Can we Create Better Links by Playing Games?

Jens Lehmann, Tri Quan Nguyen, Timofey Ermilov
AKSW Group, University of Leipzig, Germany
http://aksw.org, lastname@informatik.uni-leipzig.de

Abstract—**Just like links are the backbone of the traditional World Wide Web, they are an equally important element in the Data Web. There exist a variety of automated tools, which are able to create a high number of links between RDF resources by using heuristics. However, without manual verification of the created links, it is difficult to ensure high precision and recall. In this article, we investigate whether game based approaches can be used to improve this manual verification stage. Based on the VeriLinks game platform, which we developed, we describe experiments using a survey and statistics collected within a specific interlinking game. Using three different link tasks as examples, we present an analysis of the strengths and limitations of game based link verification.**

I. Introduction

The success of Linked Data suggests that the connection of knowledge bases via links is of high importance for applications of semantic technologies, e.g. in the areas of data integration and federated queries. While there has been significant recent progress on the automation of this linking process [1], [2], fully automatic approaches are unlikely to ever achieve extremely high precision and recall over a set of complex scenarios. This means that ensuring a high quality of links requires a manual process, in which humans have to validate links. Such a process can be very time-consuming and as pointed out in [3], dataset maintainers may not always have sufficient incentive for performing that task. For this reason, we want to investigate whether game based approaches can be used for this problem, which is too complex for machines to solve without help.

To do this, we created the platform VeriLinks – a lightweight framework in which users validate links while playing a game. VeriLinks is designed to be flexible while still providing an entertainment value for players. It is flexible, since it only requires a set of links, the SPARQL endpoints containing information about the resources which should be interlinked and a template for the visual presentation of the link validation task. The game will then automatically calibrate its reward and balancing system using game play statistics. For instance, questions with high agreement with other players will be rewarded with coins and special bonuses. The link validation questions influence a main game, which needs to be implemented against the VeriLinks API. Specifically, we created a tower defense style game and a turn based space battle game.

Previous work on games with a purpose for the Semantic Web, such as [4], has mostly considered multi player games in which two players compete directly against each other. In VeriLinks, the player competes against the community of all users, which played the game, including those who are currently playing it. For this, we devised a reward and balancing system, which takes into account that users may not always be able to answer questions. This system is independent of the user interface and link task, such that it can be used on top of other linking games as well.

Our main research hypothesis, which we investigate using VeriLinks, is whether the precision and recall of interlinking can be increased using game playing approaches. For this, we compute and combine confidence values from linking specifications and the game itself. We tested whether this leads to higher accuracy using three different linksets from the LATC project[1].

Our core contributions are as follows:

- the flexible VeriLinks platform and implements of two games for validating arbitrary linksets (presented in Section II)
- automatic balancing and reward methods for interlinking games (Section III)
- design and execution of an experiment investigating whether game playing can improve link quality (Section IV) and a discussion of the results (Section V)

In Section VI, related work in the area is reviewed. Finally, we conclude and give pointers to future work in Section VII.

II. Platform Structure and Overview

VeriLinks is a light-weight game platform with the purpose of validating links in the Web of Data. It consists of core functionality for game administration as well as game rewarding and balancing as described in Section III.

A. Core Functionality

Most of the basic game principles of VeriLinks are derived from other games with a purpose (GWAPs). For instance, from [5] we used design elements such as the estimation of player skill levels, score keeping mechanisms, high score lists and a randomnization of game play. The mathematical background for those functions is described in Section III. Technically, VeriLinks provides a simple API for game implementations as shown in Table I and Figure 1. The VeriLinks platform takes over server side functionality, administrative tasks for adding linksets as well as authentication mechanisms. For this reason, only few methods need to be implemented to create an interlinking game. A minimal game example[2] and documentation is provided for game developers.

[1] http://latc-projet.eu

[2] https://github.com/AKSW/VeriLinks/tree/master/games/verilinks-example

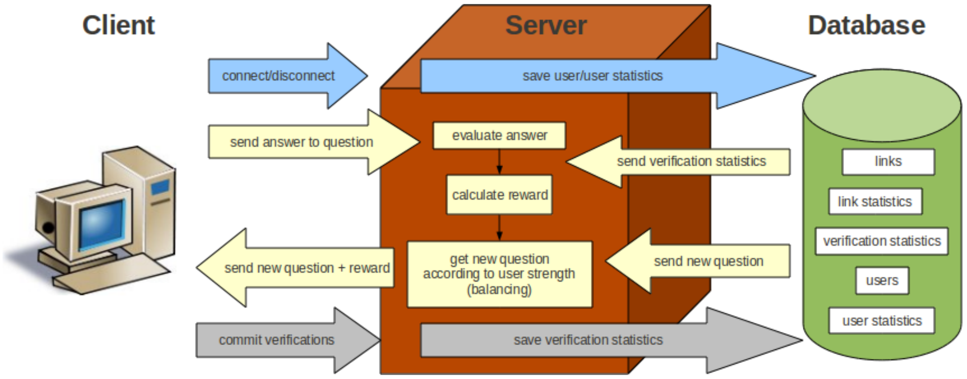


Fig. 1. VeriLinks Activity Chart

method	input	results
get user data	username	user ID, name, previous game statistics
get highscore		user and scores ranked by scores
get linksets		retrieve available linksets
get link	user ID, linkset ID, last link ID, last verification result	statistics about link correctness
post score		success (true/false)
post level statistics		success (true/false)

TABLE I
THE VERILINKS SERVER API PROVIDES METHODS FOR MANAGING USERS, LINKSETS AND GAME BALANCING. ONLY FEW METHODS NEED TO BE IMPLEMENTED FOR CREATING FURTHER GAMES. DETAILS FOR DEVELOPERS ARE AVAILABLE AT HTTPS://GITHUB.COM/AKSW/VERILINKS.

