Department of
Information Engineering
and Computer Science
DISI

UNIVERSITY
OF TRENTO - Italy

DISI - Via Sommarive 14 - 38123 Povo - Trento (Italy)
http://www.disi.unitn.it

Knowledge Maintenance Via Word Games

Aliaksandr Autayeu, Fausto Giunchiglia, Mladjan Jovanovic

April 2015

Technical Report # DISI-15-011
Version 1.0

Knowledge Maintenance Via Word Games

Aliaksandr Autayeu, Fausto Giunchiglia, Mladjan Jovanovic

DISI, University of Trento, Italy

Abstract. We examine gamification as a mean to handle Linked Data quality problems that are difficult to solve in an automated way. In particular, we look at the use of word games as a knowledge maintenance strategy that is cost-efficient and accurate in terms of the level of granularity of the errors to be fixed. We have classified the most common quality problems encountered in our knowledge base — Entitypedia. Based on this classification, we have implemented a quality improvement methodology for knowledge maintenance that leverages on gamification. We empirically evaluated how this methodology could efficiently improve data quality in Entitypedia. The results show that gamification-enabled knowledge maintenance is a promising and affordable way to improve the quality of Linked Data. A subset of the games described in this paper can be played at Entitypedia Games web site[1].

1 Introduction

One of the critical aspects of Linked Data success is related to the varying quality of its sources that results from the combination of data content and data import procedures. This often poses serious problems to developers aiming to seamlessly consume the data in their applications. Linked data sources that are transformed into linked form are highly heterogeneous in terms of structure, format and vocabulary. Some of the quality issues (e.g. missing values) can be easily repaired automatically, but others require manual intervention.

Entitypedia is our knowledge base that is populated from external datasets with typical knowledge management tasks such as ontology alignment, semantic matching [1], and natural language processing. It describes ground knowledge in terms of entities, namely representations of real-world objects that can be found in different contexts in our everyday life. Each entity is described with a set of attributes and relations with other entities. In this respect, Entitypedia includes a schema for describing entities (TBox) and entities themselves (ABox). Currently it contains around 10M entities described with 80M facts (attributes and relations). It has been incrementally populated with domain knowledge from external data sources that include GeoNames [2], Wikipedia [3], YAGO [2], WordNet [4], MultiWordNet [5] and GeoWordNet [3]. Although it contains clean data, there is a large amount of mistakes and furthermore, corrections are needed

[1] Entitypedia Crosswords: http://games.entitypedia.org/crosswords/
[2] http://www.geonames.org/
[3] http://www.wikipedia.org/
[4] http://wordnet.princeton.edu
[5] http://multiwordnet.fbk.eu/

to keep this knowledge up to date. We have evaluated an average correctness of 98% [4]. It means that the number of mistakes is around 1.6 M.

We have identified two types of mistakes in Entitypedia: (i) typos (mistakes in syntax) and (ii) disagreements (mistakes in meaning). We look into gamification [5] as a cost-efficient way to correct these mistakes. In particular, we use word games because they are plenty and they are popular. Word games can take different forms, but have many common elements, such as clue-answer pairs. Furthermore, aside from textual data, word games can easily handle other types of media, such as images. We have designed a game framework that implements such common elements. On top of it, we have implemented a set of well-known word games, such as Hangman and Crosswords. The content enabling to play them is imported from Entitypedia (as triples) and adapted to a form suitable for gameplay (as clue-answer pairs). The word games we have implemented have been modified as players are expected to fix the mistakes they discover. For this reason, we measure players' reputation for quality control of provided corrections.

We have empirically evaluated how word games could be efficiently used to improve linked data quality in Entitypedia. The results show that using word games is a promising and affordable way to enhance the quality of Linked Data, which, in the long run, may address many of the problems that fundamentally constrain the usability of Linked Data on the Web in real-world applications.

The rest of the paper is organized as follows. In Section 2 we describe the problem in the context of our entity-centric data management framework. Section 3 describes the activities behind the data certification pipeline. Section 4 illustrates the crosswords word game. Then we describe how we measure players' reputation in Section 5. In Section 6 we describe the evaluation we have performed. A comparison with relevant related work is provided in Section 7. Finally, Section 8 concludes the paper.

2 Mistakes in Linked Data

Entitypedia is a multilingual knowledge base [6]. Its data model is defined following the faceted approach to organize and represent knowledge and is focused on the notion of domain. A domain is defined as a 4-tuple <C, E, R, A> where:

- C is a set of classes,
- E is a set of entities,
- R is a set of binary relations,
- A is a set of attributes.

Knowledge is organized around entities. Entities are representations of real-world objects that can be found in different contexts in our everyday life (**Fig. 1**). Each entity is described with a set of attributes and relations with other

entities. It has a reference class that actually determines its type. An entity type is defined in terms of attributes, relations (such as born-in, part-of), services (such as computeAge or computeInverseRelation) and categories of metaatributes (such as mandatory, identifying, permanent, timespan, provenance). There are relatively few common sense entity types (such as person, event) and many application and context dependent entity types.

While the combination of machine-driven extraction and human effort is a reasonable approach to produce a baseline version of the resource, there are data quality problems. The quality issues mainly come from low quality of source data. The data can be incorrect, partially overlapping and inconsistent, or incomplete. Low quality may also result from the data import procedure, such as incorrect or incomplete extraction of objects, incorrect extraction of data types or incorrect links to external resources. All these factors largely influence the quality of the services that can use Entitypedia, such as different search, navigation and exploration applications.

