidelio.

Figure 6: "Who was Beethoven?" - passive Clue

trivial as many properties have to be interpreted. As, e.g., the property `dbp:chancellor` is not self-explaining. The meaning is, the subject was in some official office while the object was chancellor. Therefore, the following template was created manually:

```
  dbp:chancellor, "This #subject# was in office
while #object# was chancellor."
```

If the object is a literal it gets displayed directly. Otherwise the object's label gets displayed. In order to prevent clues making no sense to human readers some properties are blacklisted including homepage adresses, PND-numbers, image sizes, `owl:sameAs`, wikiPageRedirects and the such. Furthermore, objects containing the name of the person fully or in parts have to be filtered out. Names can appear in many different kinds of properties like categories or in the label of a song performed by the person. Simply filtering out more property types is not sufficient as labels of related subjects sometimes contain the name of a person. And names often appear in different kinds of properties.

Passive Clues

Passive clues are constructed analogously to the active clues. However, passive clues are constructed from triples containing the person as the object. In Figure 6 you see an example for a passive clue. Since all subjects in RDF are always URIs there are no literals found in this question type. Therefore the English label of the subject is used. The property template table contains information of whether the property is used actively or passively.

Multimedia Clues

In these clues an image of the person or her signature is displayed. This clue type raises the fun of playing and indicates whether the person has been known in the first place.

4.4 Names in Literals to be displayed in Clues

In DBpedia there are many different properties which are made to contain a persons name. We want to prevent displaying the desired solution in the clue. However, a name property does not always display the solution, it might also contain the artist name or birth name of a person. Furthermore, potentially any property might contain the persons name, e. g. the scientist Schrödinger has the triple `:Schrödinger dbp:knownFor "Schrödinger's cat".` and the German pop singer Nena has the triple `:Nena dbp:author "99 Luftballons (Nena)".` Offering these hints is valuable to find out the importance of the relation. All text in brackets gets removed, as these usually contain the persons name or further hints. In every other place where the name to be found appears it gets disguised as "[person]".

4.5 Improving Clue Quality

Clues in which subject and object are too similar are not desired as they carry few information and frustrate the contestants: *"This German actor was a German Actor"*. Therefore we do n-gram comparisons to prevent such clues from appearing.

Many categories carry a professions name in plural "Actors", "American People of Irish Decent". The plural gets removed by a manually created dictionary. Some other undesired terms are also improved by a dictionary approach, so "1960 born" becomes "person born in 1960".

5. RESULTS

So far, in *RISQ!* 6,484 questions have been played by 118 different users of which 3,678 have been answered correctly. To solve these questions 12,924 clues have been displayed. 10,228 distinct triples concerning 1,265 persons have been played. The most often played person is `:Jack_Nicholson` (306 times).

5.1 Evaluation of Property Rankings

We assume a property to be relevant, if a fact using this property helps the player to decide for the sought-after person. Getting a new clue about a person, the player can choose an answer from the provided person list or deselect an uneligible person from there. The more often a fact containing a property leads to a correct choice by the user, the more relevant is this property regarded. *RISQ!* internally keeps track of how often a triple is played and how often this leads to a correct answer or deselection. We calculate a score for each property from the ratio of wins ($winrate_i = \frac{wins_i}{played_i}$), wrong choices ($lossrate_i = \frac{losses_i}{played_i}$) and correct deselections a clue resulted in ($deselectionrate_i = \frac{deselections_i}{6 \cdot played_i}$). We normalize the number of correct deselections with an average length of the provided person list, which is 6.

$$score_{prop\,i} = winrate_i - lossrate_i + deselectionrate_i$$

The top fifteen and last five properties for the upper category `dbo:Person` are shown in Table 1. In order to get significant results only properties that have been played at least ten times are included in the analysis. Passively used properties are followed by a detached *Of*. Some property rankings must be regarded with care, since the displayed clues may contain redundant information that could not be filtered out, as e. g. `dbp:birthname` allows to easily pick out a person from the provided suggestions by its similarity to the real name, as in:

```
:Sophia_Loren
  dbp:birthname "Sofia Villani Scicolone".
```

Regarding a more specific category, such as e. g. the members of the class `dbo:Politician`, which has been one of the most played (2,243 clues) direct subcategories of `dbo:Person`, the property ranking changes, as can be seen in Table 2. Whereas some properties are not included, since they are not used for politicians, some rankings are reordered significantly, as e. g. the rankings of the properties `dbp:predecessor` and `dbo:party` that seem more expressive for politicians than for people in general.

Table 1: Relevance ranking for DBpedia properties.