B. Game Implementations

So far, two different games have been implemented in VeriLinks: pea invasion and space ships. VeriLinks can be applied to arbitrary linksets by specifying a set of links, SPARQL endpoints containing information about the resources which should be interlinked and a template for the visual presentation of the link validation task. In the sequel, we call the combination of those 3 elements a *game task*. A set of links for a game task can be specified either in any typical RDF format or in OAEI Alignment format[3]. The latter simplifies running analytical tasks, but is not strictly required.

The first game, pea invasion, is a single player web game in which players indirectly compete against other players, i.e. their scores depend on the evaluation of others, which is typical for GWAPs. The game is divided into two components as illustrated in Figure 2:

- The *game panel* in which players have to defend from an enemy invasion.
- A *validation panel*, which displays a link using a template and lets the user validate it.

Activities in the validation panel influence the game panel: If a player answers questions correctly, he earns coins. Those coins can be used in the game to defend better against enemies. The validation panel is integrated via only two HTML tags — one container tag to indicate where the panel should be placed and one script tag to load the panel. The panel automates the displaying process of the links and provides access to all Server API methods. Details on the validation panel, which uses JsRender templates, are described in the next subsection.

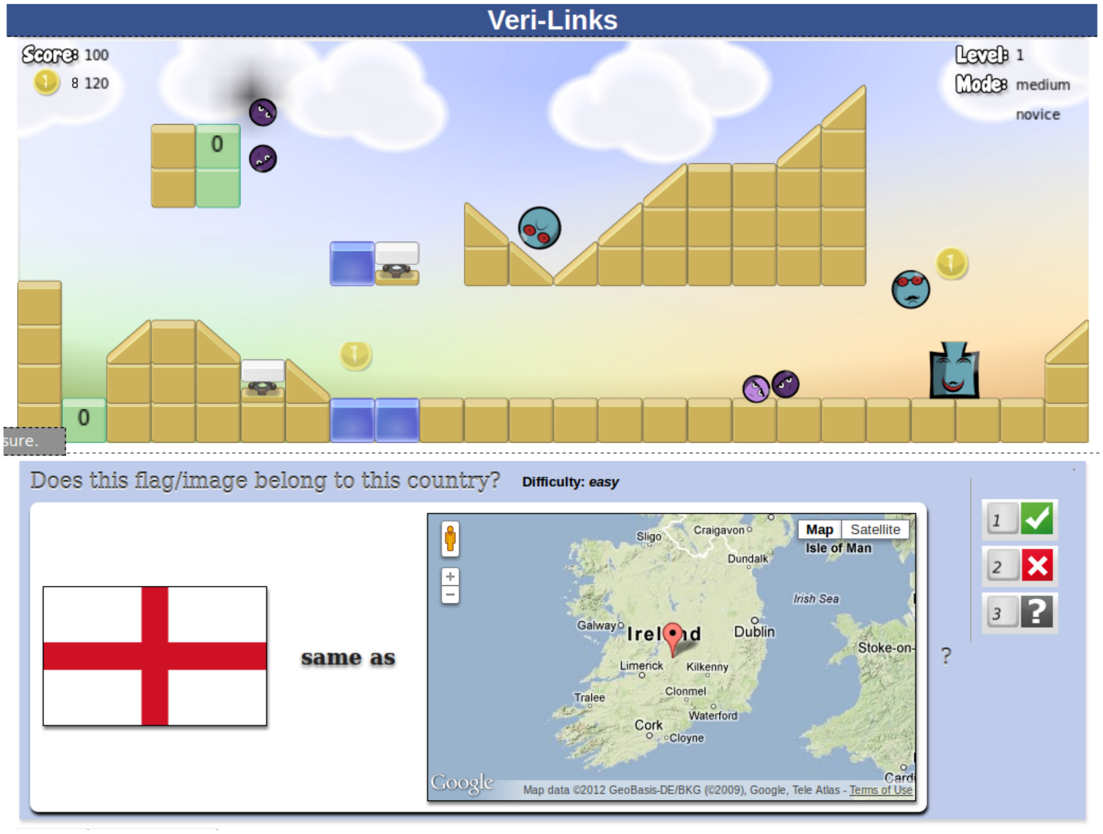
The second game, *space ships*, is based on the outcomes of our user study in Section IV. Some players found pea invasion to be a cognitively difficult game, since they had to evaluate links and play the main game simultaneously. Based on this feedback, we provided a turn based game in which link verification and game playing phases alternate. The target of the game is to destroy the ship of the opponent before he destroys yours. Within the verification phase, users can validate links in a way which favours correct validation over performing a large number of validations. The damage done to an enemy ship is calculated at the end of turn and is based on the number of verified links as well as the precision of the verification. A screenshot of the game is shown in Figure 3. The turn based game allows to implement an asynchronous multiplayer mode, in which a user can compete against an AI of a particular strength or against a ship of another user of the system. Such an approach together with integration into social networks can engage users to play more and thus verify more links.

