

ASSESS — Automatic Self-Assessment Using Linked Data

Lorenz Bühmann, Ricardo Usbeck^(✉), and Axel-Cyrille Ngonga Ngomo

IFI/AKSW, Universität Leipzig, Leipzig, Germany
`{buehmann,usbeck,ngonga}@informatik.uni-leipzig.de`

Abstract. The Linked Open Data Cloud is a goldmine for creating open and low-cost educational applications: First, it contains open knowledge of encyclopedic nature on a large number of real-world entities. Moreover, the data being structured ensures that the data is both human- and machine-readable. Finally, the openness of the data and the use of RDF as standard format facilitate the development of applications that can be ported across different domains with ease. However, RDF is still unknown to most members of the target audience of educational applications. Thus, Linked Data has commonly been used for the description or annotation of educational data. Yet, Linked Data has (to the best of our knowledge) never been used as direct source of educational material. With ASSESS, we demonstrate that Linked Data can be used as a source for the automatic generation of educational material. By using innovative RDF verbalization and entity summarization technology, we bridge between natural language and RDF. We then use RDF data directly to generate quizzes which encompass questions of different types on user-defined domains of interest. By these means, we enable learners to generate self-assessment tests on domains of interest. Our evaluation shows that ASSESS generates high-quality English questions. Moreover, our usability evaluation suggests that our interface can be used intuitively. Finally, our test on DBpedia shows that our approach can be deployed on very large knowledge bases.

1 Introduction

The amount of RDF data available across the globe has grown significantly over the last years. As pointed out in previous works, a large portion of the open data available in this format is encyclopedic in nature [9]. While RDF data is being used for the description and annotation of education material and websites, this data format has (to the best of our knowledge) never been used as source of educational material. This is simply due to the target audience of educational material not being familiar with this technology. However, given (1) the large number of domains already described as RDF, (2) the large number of websites annotated with RDFa and (3) the simple structure of RDF statements, RDF

L. Bühmann and R. Usbeck—Both authors contributed equally to this work.

knowledge bases seem to be an optimal source or the automatic generation of educational material.

With ASSESS, we address the automatic generation of education material directly from RDF data. With this contribution, we aim to show that the Web of Data has the potential to contribute to the dissemination of educational materials across borders, especially to the less privileged, an endeavor in line with platforms such as Coursera¹ and EdX.² ASSESS' main contribution is the automatic generation of self-assessment tests containing a variety of user-selected question types directly from RDF. The intended users are consequently (1) persons who aim to assess their knowledge on a particular domain of interest (we call these persons *learners*) and (2) persons in charge of assessing the knowledge of learners (called *teachers* in the following). Manifold usage scenarios can be envisaged for the approach, including the preparation of tests on a particular domain, the training of employees on novel products, the generation of tests for exams, casual gaming, the extension of ones general knowledge and many more.

ASSESS achieves its goal by providing innovative solutions to the following:

1. *Automatic verbalization of RDF graphs*: While RDF benefits the applications by making the generation of questions easy, most users do not understand RDF. We are able to hide the RDF data completely from the end user while making direct use of this data to generate questions of different types. Our verbalization approach is generic and thus independent of the underlying knowledge base. In addition, it is time-efficient as demonstrated by our evaluation on selected knowledge bases.
2. *Entity summarization*: Our entity summarization approach allows detecting key properties for describing a resource. This allows for the generation of succinct (natural-language) descriptions of resources that can then be transformed into questions of different difficulty.
3. *RDF fragment extraction*: We use Concise Bound Descriptions (CBDs) of RDF resources to generate fragments of the input knowledge that contain resources which abide by the domain description provided by the user. These fragments are used both for the generation of questions and of hints towards the answer of questions.

By using these techniques our tool enables users to generate customized quizzes pertaining to a user-defined domain. Currently, our tool supports the generation of Yes/No, Jeopardy-style and Multiple-Choice questions. Other types of questions are being added. Our system is now a working system deployed at <http://assess.aksw.org/demo/>. The system is open-source and abides by the GPL license. The server code is part of the SemWeb2NL project and can thus be found at <http://github.com/AKSW/SemWeb2NL>. The client code can be found at <https://github.com/AKSW/ASSESS>. Throughout this paper, we assume that the learner is a medical student aiming to test his/her knowledge about the human circulatory system from DBpedia.

¹ <https://www.coursera.org/>

² <https://www.edx.org/>

2 System Description

ASSESS is an application for the generation of questions and answers from RDF. It abides by the client/server paradigm, where the client is a user interface that displays the information returned by the server. The communication between client and server is ensured by REST interfaces which consume and generate JSON messages. The server side of ASSESS consists of four layers.