By analyzing Entitypedia content at the level of triples (**Fig. 1**), we can find only two basic types of mistakes:

- **Typos** – refer to mistakes in syntax that can appear in an entity name (**Fig. 1a**), in the relation or attribute name or value;
- **Disagreements** – describe mistakes in meaning that can be found in an entity, relation or attribute value (**Fig. 1b**).

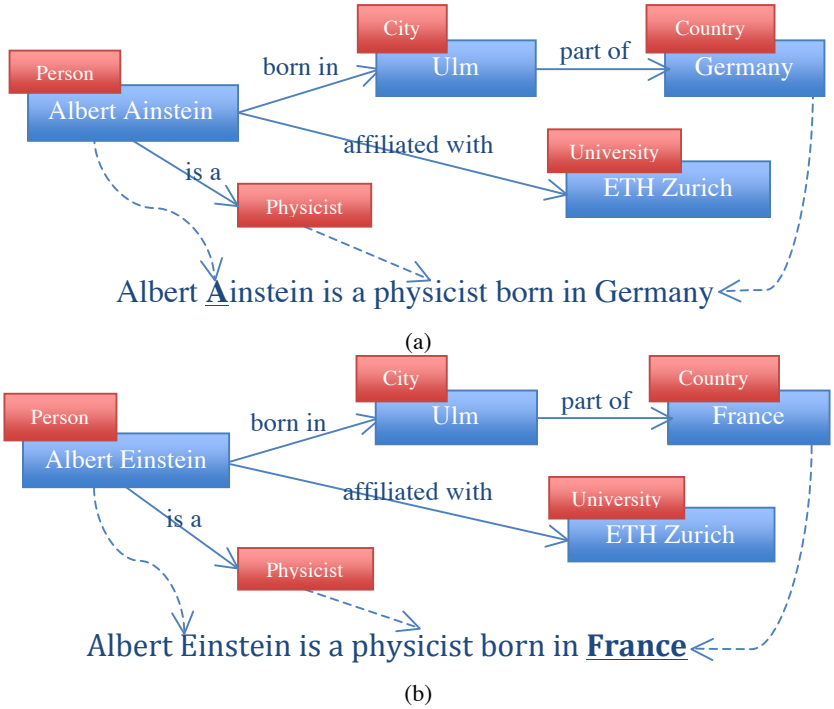

Fig. 1. Simplified examples of mistakes that can be found in entity-centric knowledge graph, (a) syntax-level mistake (typo) and (b) semantic-level mistake (disagreement).

3 Data certification pipeline

We maintain knowledge via a data certification pipeline. This pipeline is organized as a five-stage process, defined as follows (**Fig. 2**).

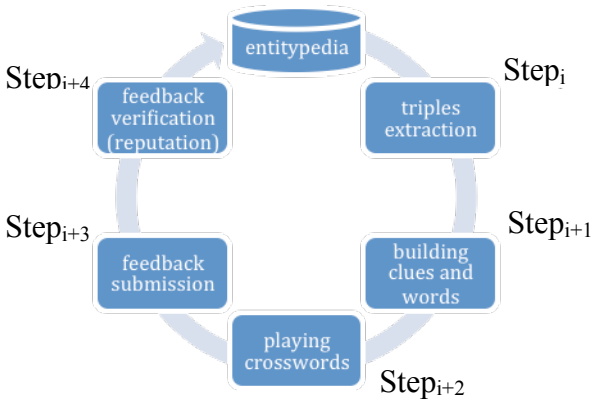

Fig. 2. Data certification pipeline. Entity attribute-value pairs are expressed as word-clue pairs to be used in word games.

The general idea is to embed data directly into crosswords and have such games solved by many players. We have Entitypedia repository as the source of data for certification. The activities are organized as follows:

Step $_i$ – we take the Entitypedia content and convert it into different triple forms, such as *entity-relation-attribute* or *attribute-relation-value*. Triples provide necessary information to generate game content (**Fig. 3**). Triple extraction allows for generality and flexibility in selecting specific data configurations to certify.

Step $_{i+1}$ – triples are transformed into clue-word pairs that are common for many word games (**Fig. 3**). From there, games select the content and present it to players.

Step $_{i+2}$ – refers to gameplay (see Section 4). There are two basic types of gameplay:

- Creating crossword puzzles, and
- Solving crossword puzzles.

Step $_{i+3}$ – during gameplay players submit feedback, namely spot mistakes in words and clues and provide corrections, or confirm the correctness implicitly while playing (see Section 4.1, **Fig. 6**).

Step $_{i+4}$ – quality of the feedback is measured against players' reputation. The feedback verification stage corresponds to a final quality control of the feedback

(described in Section 5).
Fig. 3 illustrates the principal way of using data for crossword puzzles. It shows an excerpt of the knowledge graph, which contains three entities: an entity representing a person, Leonardo Da Vinci, a city of Florence, where he was born, and a country, Italy, which Florence is part of. The entity graph displays also entity types. All this information can be used to create clue-answer template. Such template describes a particular configuration of entities in the graph and contains a textual phrase with blanks. Usually there are many configurations where such template applies. An example would be as per **Fig. 3**, the phrase "an X born in Y", where X is constrained to be an entity type Person, with X representing the profession, and Y being the location where the Person was born. In our example, the X is Leonardo Da Vinci the artist and the Y is Italy the country.