C. Creating a Game Task from a Linkset

VeriLinks provides a user interface for adding new game tasks to its internal database. Using this interface, the user can create a new template for a linkset. The template describes the data that should be queried and which parts of the queried data should be displayed in the game. It is necessary to declare the name of both knowledge bases and their SPARQL endpoints. The SPARQL endpoint is used to retrieve all required data specified in the template. VeriLinks employs several quality checks on the data, e.g. checking whether images can actually be retrieved correctly. Only valid resources are cached in its internal database, which serves as backend for the VeriLinks API.

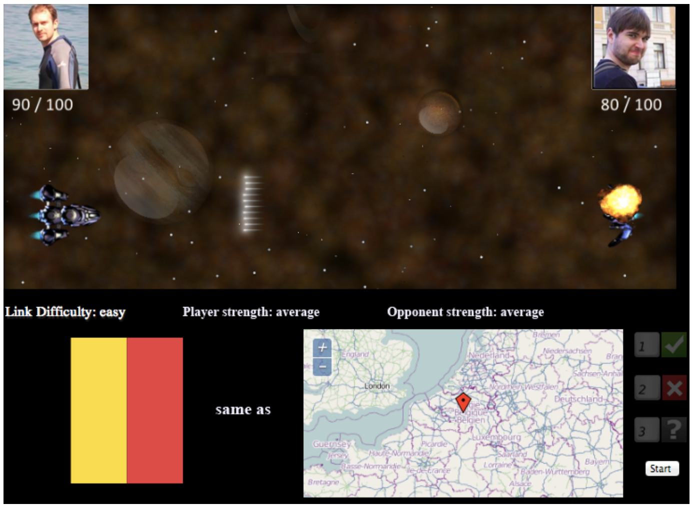
The visual presentation of resources can be adapted by

[3] http://alignapi.gforge.inria.fr/format.html



Does this flag/image belong to this country? Difficulty: easy

Fig. 2. Screenshot of the pea invasion game based on VeriLinks.



Link Difficulty: easy Player strength: average Opponent strength: average

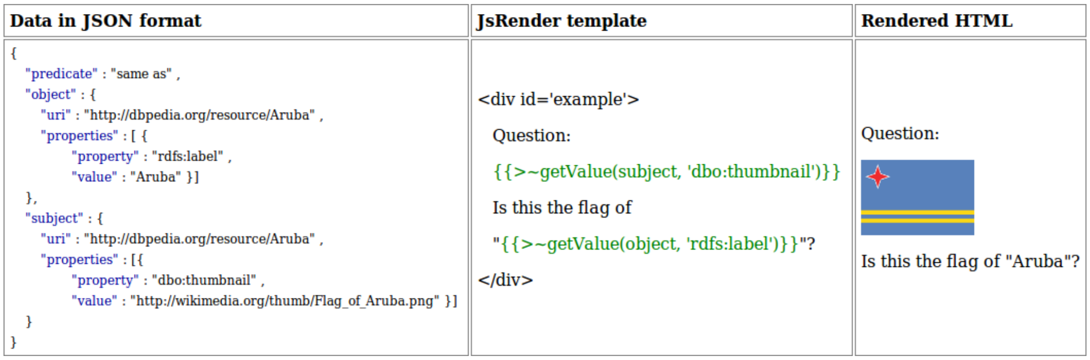
Fig. 3. Screenshot of the space ship game, which differs from pea invasion in that it has alternating verification and game phases.

using JsRender templates and CSS stylesheets. JsRender is a JavaScript templating library that allows the generation of dynamic HTML from a predefined boilerplate structure. JsRender has a code less tag syntax, supports the creation of custom functions and uses pure string-based rendering paired with high performance. VeriLinks integrates the features of JsRender to dynamically generate HTML by taking JSON input and combining it with a JsRender template, which consists of HTML markup and JsRender tags. Figure 4 illustrates how JsRender templates can be used to display a semantic web link. These JsRender templates are stored on the server and can be automatically inserted into a container tag of the game page. This approach enables a flexible way to display and arrange the retrieved data in HTML. In addition to displaying resource data as text or images, it is also possible to draw instances on a map. VeriLinks provides a default solution in which the user can select a single property per resource he prefers to display.

Overall, adding a new linkset requires only little effort and the system is sufficiently flexible to render data in a visually appealing way suitable for games. We considered this aspect to be particular important, since the effort of creating a game task should be much lower than the effort of manually evaluating many links, as otherwise there would be no benefit in terms of costs. We believe that the above template system provides a good tradeoff between flexibility and ease-of-use.

III. Game Rewarding and Balancing

One of the most important aspects of games is to balance them in such a way that they are both, challenging and rewarding, for players. In this section, we describe how those aspects were implemented in VeriLinks. As a basis for the internal rewarding and balancing functions, we use a three valued approach to link verification: A user can confirm a link, reject a link or make no decision (”not sure” option).