Rank	Property	Score	Played	Winrate	Lossrate	Deselectionrate
1	dbo:birthName	1.36	17	0.94	0.18	0.60
2	dbp:mus Of (music)	1.03	11	0.91	0.27	0.39
3	dbp:imageCaption	0.89	11	0.36	0.00	0.53
4	dbp:field	0.88	31	0.77	0.16	0.27
5	dbp:afterElection Of	0.84	22	0.55	0.05	0.34
6	dbo:monarch Of	0.75	10	0.80	0.20	0.15
7	dbp:profession	0.69	12	0.42	0.08	0.36
8	dbp:successor	0.67	27	0.52	0.22	0.37
9	dbp:movement	0.64	11	0.45	0.00	0.18
10	foaf:depiction (pic)	0.63	143	0.52	0.15	0.26
11	dbp:knownFor	0.60	43	0.65	0.23	0.19
12	dbo:knownFor	0.60	42	0.55	0.14	0.19
13	dbp:birthname	0.59	75	0.44	0.17	0.32
14	dbo:president	0.58	14	0.50	0.43	0.51
15	dbp:namedafter Of	0.56	18	0.67	0.22	0.11
225	dbp:district	-0.23	25	0.12	0.52	0.17
226	dbp:birth	-0.27	15	0.00	0.27	0.00
227	dbp:period	-0.29	28	0.21	0.54	0.04
228	dbp:participants Of	-0.34	30	0.10	0.47	0.03
229	dbo:father Of	-0.37	10	0.10	0.50	0.03

Table 2: Relevance ranking for properties of dbp:Politician.

Rank	Rank in Person	Property	Score	Played	Winrate	Lossrate	Deselectionrate
1	127	dbp:after Of	0.61	11	0.27	0.09	0.42
2	114	dbp:predecessor	0.58	12	0.75	0.17	0.00
3	38	dc:description	0.55	14	0.29	0.21	0.48
4	30	dbp:termStart	0.44	28	0.39	0.11	0.15
5	17	dbo:party	0.44	13	0.54	0.46	0.36
25	163	dbp:starring Of	-0.45	11	0.00	0.45	0.00
26	226	dbp:participants Of	-0.47	19	0.11	0.58	0.00
27	223	dbp:district	-0.53	15	0.13	0.67	0.00

Table 3: Ranking of facts about Dwight D. Eisenhower.

	Property	Object	Score
1	dbp:branch	:United_States_Army	0.40
2	dbp:birthDate	”1890-10-14”	0.06
3	dbp:leader Of	:Allied-occupied_Germany	0.00
4	foaf:givenname	”[person] David”	0.00
5	dbp:starring Of	:The_True_Glory	-0.57

Table 4: Ranking of facts about Jack Nicholson.

	Property	Object	Score
1	dbp:birthname	”John Joseph”	0.48
2	dbp:occupation	”Actor, director, producer”	0.30
3	dbp:reg Of	:Goin_South	0.05
4	dbp:spouse	”Sandra Knight; 1 child”	0.00
5	dbp:birthdate	”1937-04-22”	-0.04
6	dbp:pro Of	:The_Two_Jakes	-0.06
7	dbp:yearsactive	”1958–present”	-0.14
8	dbp:expiry	”2011-01-08”	-0.17

In case sufficient triples for a certain person have been played, we can deduce a ranking of facts about this person. Table 3 shows the ranked facts known about Dwight D. Eisenhower, Table 4 the ranked facts of Jack Nicholson in comparision. Only facts played at least five times were taken into account. The ranking of these facts corresponds quiet well to the ranking of the used properties. It shows, which facts are characteristic for Eisenhower, while the fact that he performed in a movie is rather misleading.

Currently, we do not have enough data to make valid a statement about inconsistent triples. Triples using the property `dbp:title` have been most often (6 times) marked as "no sense", including e. g. `:Chiang_Kai-shek dbp:title "School name"`. Besides such real inconsistencies players reported on properties that have ambigous meaning, as e. g. `dbp:title` appears in both of the following triples:

```
:Angela_Merkel
 dbp:title "Chancellor of Germany".
:Antonio_Vivaldi
 dbp:title "Op. 4 Concerto 3".
```

6. OUTLOOK ON FUTURE WORK

6.1 Usage of Collected Data

The data gathered within this game can be used in various areas of application. Having better rankings for properties in linked data allows us to improve future semantic search applications by presenting users only the most relevant information about entities. By tracking the players' correct answers, it is possible to determine their fields of interests. So we can propose an interest profile, that can be applied for personalized services.

The game allows to report problems. When a player feels like the given hint does not make sense, she can tag the hint as senseless or write a comment. These tags can help to find inconsistent triples, e. g. where the properties range and domain restrictions are not respected. After the correct answer has been revealed, it is also possible to give a feedback for the round. If a given hint was not correct, the player can mark it as wrong, which allows to find false statements. Furthermore the player can state, whether the person did not fit in the scope of the category, the person is no person at all or there was a character problem. Since the triples in DBpedia are not thoroughly clean, this feedback is a welcome adding to the property ranking.