2.1 Data Layer

The *data layer* ensures the communication with the knowledge base(s) and implements the extraction of CBDs for resources. The data layer can deal with one or several knowledge bases. As input, the layer expects a set of classes or an OWL class expression E . When provided with this description of the domain of interest for a user, the data layer selects the fragment of the underlying knowledge base which pertains to the users domain description. To this end, it begins by selecting all the resources that abide by the description E . For example, if E is simply the class `:Vein`, we begin by retrieving all instances of the class, including for example the *inferior vena cava* (IVC). The data layer then retrieves the CBDs of each of these resources via SPARQL queries. Depending on the required difficulty of the questions (which the user can set), the layer then performs further CBD retrieval for all the resources in the graph for which we do not yet have a CBD. For example, the CBD of the *abdominal aorta*, the artery of which the IVC is the venous counterpart, would be retrieved if a second round of CBD retrievals were required. By these means, ASSESS supports the generation of deeper queries over several hops.

For the sake of scalability, the data layer also implements several in-memory and hard drive-driven caching solutions. By these means, we ensure that our approach scales to large knowledge bases such as DBpedia (see Section 4 for our evaluation results). The output of the data layer is a graph which contains a set of CBD out of which the questions are generated. Note that we do not target data quality with ASSESS (see, e.g., [15] for approaches that do this). Instead, we assume that the underlying knowledge base is a curated knowledge base. We argue that such knowledge bases will be increasingly available in the near future (e.g., through projects such as SlideWiki³ or LinkedUp⁴). Still, we aim to add mechanisms for processing user feedback (e.g., flags for triples that lead to incorrect questions) in the near future.

2.2 Natural-Language Generation Layer

This layer provides the mechanisms necessary to verbalize RDF triples, SPARQL basic graph patterns (BGPs) as well as whole SPARQL queries. This layer is an extension of the SPARQL2NL framework [13], which has been shown to

³ <http://slidewiki.org/>

⁴ <http://linkedup-challenge.org/>

achieve high-quality verbalizations for the Question Answering on Linked Data (QALD) benchmark. This makes the framework particularly well suited for the generation of natural-language questions out of RDF. This layer contains two main components: a verbalizer and a realizer. Given a set of triples or BGPs, the verbalizer can generate a sentence parse tree by employing syntactic and semantic rules. This sentence parse tree is then the input for the realizer, which generates natural language.

The realization of a triple pattern or of a triple $s \ p \ o$ depends mostly on the verbalization of the predicate p . If p can be realized as a noun phrase, then a possessive clause can be used to express the semantics of $s \ p \ o$, as shown in 1. For example, if p is a relational noun like `author`, then the verbalization is $?x$'s `author` is $?y$. In case p 's realization is a verb, then the triple can be verbalized as given in 2. For example, if p is the verb `write`, then the verbalization is $?x$ writes $?y$.

1. $\rho(s \ p \ o) \Rightarrow \text{poss}(\rho(p), \rho(s)) \wedge \text{subj}(\text{BE}, \rho(p)) \wedge \text{dobj}(\text{BE}, \rho(o))$
2. $\rho(s \ p \ o) \Rightarrow \text{subj}(\rho(p), \rho(s)) \wedge \text{dobj}(\rho(p), \rho(o))$

The combination of verbalizations of single triple patterns is also carried out using rules. For example, the object grouping rule (also known as *backward conjunction reduction*) collapses the subjects of two sentences $\rho(s_1 p_1 o_1)$ and $\rho(s_2 p_2 o_2)$ if the realizations of the verbs and objects of the sentences are the same:

$$\begin{aligned} \rho(o_1) = \rho(o_2) \wedge \rho(v_1) = \rho(v_2) \wedge \text{cc}(v_1, \text{coord}) \\ \Rightarrow \text{root}(Y, \text{PLURAL}(v_1)) \wedge \text{subj}(v_1, \text{coord}(s_1, s_2)) \wedge \text{dobj}(v_1, o_1), \end{aligned}$$

where the $\text{coord} \in \{\text{and}, \text{or}\}$ is the coordination combining the input sentences, and $\text{coord} \in \{\text{conj}, \text{disj}\}$ is the corresponding coordination combining the subjects. Other rules can be found in the SemWeb2NL code as well as in [13].