Data in JSON format

```
{
  "predicate" : "same as" ,
  "object" : {
    "uri" : "http://dbpedia.org/resource/Aruba" ,
    "properties" : [ {
        "property" : "rdfs:label" ,
        "value" : "Aruba" }]
  },
  "subject" : {
    "uri" : "http://dbpedia.org/resource/Aruba" ,
    "properties" : [{
        "property" : "dbo:thumbnail" ,
        "value" : "http://wikimedia.org/thumb/Flag_of_Aruba.png" }]
  }
}
```

JsRender template

```
<div id='example'>
  Question:
  {{>~getValue(subject, 'dbo:thumbnail')}}
  Is this the flag of
  "{{>~getValue(object, 'rdfs:label')}}"?
</div>
```

Rendered HTML

Question:

Is this the flag of "Aruba"?

Fig. 4. VeriLinks template rendering. A template needs to be provided to create a game from a set of links.

A. Rewards

A difference of VeriLinks compared to many other Semantic Web games (see [4] for an overview) is that a player does not play directly against another player[4]. VeriLinks is designed for single player games in which the in-game questions and rewards depend on the history of all games played so far. It is not required that other players are online at the same time. However, different players can compete by trying to reach a good position on the VeriLinks highscore, play against AIs or recorded answers of another user. Two player games, which are common in games with a purpose [3], often have a simple reward scheme: When the players agree, they get a certain reward. The VeriLinks reward system, which we describe in the sequel, has to take more factors into account.

One criterion for the obtained reward is the difficulty of the a question. More difficult questions should lead to a higher reward. At least two straightforward criteria can be used to assess the difficulty of a question automatically: 1.) How often do players selected the ”not sure” option? 2.) How strong is the disagreement between players on that question? We only take the first criterion into account as including the second would conflict with the reward system described below[5]. A simple way to compute difficulty is to divide the number of times the ”not sure” option was selected by the number of times, the question was played. We use an enhanced version of this formula, which takes the uncertainty into account in cases when the question was played only few times. Specifically, we use the center of the 80% confidence interval of the Wilson method, which has been shown to work well for small samples in [6], to estimate the difficulty d:

$$d = \frac{\hat{p} + \frac{1}{2n} z^2_{1-\alpha/2}}{1 + \frac{1}{n} z^2_{1-\alpha/2}} \tag{1}$$

In this formula, n is the number of times the link has been played, $\hat{p}$ is the percentage of ”not sure” replies, α is the error percentile and $z_{1-\alpha/2}$ is the percentile of the standard normal distribution. For VeriLinks, we use $\alpha = 20\%$, which yields $z_{1-\alpha/2} = 1.2186$. For instance, a link played 10 times with 3 unsure replies has a difficulty of 33%, whereas 9 unsure replies lead to a difficulty of 85%. Low values of α provide conservative estimates, i.e. many questions are judged to be of medium difficulty, whereas high values of α lead to more extreme judgements, i.e. many questions are either judged as very easy or very hard.

We group the difficulty in categories ”easy” ($0 \leq d < 0.2$), ”medium” ($0.2 \leq d < 0.5$), ”hard” ($0.5 \leq d < 0.7$), ”very hard” ($0.7 \leq d < 0.9$) and ”ultimate” ($0.9 \leq d \leq 1.0$). For those questions, the rewards are 10, 20, 40, 100, and 500 coins, respectively. We give additional points for *winning streaks*: If a player answers 5 questions correctly in a row, the rewards for those questions are doubled.

A player earns those awards depending on the agreement with other players on the same link. We can use the same technique as for computing the difficulty of question for also estimating the agreement a. In that case, $\hat{p}$ in Equation 1 is the percentage of times other users have previously made the same decision (confirming or rejecting the link). We then give the full reward if a is greater than an upper threshold t_u (by default $t_u = 0.7$). A penalty is given if a is below a lower threshold t_l (by default $t_l = 0.3$. A small reward is given if $t_l \leq a \leq t_u$ in order to compensate the user for evaluating a link even though we do not yet know whether the evaluation is likely to be correct.

B. Balancing

According to [7], primary goals of game balancing are to (1) quickly identify the player’s level, (2) track evolutions and regressions of player strength and (3) adapting game difficulty accordingly.

For (2), we use the last questions played by the user: The player strength s is estimated from how many questions he answered correctly in his previous questions. In order to estimate player strength, we need to factor out correct validation by chance. Since we consider link validation as a binary problem, i.e. a link is either correct or incorrect, there is usually a 50% chance that a user does the right decision[6] Let w be the window size, i.e. the number of

[4] Note that in case of the OntoGame framework, the second player could be a simulated player, but the basic concept is still the same.

[5] We give rewards when we are confident that the player provided a correct answer.