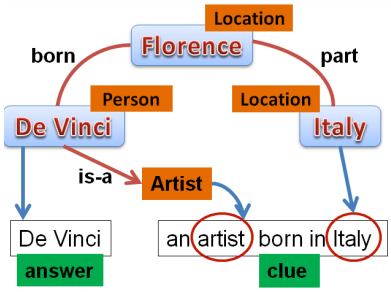

Fig. 3. Template for mapping triples from entity-centric knowledge graph into word games primitives (word and clues).

To implement the pipeline, we have designed and developed a set of tools organized as word games framework. The framework contains common components (content management, feedback management, reputation management) and a set of word game implementations.
The word games framework implements data flow described below (**Fig. 4**).

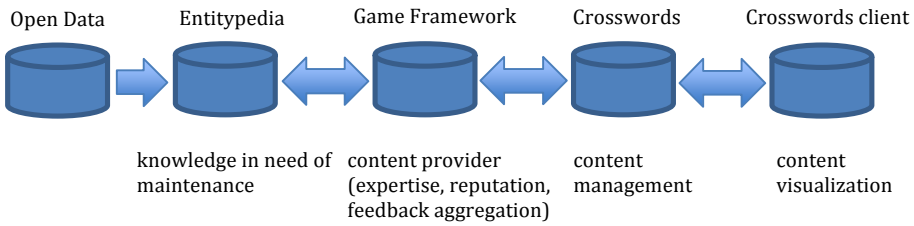

Fig. 4. Word games framework data flow.

Entitypedia provides content for games, namely, the data for verification. As explained in the Introduction, it is populated from existing Open Data ources.
The Game Framework provides content for word games and several services, such as word, clues and feedback management. Entitypedia content is represented by *words* and *clues*, which are created from entities using *templates*

(**Fig. 3**). The template is a starting point for content generation. It identifies certain entity configurations in the knowledge base and extracts about such configuration enough information to generate words and clues. **Words** [6] in the game framework include common words ("apple"), parts of words ("clue" like in "clueless") and multiwords ("SanVigilio") with spaces removed. The information about the latter two categories is kept in the respective clues. **Clues** [6] can be textual or media. Textual clues are well-known type of clues, such as "country in Europe" with an answer "italy". Media clues, in addition to textual clue, contain an image (URL of the image) which provides the visual hint.
The Crosswords Client visualizes generated content and implements interaction with the player (see Section 4). Namely, the direction from Entitypedia to games represents the flow of data to be verified and the reverse direction represents the flow of feedback on such data.

4 Crosswords

The Crosswords game allows players to create or solve word puzzles, identify mistakes in clue-answer pairs and submit feedback according to a predefined data certification pipeline.
Feedback can be:

- *Explicit feedback, and*
- *Implicit feedback*.

Explicit feedback refers to spotting and fixing mistakes in words and clues. It is used to provide corrections and, based on the correction, is collected per word-clue pair. Any element in the pair can be incorrect and fixed as such. *Implicit feedback* is a result of playing the game and giving correct answers (implicit positive feedback) and incorrect answers of the correct length (implicit negative feedback). *Implicit feedback* corresponds to confirming the correctness of the content. It results from a player's actions in the game. For example, correct guess from the first attempt without any assistance indicates the player's confidence in the knowledge triple used to generate that clue-answer pair. As such, positive feedback is used to measure player's reputation. In Section 5 we describe how we calculate reputation in Crosswords game.
Our implementation of Crosswords exploits common features of crossword puzzles. However, we have implemented additional features needed for knowledge maintenance, such as fixing mistakes and reputation-based quality control. There are two principal ways of engaging the players in data certification: crosswords creation and crosswords solving.

4.1 Crosswords creation

[6] Game Framework API to access words and clues: http://games.entitypedia.org/apidocs/

The game allows creating crossword puzzles in two ways: manual and assisted mode (**Fig. 5**). The manual mode relies only on the knowledge of the author. That is, the author creates or picks the layout, creates the grid and writes the clues. Assisted mode of creating a puzzle is computer-aided. In this mode, the involvement of computer is that of helping the author, rather than substituting the author.

The authors are usually very attentive during puzzle creation and therefore are more likely to report errors in the content they use to build the puzzle. Assisted crossword editing features a tool to choose words, which fit the grid and choose clues for words. This tool draws its content from Entitypedia. The entities from the repository are transformed and displayed to the authors in the clue-answer format (**Fig. 5**).

In **Fig. 5**, coloured bars next to each word denote difficulty. Word frequencies are a convenient measure of word familiarity. We use word frequencies from the Google NGgram corpus [7] to measure the words difficulty. Raw frequencies need more simplification to be used with ease. Therefore we normalize them into difficulty levels, from very easy words, recognizable by pretty much anybody, to very hard words, recognizable only by the individuals with extensive knowledge. As for the clue difficulty, we use following criteria:

- the familiarity of words used in the clue itself;
- the amount of information included in the clue, such a number of entity attributes used to generate the clue.

The mistakes can be reported using feedback submission form (invoked by clicking on the exclamation triangle icon next to the clue).