Our further aim is to use this data for a forthcoming linked data cleansing project, in which we collect large amounts of corrections by human users to identify inconsistent triples, check them for significance and publish this data as "linked data patches".

6.2 Further development of the game

The scope of the game is supposed to be extended to categories beyond persons. Categories feasible for being played include but are not limited to movies, music bands, recent events, locations and many more. Specialised topics like chemic elements, zoologic taxonomy, historic events, and many more could be played either by students of the topic or by an interested audience.

The two main points that have to be improved in the game are the textual representations of the properties and the reporting possibilties for inconsistent triples. Identical properties have a different semantic meaning and thus the grammar also depends on the domain and range of the triple. The name of a musical piece and the title of a person do not have the same semantics but are represented by the same URI. To get more precise annotations of inconsistent triples, the player needs to be able to specify which fact he assumes to be wrong. Nevertheless, the interface should remain easy to use. Providing a bonus for consensually identifying inconsistencies might draw additional attention to this task.

The importance of individual items/persons as well as the importance of properties is depending on the ethnic and social background. In order to attract more German contestants we consider to develop a German language version of the game based on the German DBpedia[12]. Since the German Wikipedia contains less info-boxes than the English one, the recent version of the German DBpedia contains fewer triples. It has to be tested whether the number and quality of the triples is sufficient to create an interesting game or which other sources can be used. Furthermore, the German grammar is more complex than in English. Gender of articles ('der', 'die', 'das'), casus of words and declinations of adjectives can make constructing natural sounding questions a true challenge.

7. REFERENCES

[1] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. DBpedia: A Nucleus for a Web of Open Data. In *Proceedings of 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference (ISWC+ASWC 2007)*, volume 4825 of *Lecture Notes in Computer Science*, chapter 52, pages 722–735. Springer Berlin / Heidelberg, Berlin, Heidelberg, Nov. 2007.

[2] D. Ferrucci. Building Watson: An Overview of the DeepQA project. In V. Salapura, M. Gschwind, and J. Knoop, editors, *PACT*, pages 1–2. ACM, 2010.

[3] Y.-l. Kuo, J.-C. Lee, K.-y. Chiang, R. Wang, E. Shen, C.-w. Chan, and J. Y.-j. Hsu. Community-based game design: experiments on social games for commonsense data collections. In *Proceedings of the ACM SIGKDD Workshop on Human Computation*, HCOMP '09, pages 15–22, New York, NY, USA, 2009. ACM.

[4] T. Markotschi and J. Völker. Guesswhat?! – human intelligence for mining linked data. In *Proceedings of the Workshop on Knowledge Injection into and Extraction from Linked Data (KIELD) at the International Conference on Knowledge Engineering and Knowledge Management (EKAW)*, 2010.

[5] R. Mirizzi, A. Ragone, T. D. Noia, and E. D. Sciascio. Ranking the Linked Data: The Case of DBpedia. In B. Benatallah, F. Casati, G. Kappel, and G. Rossi, editors, *ICWE*, volume 6189 of *Lecture Notes in Computer Science*, pages 337–354. Springer, 2010.

[6] K. Siorpaes. OntoGame: games with a purpose for the semantic web. *ESWC 2008 Ph. D. Symposium*, 2007.

[7] K. Siorpaes and M. Hepp. Ontogame: Weaving the semantic web by online gaming. In M. Hauswirth, M. Koubarakis, and S. Bechhofer, editors, *Proceedings of the 5th European Semantic Web Conference*, LNCS, Berlin, Heidelberg, June 2008. Springer Verlag.

[12] http://de.dbpedia.org/

[8] N. Toupikov, J. Umbrich, R. Delbru, M. Hausenblas, and G. Tummarello. DING! Dataset Ranking using Formal Descriptions. In *WWW 2009 Workshop: Linked Data on the Web (LDOW2009)*, 2009.

[9] L. von Ahn. Games with a purpose. *IEEE Computer Magazine*, 39(6):92–94, 2006.

[10] L. von Ahn and L. Dabbish. 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*, pages 319–326, 2004.

[11] L. von Ahn, S. Ginosar, M. Kedia, R. Liu, and M. Blum. Improving accessibility of the web with a computer game. In R. E. Grinter, T. Rodden, P. M. Aoki, E. Cutrell, R. Jeffries, and G. M. Olson, editors, *CHI*, pages 79–82. ACM, 2006.

[12] L. von Ahn, R. Liu, and M. Blum. Peekaboom: a game for locating objects in images. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 55–64, New York, NY, USA, 2006. ACM Press.

[13] J. Waitelonis and H. Sack. Towards exploratory video search using linked data. *Multimedia Tools and Applications*, pages 1–28, 2011. 10.1007/s11042-011-0733-1.