[6] This assumes that the number of correct and incorrect links is equal. We make this assumption, since we usually have no prior knowledge of the true distribution. If we assume a percentage a of correct links, then 0.5 can be replaced by $a^2 + (1-a)^2$.

previous questions considered, and c the number of correct links from those w questions. Player strength is then defined as $s = \max(0, (p/w - 0.5)/(1 - 0.5))$[7]. For instance, answering 1 or 9 out of 20 questions correctly both leads to a strength of zero as both are below the performance of random guessing, whereas 17 out of 20 correct answers results in a strength of 0.7. For VeriLinks, we use window size $w = 100$[8]. In order to satisfy requirement (1) above – a quick identification of the player's level – we assume that a user who has only played x links with $x < w$ so far has evaluated 75% of the remaining $(w - x)$ links correctly, which corresponds to a novice level in VeriLinks, but allows the user to obtain a high strength within a few minutes of game play. Strength levels, e.g. "novice" or "master" are displayed in the user interface to motivate the player.

For objective (3), we need to ensure that better players get more challenging questions. At the same time, we need to take into account how often a link has been validated: VeriLinks can validate more links by always presenting the player non-validated links, but this will reduce his joy in playing the game, since the player will only earn agreement rewards after all links have been validated at least once, which may never happen in very large linksets. For this reason, we do not select links randomly, but define an exploration factor e[9]. If a generated random number in the interval [0,1] is smaller than e, we select a link which has not yet been validated. Otherwise, we pick an already validated link. In that case, we draw a random number based on a configurable set of skill levels. For instance, for a "novice" rank, a random number between 0 and 0.3 is drawn. After that, we detect the closest difficulty value to this random number in the database. Finally, we randomly select one link from all of those having this difficulty value. The above operations are implemented in an efficient manner directly in the database to scale to large linksets. In addition to the above criteria, we also established methods to avoid that a user plays the same link twice in one session. We also discard all validations made by users, which are correct in less than 70% of all cases within one level. This is done to ignore validations made by users who just click through the game to test or try it without thinking about their validations.

IV. Evaluation and Experiments

In our experiments and the survey, we addressed the following research questions:

1) How accurate are game based interlinking methods?
2) Can they be combined with manually crafted link specification to boost F-measure?
3) Are players willing to invest time in linking games?
4) Did the balancing and rewarding as well as the overall game play work well?

A. Experimental Setup

To perform the experiment, we released the VeriLinks survey and platform. Both were announced on Semantic Web mailing lists, which led to a lively discussion on the subject. We recorded in-game statistics and survey results over 3 days in which 118 players used VeriLinks, specifically the pea invasion game, and 42 of those completed the survey. All survey questions and results, the used link specifications and linksets as well as the source code of the game and its configuration files are available at http://aksw.org/projects/VeriLinks. Using this material, equivalent experiments can be reproduced in the future.

The linksets we picked are displayed in Table II. They were selected from the LATC link specification repository[10]. We selected linksets, which a general audience can understand. For all three datasets, we created a gold standard by using the existing link specifications from the LATC repository, but lowering the confidence threshold in those specifications. This results in a linkset with high recall, i.e. it is likely to contain most or all of the links, which could actually be created.[11] We reviewed all resulting links individually by looking up the RDF information for subject and object as well as performing web searches when necessary. In a first phase, obvious correct and incorrect links were detected. More complex cases were reviewed by a second person. The gold standard is available on the VeriLinks project website. Note that we selected link tasks of a size, which still allow a manual evaluation and creation of a gold standard. In practice, the scalability is mostly limited by the underlying store, i.e. the game should scale to millions of links.

We created templates from the available data for subjects and objects. The templates were kept as simple as possible, since a previous internal evaluation phase revealed that players are overwhelmed when they are presented detail information while having to play the main game simultaneously. In detail, we asked users to match pictures of animals, map a flag of a country to a country displayed on a map and to map languages to countries.

For estimating link confidence, we adopt Equation 1: In this case, n is the number of times a player made a decision on a link (confirming or rejecting). $\hat{p}$ is the number of times a player has confirmed the link divided by n. This means that we now have two confidence measures for each link derived from link specification and the game, respectively.

B. Results

In order to evaluate the above described confidence measures, we need to define a threshold: Everything above the threshold is judged to be a correct link and everything below is classified as incorrect. In order not to choose an arbitrary threshold, we computed F-Measures for all thresholds and plotted them in Figure 5 for links, which were evaluated in

[7] This approach is very similar to the computation of Cohen's Kappa.

[8] Note that we require a login in VeriLinks, so previously validated links may be from a previous session.

[9] This technique is used in reinforcement learning and various search algorithms.

[10] https://github.com/LATC/24-7-platform/tree/master/link-specifications

[11] To be certain not to miss a possible link, we would have to analyze the cross product of resources belonging to the interlinked classes in source and target knowledge base, which is not feasible given the size of the datasets.

name	number of links	source class	target class
DBpedia-LinkedGeoData	204	dbpedia-owl:Country	lgdo:Country
DBpedia-Factbook	142	dbpedia-owl:Language	factbook:Country
DBpedia-BBCwildlife	108	dbpedia-owl:Species	bbc-wildlife:Species

TABLE II
INFORMATION ON THE TESTED LINKSETS.

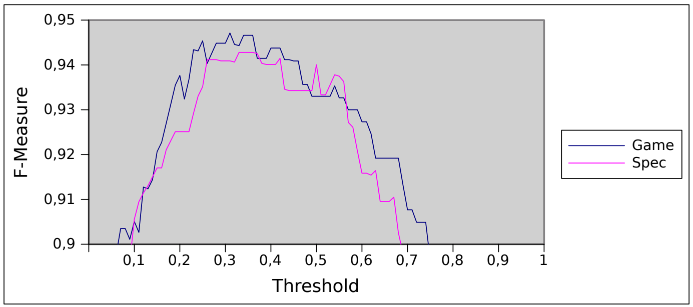


Fig. 5. F-Measure for different thresholds in the [0,1] interval.