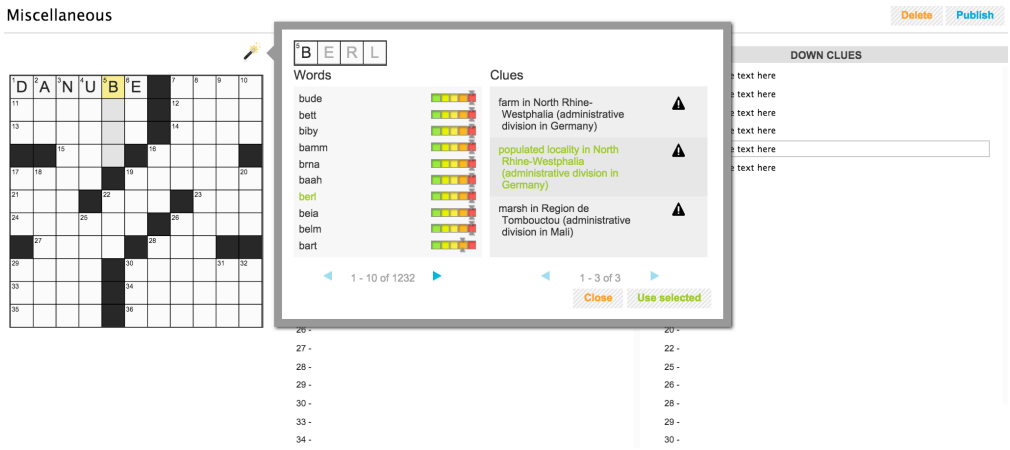

Fig. 5. Crosswords creation — tool for assisted clue editing.

Fig. 6 shows the feedback submission form for the word-clue pair. The form displays information following the original template used for the creation of the word-clue pair. The word or the clue can be marked as wrong and corrections are provided in the respective fields. In case of creating crosswords we take into account only explicit feedback submitted by players using the form.

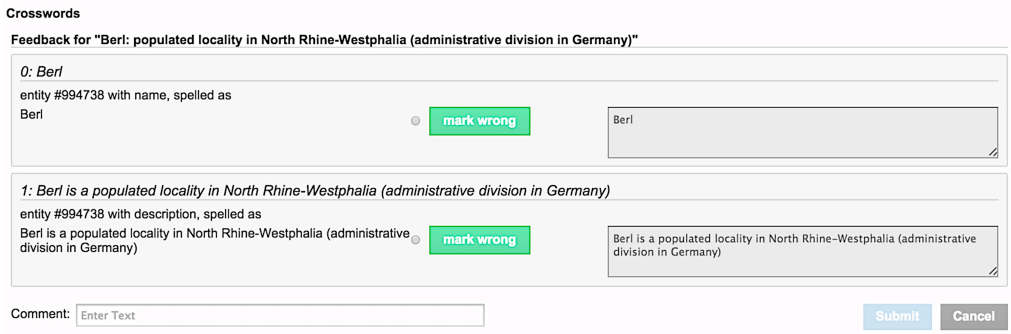

Fig. 6. Feedback submission form.

4.2 Crosswords solving

The second way to certify the data is to actually solve crossword puzzles (**Fig. 7**). When solving a crossword puzzle, players report errors. Namely, while a player is solving the puzzle, active processing of information during answer search loads the puzzle into short-term memory and makes it easier to notice discrepancies in the data. Crossed clues give players hints and make guessing a word easier.

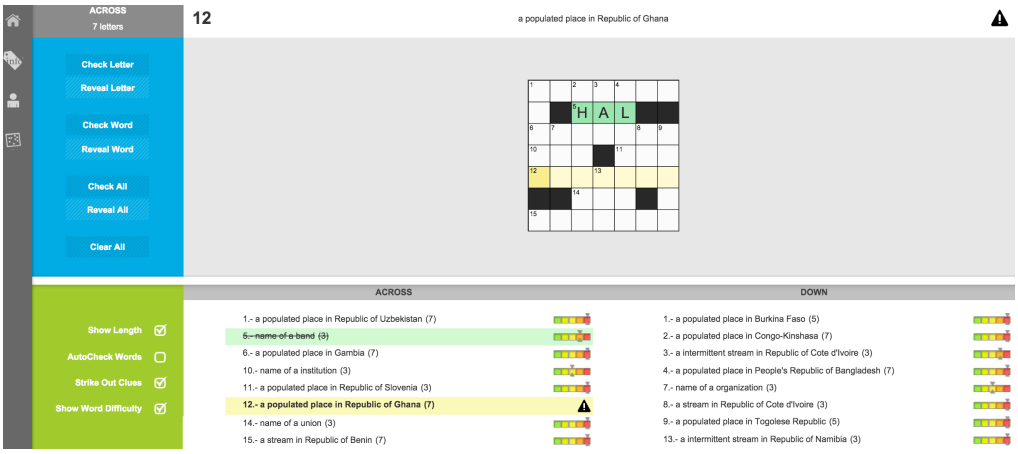

Fig. 7. Basic interface for solving crosswords.

In case of solving crosswords, players can provide both types of feedback. Explicit feedback is provided in the same way as explained in creating crosswords (the exclamation triangle icons next to the clue in focus). Implicit feedback collection consists of recording the player's actions while solving the puzzle, that is, typing in answers.