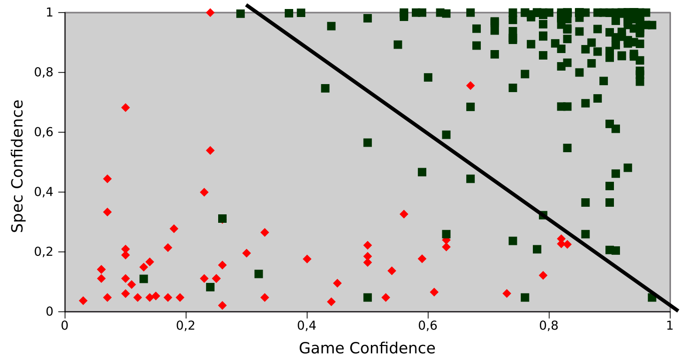


Fig. 6. Scatter plot of game and spec confidence values: Correct links are shown as green squares and incorrect links as red diamonds. A classifier with high precision is shown as black line.

measurement	value
spec F-Measure (original, no threshold optimisation)	0.711
spec F-Measure (SVM, 10 fold CV)	0.914
game F-Measure (SVM, 10 fold CV)	0.893
combined F-Measure (SVM, 10 fold CV)	0.936

TABLE III
EVALUATION RESULTS SUMMARY.

criterion	value
number of players	118
number of distinct evaluated links	454
number of agreements	4738
number of disagreements	1053
number of times unsure option has been selected	507
total hours played	17.24 h
average playing time per session	6.73 min

TABLE IV
USAGE STATISTICS FOR VERILINKS DURING THE 3-DAY EVALUATION PHASE.

the game at least 5 times. The figure shows that the game and the link specifications achieve similar performance on the given tasks. We expected that the game would not outperform the specs, because the specs were of good quality and we intentionally minimised the information shown to the gamer while playing. The goal should not be to maximise accuracy of the game predictions itself, but maximising the combination of link spec predictions combined with game predictions. This reduces the information load players have to deal with and we obtain orthogonal additional information, e.g. we do not want players to handle tasks like judging spatial distance, which can be accurately done in link specs. To visualise the correlation between game and link spec confidence values, we created a scatter plot depicted in Figure 6. The plot shows that the combination of both confidence values allows very precise predictions. In fact, it can be seen that there is a simple linear separation, i.e. a classifier, which does only a single error while still achieving remarkably high recall.

We also confirmed this by running a support vector machine classifier using the two confidence values as input. This classifier outperformed the optimal thresholds for each single criterion in ten fold cross validation as shown in Table III. Specifically, we ran the `SMO` algorithm from the Weka machine learning toolkit[12] with default settings.

In addition, we also collected usage statistics for the game in the 3-day evaluation phase, which are shown in Table IV. We consider the number of judgements made by players to be sufficiently high to lead to statistically significant results.

C. VeriLinks Survey

The survey was divided in two parts: In the first part, we asked personal questions and in the second part questions on the game experience with the pea invasion game. Overall, 51 people filled the survey with 42 of them answering all questions. The list of questions and full survey results are available on the VeriLinks homepage.

Personal Questions:: People who filled the survey mostly just play a little every few months (38%). 22% of them play each day and 12% more than 3 hours per day. 33% of all players would play games for public benefit even if the games are less entertaining than usual. Only 1 player did not care about public benefit and all the remaining players said that they would play GWAPs if they provide at least some entertainment value. 67% of the players had a Semantic Web background, but the game also attracted users outside of academia.

Game Questions:: Matching animals (62%) was the most popular game task, followed by flags (40%) and languages (28%), which shows that games with a purpose involving images tend to attract interest. 43% of the players understood immediately what to do, whereas 43% required some time to understand the game. We provided a game tutorial, which was, however, not interactive and intuitive enough for all players. About 62% of the players said the game was ”quite good” or ”a pleasure to play”, 38% did ”not really” enjoy it and no one said that it is among the worst games. From the comments on that question, reasons for dissatisfaction were that it is hard to

[12] http://www.cs.waikato.ac.nz/ml/weka/

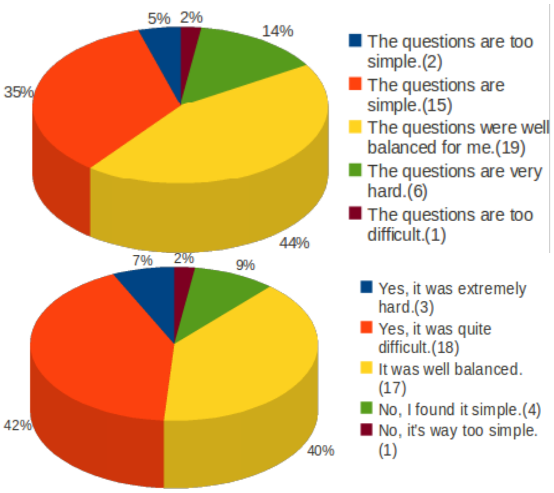


Fig. 7. Question and overall game difficulty.

handle two tasks simultaneously (validating and playing) and browser issues caused by the underlying software framework. The results on question and game difficulty are shown in Figure 7. The questions appear to be well balanced on average with only few people who found it too simple or too hard. For the overall game, players judged it to be quite challenging due to the simultaneous tasks, which made the game interesting for some players, but also discouraged others. In the last question, 60% of the players said that they would probably or certainly play the game again. 22 players (21%) did actually already return and play again within the 3 day evaluation phase.

V. Discussion

We reached the following insights on the evaluation objectives mentioned at the beginning of Section IV:

1) *accuracy of game based linking:* In our evaluation, hand crafted specifications show similar performance to game based linking when selecting optimal thresholds based on a gold standard. Please note, however, that we presented only very limited information to players in the game as discussed before, so this result is encouraging.
2) *combination of initial specs and games:* It seems to be very promising to combine link specifications with game based linking as shown in Figure 6. If orthogonal criteria are used in game and specs, the combination of both can lead to highly precise linkage. Our vision in future work is to integrate an unsupervised machine learning approach into VeriLinks, which creates initial specifications, which are then validated and improved in the game. In our observations, a game based approach can turn a linkset with high recall and low precision into a linkset, in which both precision and recall are high.
3) *player interest:* Only very few players were not interested in games with a purpose and people were very interested in the topic on the lists, which we used for dissemination. More than 20% of the players returned within 3 days and the average time per session also indicates that most players were not just superficially looking at the game. More than 5000 evaluation tasks were performed, which would otherwise have been very time consuming. (The time required for compiling the gold standard was 10 hours.)
4) *balancing:* The question balancing did work well as indicated in the survey. The frequency of times each link was played is quite uniform, so the balancing does not have a severe negative impact on the number of distinct links played. For the overall game play, results were generally encouraging, but areas for improvement have been identified (see discussion below).

While we achieved our main objectives, we also learned valuable lessons from the survey and the game feedback. For instance, some players found it too difficult to focus on two things at once. This feedback was incorporated into the second ”space ships” VeriLinks game, which offers a turn based approach with alternating validation and playing phases. We also noted limitations of games based on user agreement. For instance, similar looking flags were sometimes incorrectly assigned to a country. Since most users make this mistake systematically, it will persist even in the long run. This also happened with very similar looking animals. In few cases, the original sources are not perfect: If incorrect images are displayed, because of errors in the source data, this also leads to misjudgements in the game. Overall, however, those cases were very rare and most of them are not critical when combining the game validation results with the spec results, which are usually based on criteria like string matching, spatial distance and not image similarity.

VI. Related Work

The games with a purpose idea developed by Luis von Ahn aims to entice humans into playing games that will help computers get smarter. He has shown that masking useful tasks in entertaining online games can motivate large user groups to work on these tasks for free [8]. The website gwap.com presents some of his games that can help improve internet image, video and audio searches, enhance artificial intelligence and teach computers to see. One of the first GWAPs developed by Luis von Ahn is the ESP game. It was an effort to make two random players assign the same labels to an image to facilitate image searches. Another GWAP designed to improve image searching is the Matchin game. By letting players judge which of two images is more appealing, Matchin could eventually enable image searching ranked on which image looks best. Also included into gwap.com is Tag a Tune, an approach to let players annotate music other than by title or interpret. Luis von Ahn’s latest project is called Duolingo.[13] With Duolingo, the user learns a language for free and simultaneously helps to translate websites and other documents. Duolingo will show the user a sentence in the language he wants to learn and asks him to translate it to his native language. While taking the skill level of the user into account, Duolingo also provides translations for words the user doesn’t know.

A project adopting the findings of Luis von Ahn’s GWAPs in the semantic web context is OntoGame[4]. It includes a

[13] http://duolingo.com

series of games that aim at building and aligning vocabularies. OntoPronto is a two player quiz game of the OntoGame series, where both players try to map randomly chosen Wikipedia articles to the most specific class of the Proton ontology. If they agree on a Proton class for this article they get points and can proceed with the next specific level. This game was designed for building a Proton grounded domain ontology. SpotTheLink is another two player quiz game of the OntoGame series, which focuses on the alignment of DBpedia concepts to the Proton upper ontology. A recent game of the series is called sLODMachine and is the successor of SpotTheLink. Players will get a triple-based challenge in the sLODMachine and have to decide whether the statement makes sense or not. Apart from that, sLODMachine contains many more usage scenarios such as annotating images with concepts, verifying the validity of existing statements and others. The website semanticgames.org serves as a comprehensive overview of the currently available semantic games. Another recent releated game is Urban Match [9], a location based game for mapping locations in cities.

The main difference of VeriLinks to the above games in the semantic web area is that it allows arbitrary instance level interlinking. It provides high flexibility as well as a reward and balancing system, which allows to play new linksets without the need to balance them manually. Due to the rising importance of interlinking over the past years, we consider game based approaches, such as VeriLinks, to become an attractive area for research and applications. At the start of the VeriLinks project, the OntoGame/insemtives toolkit did not have the features required by VeriLinks, but many of those have been added recently [10], which renders the toolkit an attractive choice for VeriLinks in the future.