5 Computing Reputation

We measure player's reputation by calculating two characteristics: confidence and correctness. Confidence measures how well the player knows the fact used as a basis for the clue. Correctness shows how correct the player is overall. We use implicit and explicit feedback to calculate the characteristics.

5.1 Computing Confidence

Let us consider a typical case in a game, a single word as from below (**Fig. 8**).

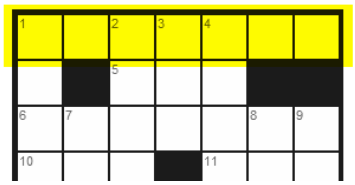

Fig. 8. Fragment of puzzle layout.

Crosswords can contain letters that belong to two clues (checked letters), and letters that belong to one clue only (unchecked). For example, a top left cell contains a checked letter: across and down clues check it. The letters next to the top left cell, both across and down are unchecked: they only belong to one clue each. The ratio of checked to unchecked letters varies by crosswords style. Unchecked letters should be known, guessed or revealed, while checked letters can be guessed by a crossing clue. Players also can reveal both kinds of letters, separately as a single letter or as a part of a revealed clue.
We assume players in word games:

- fill known words first;
- complete known words in streaks (uninterrupted sequences of keystrokes);
- need less assistance (checked letters) for known words.

In addition, we know both clue and word difficulty. We can calculate player confidence in the fact expressed by the triple (*entity_id, attribute_id, attribute_value*) by the length of "winning streak". That is, how many letters user typed sequentially without distractions to other letters minus letters present in the grid before streak start. We then discount that by clue difficulty. Harder clues are usually more "distant" from the right answer. In such cases players might need more help recalling the answer even if they know it confidently. We account for difficult by discounting 30% of known letters for each level of difficulty, word and clue combined. Therefore our discount goes from 0 to 30% of known letters, rounded up to the nearest integer. In other words, in the hardest clue and answer possible, a player can reveal up to 30% of word letters before

finishing the word and still be considered knowing the word confidently. This leads us to the following formula for the confidence:

$$confidence = \begin{cases} 1 - \dfrac{kL * (1 - 0.03 * d)}{wL}, & if\ kL < wL \\ 0, & if\ kL = wL, \end{cases}$$

where kL (known letters) means the amount of letters visible before the winning streak (completion of the word), d is a sum of word and clue difficulty and varies from minimum 2 to maximum 10 and wL is the word length. Confidence varies from 1 (meaning fully confident, which is the case when the answer was typed in without any assistance) to 0 which is the case when the answer was revealed. Confidence is assigned per-user per statement, that is, per-user per triple (*entity_id, attribute_id, attribute_value*). **Table 1** illustrates the confidence formula in action.

Table 1. Confidence values for 5-letter words of various difficulties.

difficulty	**known letters**					
	0	1	2	3	4	5
2	1.0000	0.8120	0.6240	0.4360	0.2480	0.0000
3	1.0000	0.8180	0.6360	0.4540	0.2720	0.0000
4	1.0000	0.8240	0.6480	0.4720	0.2960	0.0000
5	1.0000	0.8300	0.6600	0.4900	0.3200	0.0000
6	1.0000	0.8360	0.6720	0.5080	0.3440	0.0000
7	1.0000	0.8420	0.6840	0.5260	0.3680	0.0000
8	1.0000	0.8480	0.6960	0.5440	0.3920	0.0000
9	1.0000	0.8540	0.7080	0.5620	0.4160	0.0000
10	1.0000	0.8600	0.7200	0.5800	0.4400	0.0000

5.2 Computing Correctness

To calculate correctness we introduce a ground truth (GT) and extrapolate player correctness on the ground truth (where we can check it) to player contributions. To introduce ground truth, we mark certain amount of statements, that is, triples (*entity_id, attribute_id, attribute_value*) as definitely correct (positive) or definitely wrong (negative). These marked statements will constitute our body of ground truth. We should note that marking the statement correct means we have a correct answer. However, marking the statement wrong means that the provided answer is wrong, but the correct answer might be unknown or not provided. We expect that the amount of negative ground truth is negligible, based on the estimation of knowledge base correctness at 98%, nevertheless, we consider such cases.

Having established the ground truth, we take into account both kinds of feedback: implicit (coming from gameplay) and explicit (coming from report error form) to calculate player correctness. Let us consider possible cases:

- *Positive ground truth (GT+)*. Suppose we have a positive ground truth

statement “Rome is a capital of Italy”, and the clue-answer pair “capital of Italy”-“Rome”.
    - *explicit feedback (EF)*
        - *correct (EF+):* ignored; player feedback: “Rome is a capital of Italy”. This is a repetition, probably not intentional, and it is discarded.
        - *incorrect (EF-):* decreases correctness; player feedback may vary from simpler “Rome is NOT a capital of Italy” to more specific “Paris is a capital of Italy”. Since this is the ground truth, the player is wrong here and therefore we decrease player correctness value.
    - *implicit feedback (IF)*
        - *correct (IF+):* increases correctness; player typed “Rome” as an answer. This is correct and player correctness value is increased, taking confidence into account.
        - *incorrect (IF-):* ignored; player typed “Baku”, then “Oslo”. This input is wrong and it is ignored, interpreted as guesses.
- *Negative ground truth (GT-)*. Suppose we have a negative ground truth statement “Rome is a capital of Greenland”, and the clue-answer pair “capital of Greenland”-“Rome”.
    - *explicit feedback (EF):* increases correctness; player reports the error, marking the clue-answer pair as wrong and (optionally) provides correct value. We increase the correctness because the player has spotted known mistake.
    - *implicit feedback (IF -> EF):* ask for explicit user feedback; player types “Rome”, either by belief (mistakenly) or to fill the grid and check the clues. It’s not possible to tell these apart, therefore at this point the game can tell the player that the answer “fits the grid” and from the game point of view (points, bonuses, achievements, etc) considered correct but is inherently incorrect and ask for correct value.