There is a body of research work in game balancing. For instance, the line of research around [11] investigates reinforcement learning approaches. In VeriLinks, however, we do not have a state based game representation, which is usually the case for reinforcement learning, nor do we have an a priori classifier as obtaining a classifier is the underlying goal of the game. Optimisations for state based game representations are also discussed in other works such as [12]. In general, most prior work in the artificial intelligence such as [13] assumes an existing game AI, which is then balanced to play at a particular level, which is a different setting than the balancing in VeriLinks. Work around [14] and several others focus on other types of games. For instance, in many multiplayer online role games, the goal is to balance the effect and availability of certain items.

VII. Conclusion and Outlook

We presented an experiment, which shows that game based interlinking can be useful to increase precision and recall of interlinking. To do this, we implemented the VeriLinks platform, two game prototypes and executed a survey and in-game study. Our main result is that games need to be designed in a way, which either exploits the strengths of the human brain or draws on existing expert knowledge. Our preliminary studies indicated that using the same mechanisms in link specifications and in games, for instance comparing string labels of resources via string metrics and via humans, does not deliver a significant benefit. However, presenting players images or content they are knowledgable about and intentionally hiding other relevant information like string labels, which can be effectively used by machines, did lead to more promising results when combining both sources of evidence. Thus, we argue that expert crafted or learned link specifications in combination with carefully selected games can provide a very effective mechanism for high quality interlinking.

Please note that all material to reproduce our experiment and the survey is available from the VeriLinks project homepage. From a technical point of view, we plan to experiment with further game mechanics, e.g. using elements of Mahjong. We also aim to provide more boilerplate code for developers to bootstrap the creation of HTML5 and Javascript based games. In addition to uploading already generated linksets to VeriLinks, we also aim to integrate tools like SAIM [15], which allow to learn link specifications via active learning using only minimal user input. Another target is a better integration of social web features to provide additional incentives for playing the games.

References

[1] A. Ferrara, A. Nikolov, and F. Scharffe, "Data linking for the semantic web," *Int. J. Semantic Web Inf. Syst*, vol. 7, no. 3, pp. 46–76, 2011.

[2] A.-C. Ngonga Ngomo, J. Lehmann, S. Auer, and K. Höffner, "Raven – active learning of link specifications," in *Proc. of OM@ISWC*, 2011.

[3] K. Siorpaes and M. Hepp, "Games with a purpose for the semantic web," *IEEE Intelligent Systems*, vol. 23, no. 3, pp. 50–60, 2008.

[4] ——, "Ontogame: Weaving the semantic web by online games," in *Proc. of ESWC*, ser. Lecture Notes in Computer Science, vol. 5021. Springer, 2008, pp. 751–766.

[5] L. von Ahn and L. Dabbish, "Designing games with a purpose," *Commun. ACM*, vol. 51, no. 8, pp. 58–67, 2008.

[6] L. D. Brown, T. T. Cai, and A. Dasgupta, "Interval estimation for a binomial proportion," Jul. 30 1999.

[7] G. Andrade, G. Ramalho, A. S. Gomes, and V. Corruble, "Dynamic game balancing: An evaluation of user satisfaction," in *AIIDE*, J. E. Laird and J. Schaeffer, Eds. The AAAI Press, 2006, pp. 3–8.

[8] L. von Ahn, "Games with a purpose," *IEEE Computer*, vol. 39, no. 6, pp. 92–94, 2006.

[9] I. Celino, S. Contessa, M. Corubolo, D. Dell'Aglio, E. D. Valle, S. Fumeo, and T. Krüger, "Urbanmatch – linking and improving smart cities data," in *Proceedings of the Linked Data on the Web Workshop (LDOW)*, 2012.

[10] E. Simperl, S. Wölger, S. Thaler, B. Norton, and T. Bürger, "Combining human and computation intelligence: the case of data interlinking tools," *International Journal of Metadata, Semantics and Ontologies*, vol. 7, no. 2, pp. 77–92, 2012.

[11] G. Andrade, G. Ramalho, H. Santana, and V. Corruble, "Automatic computer game balancing: a reinforcement learning approach," in *Proc. of the international joint conference on Autonomous agents and multiagent systems*. ACM, 2005, pp. 1111–1112.

[12] A. Jaffe, A. Miller, E. Andersen, Y.-E. Liu, A. Karlin, and Z. Popovic, "Evaluating competitive game balance with restricted play," in *Proc. of AI and Interactive Digital Entertainment*, 2012, pp. 26–31.

[13] P. Spronck, M. Ponsen, I. Sprinkhuizen-Kuyper, and E. Postma, "Adaptive game ai with dynamic scripting," *Machine Learning*, vol. 63, no. 3, pp. 217–248, 2006.

[14] R. Hunicke and V. Chapman, "Ai for dynamic difficulty adjustment in games," in *Challenges in Game Artificial Intelligence AAAI Workshop*, 2004, pp. 91–96.

[15] K. Lyko, K. Höffner, , R. Speck, A.-C. Ngonga Ngomo, and J. Lehmann, "Saim - one step closer to zero-configuration link discovery," in *Proc. of the Extended Semantic Web Conference Posters & Demos*, 2013.