For positive ground truth explicit correct feedback is ignored, because it is just repetition and is not relevant. Explicit incorrect feedback decreases correctness. Implicit correct feedback increases correctness, taking confidence into account. Implicit incorrect feedback is ignored, as it might represent guesses.

For negative ground truth, explicit feedback increases correctness and provides confirmation of the error. Implicit feedback might be used to generate a hypothesis to test. In this case, implicit feedback might be interpreted as player’s attempt to provide the correct value (which might not fit the grid).

The above categories count as positive or negative, discounted or full votes in user correctness. The correctness itself is a share of correct answers and varies from 0 to 1. The users without any intersection between their feedback and ground truth are assigned a neutral correctness value of 0.5.

With respect to the cases above, we have a formula to calculate overall player correctness using player feedback on ground truth:

$$Correctness = \frac{\sum_{i \in GT^+} Conf_i * IF_i^+ - \sum_{i \in GT^+} EF_i^-}{Count(GT^+)} + \frac{\sum_{i \in GT^-} EF_i}{Count(GT^-)},$$

where GT^+ (GT^-) is player feedback on positive (negative) ground truth, $Conf_i$ is a confidence of implicit correct feedback item IF_i^+, $Count(GT^+)$ is the amount of player feedback items on positive ground truth, $Count(GT^-)$ is the amount of player feedback items on negative ground truth, with the rest of the variables as per above classification of feedback. Each feedback item is considered as having a numerical value of 1.

6 Evaluation

So far, we have only evaluated the effectiveness of the game. No studies have been carried out on the reputation mechanism. We have in fact investigated the following research question: **(RQ)** Whether players can spot and correct the mistakes contained in puzzle game content, thereby improve correctness of the data contained in the knowledge base. For this purpose, we measured the amount of corrections (feedback items) that participants were submitting while either solving or creating crossword puzzle games. In the following, we describe the experiment and discuss the results from the study.

6.1 Experiment Design

Experiment includes 70 participants split into two groups: crossword solvers and crossword builders. We asked crossword solvers (58) to solve 3 crosswords. We asked crossword builders (18) to create 2 crosswords. We had 16 crossword puzzles in the system available for solving. A puzzle on average contained 37 clues. Participants were also allowed to create new crosswords. There were 1 430 596 answers (words) and 2 730 719 clues available for creating crossword puzzles. The majority (around 1.3 million) of answer-clue pairs consisted of names of places (countries, villages, cities, rivers, etc). Smaller amount (approximately 120 000) of answer-clue pairs included common English words, their definitions and relations among them.

We have introduced a small percentage of mistakes into puzzle content. There were two categories of mistakes: typos in the clues or answers, and disagreements between the clue and the answer (**Table 2**). The amount of mistakes varied from 0 to 5 mistakes per puzzle. For the experiment, there were 626 answer-clue pairs in the puzzles, of them 25 contained mistakes.

Table 2. Mistake examples.

Category	Original answer-clue pair	Answer-clue pair with mistake
Typo in answer	'**e**licit', 'To induce the truth by logic reasoning'	'**r**licit', 'To induce the truth by logic reasoning'
Typo in clue	'filter', 'Device that par**t**ially removes what passes through it	'filter', 'Device that par**sh**ially removes what passes through it'
Disagreement	'ger', 'First three characters, its capital is **Berlin**'	'ger', 'First three characters, its capital is **Paris**'

6.2 Results

The experiment was scheduled to run for 7 days and run 10. During this time, participants created 8 crosswords and solved 13 crosswords with 130 game instances in total.
Out of 25 intentional mistakes, players detected a total of 9 triples as erroneous and reported them through 17 feedback items. After obtaining the results, we classified them using the given taxonomy. A summary of these observations is shown in **Table 3**. The majority of reported feedback (88%) refers to mistakes in typos, of them 13 for the answers and 2 for the clues. This can be expected since spotting disagreement may require certain level of knowledge in the domain, whereas the presence of contextual information (either in clue or answer) for the typos may ease their identification. All intentional mistake reports were by players solving puzzles. From the table we can notice that number of feedback items follows increase in introduced mistakes per puzzle. Combination or intersection of different clues in a puzzle makes it easier to perceive incorrect answers or clues by players.

Table 3. Intentional mistakes and feedback distribution for puzzle games.

Number of mistakes per puzzle	Number of puzzles	Number of reported mistakes			Feedback count		
		typo ans.	typo clue	disag.	typo ans.	typo clue	disag.
1	6	1			1		
		-	-	1	-	-	1
2	4	4			9		
		4	-	-	9	-	-
3	2	4			7		
		2	1	1	4	2	1
5	1	-			-		
Total	**13**	**9**			**17**		
		6	**1**	**2**	**13**	**2**	**2**

It is interesting that aside from intentional mistakes in triples, participants reported mistakes that had existed in the puzzle content (**Table 4**), but were not noticed before the experiment. In particular, they have noticed 12 incorrect clue-answer pairs, out of which 11 refer to typo mistakes, 3 in answers and 8 in clues.
We have been also measuring the correctness of feedback submitted by

participants. Feedback correctness was measured against ground truth facts for answer-clue pairs. In addition, we have classified incorrect feedback into false feedback and wrong format feedback. For example, feedback `yard` as a correction for the typo in clue-answer pair `afre — Unit of area, often compared to a football field (noun),` is considered to be wrong. In this case, `acre` is correct answer. However, if a player for the same pair submits feedback as `should be acre`, it is considered as being in wrong format.

Table 4 reports the correctness of feedback related to intentional and unintentional mistakes. Unintentional mistakes were already present in puzzle content. They may originate from source, open data, or they may result from the data import process. The participants have noticed 12 original mistakes versus 9 introduced. If we look at the feedback correctness, we can notice higher percent of correct feedback for unintentional mistakes. In particular, 77% correctness for unintentional mistakes and 65% correctness for intentional mistakes. In total, we had 30 feedback items where 21 (70%) was correct. Out of 9 incorrect feedback items, majority (67%) is in wrong format and a smaller amount (33%) is incorrect. However, this may be due to poor usability or overall complexity of the feedback form.

Table 4. Feedback contributed for different types of mistakes.

Mistake type	Number of reported mistakes	Total feedback count	Correct feedback	Incorrect feedback	
				false	wrong format
Intentional	9	17	11	6	
				1 (a)	5
Unintentional	12	13	10	3	
				2	1
Total	**21**	**30**	**21**	**9**	
				3	**6**

6.3 Discussion

Referring back to the research question formulated at the beginning of the evaluation section, our experiment provides evidence of the effectiveness of using gamification to improve knowledge quality. Basically, we had two types of mistakes in the knowledge base – the ones that were already present (unintentional) and the ones that we have introduced for the purpose of the experiment (intentional). Participants reported higher number of unintentional mistakes, 12 against 9 intentional. This may be explained by the fact that introduced mistakes can be biased by human knowledge, preferences, opinions and attitudes. These factors might make them more difficult to spot.

We have measured and compared the feedback produced for each of the two categories of mistakes against manually defined gold standard. Overall, the feedback correctness was 70%. This shows that for both kinds of mistakes,

gamification is a feasible solution to enhance the quality of the data contained in a large entity-centric knowledge base, such as Entitypedia.
However, the experiment has raised some more general issues. Looking from more general knowledge perspective, we can say that participants were giving feedback on combination of machine knowledge (lemmas in the knowledge base) and human knowledge (manually designed clue-value pairs from the lemmas). Automated approaches for knowledge interpretation are typically based on a simple notion of ground truth, while in reality the situation is that truth is not universal and is strongly influenced by human perspectives and the quality of the sources. The introduction of human computation (crowdsourcing or gamification) has not fundamentally changed the way golden standards are created: humans are still asked to provide a semantic interpretation of some data, with the explicit assumption that there is one correct interpretation. Thus, the diversity of interpretation and perspectives is still not taken in consideration. A simplified example from the puzzle content (**Fig. 9**) shows that ground truth is relative. The example results from polysemy.

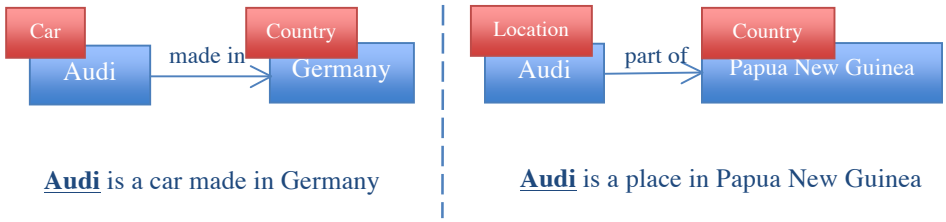

Fig. 9. Simple example of an answer that matches different clues.

We can notice that defining ground truth requires agreement. In this respect, feedback submission mechanism enables composing different viewpoints on correctness, both humans versus humans and humans versus machines. Moreover, it treats these relationships in the same way. Coupled with a reputation mechanism, it helps us to reach the agreement on what is considered to be true.

8 Related Work

Our work is situated in a larger research area concerned with human computation, namely crowdsourcing [8] and gamification [9], for linked data management and quality improvement.
At a more technical level, specific linked data management tasks have been subject to human computation. The examples include games for ontology alignment [10], for building domain ontologies from linked data [11], or interlinking of open datasets [12]. On the other hand, crowdsourcing has been used for guiding entity resolution algorithms to produce accurate results where

humans are asked questions to resolve data records [13]. Here we can also find approaches that use the crowd to create linked data [14] or to taxonomize large datasets [15]. Existing frameworks for management of the Web Linked Open Data are often limited in their ability to produce interpretable results, require user expertise or are bound to a given data set.

Regarding linked data quality improvement, researchers have been mainly analyzing the quality of Web open data. The research described in [16] proposes a methodology to discover quality issues in DBPedia. They combine MTurk and TripleCheckMate [17] in a contest form to find and verify mistakes in triples. However, crowd engagement to fix the mistakes is left to be implemented.

A general solution for gathering human annotations for different types of media is introduced with CrowdTruth, crowdsourcing annotation platform [18]. Instead of the traditional inter-annotator agreement, it implements disagreement-based metrics to evaluate the data quality issues, such as ambiguity and vagueness.

To our knowledge, we are the first that designed word games for maintaining knowledge bases. Our word games implement the complete Linked Data maintenance process, including correction of mistakes and quality control of corrections provided. They are general and flexible in a sense that they can work with Linked Data coming from different domains and represented in different formats (text and images).

9 Conclusion

In this paper we presented data certification pipeline that exploits gamification. The pipeline is implemented as a word games platform that takes content from the Entitypedia knowledge base, transforms the content into a form suitable for gameplay and brings corrections back from the crowd. We selected a subset of Entitypedia content with known (intentional) mistakes (referring to typos and disagreements) and asked players to provide corrections while solving or creating crossword puzzles.

The evaluation showed that the approach is successful; in particular, the experiment revealed that players with no expertise in knowledge management can be a useful resource to identify given quality issues in an accurate and affordable manner, by playing crossword puzzles. Moreover, the participants identified unintentional mistakes that existed in the content without prior knowledge about them.

Acknowledgements

The work described in this paper was supported by European Union's 7th Framework Programme projects ESSENCE Marie Curie Initial Training Network (GA no. no. 607062).

References

1. Shvaiko, P., Giunchiglia, F., Da Silva, P. P., McGuinness, D. L.: Web explanations for semantic heterogeneity discovery. The Semantic Web: Research and Applications – ESWC. Springer Berlin Heidelberg (2005) 303-317
2. Suchanek, F. M., Kasneci, G., Weikum, G.: Yago: A large ontology from wikipedia and wordnet. Web Semantics: Science, Services and Agents on the World Wide Web 6 (3) (2008) 203-217
3. Giunchiglia, F., Maltese, V., Farazi, F., Dutta, B.: GeoWordNet: a resource for geo-spatial applications. The Semantic Web: Research and Applications – ESWC. Springer Berlin Heidelberg (2010) 121-136
4. Maltese, V.: Enforcing a semantic schema to assess and improve the quality of knowledge resources. International Journal of Metadata, Semantics and Ontologiesto (2015) to appear
5. Deterding, S.: Gamification: designing for motivation. ACM Interactions 19.4 (2012) 14-17
6. Giunchiglia, F., Maltese, V., Biswanath, D.: Domains and context: first steps towards managing diversity in knowledge. Web Semantics: Science, Services and Agents on the World Wide Web 12 (2012) 53-63
7. Lin, Y., Michel, J. B., Aiden, E. L., Orwant, J., Brockman, W., Petrov, S.: Syntactic annotations for the google books ngram corpus. In Proceedings of the ACL 2012 system demonstrations (2012) 169-174
8. Kittur, A., Nickerson, J. V., Bernstein, M., Gerber, E., Shaw, A., Zimmerman, J., Horton, J.: The future of crowd work. In Proceedings of the conference on Computer supported cooperative work. ACM (2013) 1301-1318
9. Siorpaes, K., Hepp, M.: Games with a Purpose for the Semantic Web. IEEE Intelligent Systems 23.3 (2008) 50-60
10. Thaler, S., Siorpaes, K., Simperl. E.: Spotthelink: A game for ontology alignment. In Proceedings of the 6th Conference for Professional Knowledge Management (2011) 246-253
11. Markotschi, T., Volker, J.: Guesswhat?! - human intelligence for mining linked data. In Proceedings of the Workshop on Knowledge Injection into and Extraction from Linked Data at EKAW (2010)
12. Celino, I., Contessa, S., Corubolo, M., Dell'Aglio, D., Della Valle, E., Fumeo, S., Krüger, T.: Linking smart cities datasets with human computation–the case of urbanmatch. The Semantic Web – ISWC. Springer

Berlin Heidelberg (2012) 34-49
13. Wang, S., Lofgren, P., Garcia-Molina, H.: Question selection for crowd entity resolution. In Proceedings of the VLDB Endowment 6 (2013) 349-360
14. Abdulbaki, U.: Linked crowdsourced data-Enabling location analytics in the linking open data cloud. In Semantic Computing (ICSC), 2015 IEEE International Conference on. IEEE (2015) 40-48
15. Bragg, J., Weld. D.: Crowdsourcing Multi-Label Classification for Taxonomy Creation. In Proceedings of the First AAAI Conference on Human Computation and Crowdsourcing. (2013) 25-33
16. Acosta, M., Zaveri, A., Simperl, E., Kontokostas, D., Auer, S., Lehmann, J.: Crowdsourcing linked data quality assessment. The Semantic Web – ISWC. Springer Berlin Heidelberg (2013) 260-276
17. Kontokostas, D., Zaveri, A., Auer, S., Lehmann, J.: Triplecheckmate: A tool for crowdsourcing the quality assessment of linked data. Knowledge Engineering and the Semantic Web. Springer Berlin Heidelberg (2013) 265-272
18. Oana, I., Khamkham, K., Cristea, T., Dumitrache, A., Rutjes, A., van der Ploeg, J., LuRomaszko, L., Aroyo, L., Jan Sips, R.: CrowdTruth: Machine-Human Computation Framework for Harnessing Disagreement in Gathering Annotated Data. The Semantic Web – ISWC. Springer International Publishing (2014) 486-504