2.3 Entity Summarization

This layer provides mechanisms for the detection of the most relevant predicates when aiming to describe a given resource s . To this end, we begin by finding the most specific class to which this resource belongs, i.e., the class C that is such that s is an instance of C but of none of its subclasses. Now, in addition to labeling predicates such as `rdfs:label`, we find the properties that are most frequently used in combination with instances of C . The properties are sorted in descending order of usage frequency and the top k properties are retrieved (where $k = 5$ is our default setting). The approach relies on property frequencies in combination with labeling properties [6] to detect the most important properties for a resource. Note that by choosing only k properties, we can deal with large CBDs. Moreover, the system discards resources whose description is too short. We then retrieve (as far as they exist) the objects of these k predicates w.r.t. s . Note that we select at most three of the objects of each of the k properties selected by our approach. The resulting set of triples is the summary for s . The user can obviously choose the properties of interest for his/her learning/teaching

goal. For example, only the blood flow through veins might be of importance for our user, in which case he would choose the `dbo:flow` property as target property for learning. The generated summary is forwarded to the next layer.

2.4 Question and Answer Generation Layer

The last layer contains modules which implement a generic interface for question and answer generation. Each of these modules takes a set of resources as input and generates (1) a question, (2) a set of correct answers and optionally (3) a set of wrong answers. Note that while we currently support three types of questions, a multitude of other question types can be envisaged. The types of questions we implement at the moment are:

- *Yes/No questions*: These are statements to which the user has to state whether they are true or false (see Figure 1 for an example). Giving an entity summary, we begin by selecting a triple $(s \ p \ o)$. Then, we randomly decide on whether the question to be generated should have the answer true or false. If true is selected, then we generate “Is the following statement correct:”, followed by $\rho(s \ p \ o)$. Else, we begin by generating a wrong assertion by replacing the object of the triple with an object o' such that $(s \ p \ o')$ does not belong to the input knowledge base and there exists a s' for which $(s' \ p \ o')$ holds in the input knowledge base. Note that while we actually assume a closed world here, this approach works well in real use cases.

Create new set of questions

Total Score

Question 9/10
4 correct answers
00 : 07 : 59

Is the following statement correct: Inferior vena cava's artery is Abdominal aorta.

True

False

Submit

Fig. 1. Yes/No question

- *Jeopardy questions*: Jeopardy questions describe an entity without naming the entity itself. To generate such questions, we begin with a summary of the resource to describe. We then replaced $\rho(s)$ by expressions such as **This $\rho(C)$** (where C is a class to which s belongs) or pronouns (it, he, she) depending on the type of resource at hand. The result is a summary of the entity that

does not contain its label, e.g., This anatomical structure's dorlands prefix is v 05 and its dorlands suffix is 12851372. The correct answer(s) are then the resource(s) which share with s all predicate-object pairs used to generate the description of s (in this case the hepatic portal vein). Negative answers are generated by looking for resources that share a large number of property-values pairs with positive answers but are not elements of the set of correct answers (e.g., the splenic vein).

The screenshot shows a mobile application interface. At the top right, there is a blue header bar with the text "Question 7/10", "0 correct answers", and a timer "00 : 01 : 05". Below this, the main content area has a white background. On the left, there is a vertical list of five options, each enclosed in a light gray rectangular box. The first option, "Hepatic portal vein.", is highlighted with a green background. The other four options are: "Left gastroepiploic vein.", "Superior rectal vein.", "Pancreaticoduodenal veins.", and "Splenic vein.". At the bottom right of the content area, there is a blue button labeled "Next".

Fig. 2. Jeopardy question in ASSESS' client

- *Multiple-choice question:* This type of questions is commonly used in automatic assessment tests. Here, we generate questions by verbalizing a pair (s, p) , e.g., jugular vein's artery. Then, we retrieve all o such that (s, p, o) holds. These are the correct answers to the question. In addition, we find o' such that there (s, p, o') is not in the knowledge base but there is at least one other resource s' with (s', p, o) . The user is then to choose a subset of the provided answers (see Figure 3).

Each of the layers provides a REST interface with which it can communicate with other applications. Thus, the server side of ASSESS can be integrated in any educational application that requires the verbalization of the RDF triples, BGPs or SPARQL queries.

The current client side of the application was designed with portability in mind and can be used on both stationary and mobile devices. The main motivation behind this design choice was that in less privileged countries, mobile

Question 2/10
0 correct answers
00 : 09 : 19

Please select Internal jugular vein's artery.

Common carotid artery.

Internal thoracic artery.

Axillary artery.

Tubal branches of ovarian artery.

Lesser palatine arteries.

Next

Fig. 3. Multiple-Choice question in ASSESS' client

devices are the instrument of choice to access the Web [2]. Thus, our provision of a ubiquitous-device-friendly interface has the potential to support the utilization of our application across the world. We demonstrate the usability of ASSESS by deploying on DBpedia as it is a large knowledge base with a complex ontology.

3 Distinguishing Features

Overall, the most distinguishing feature of ASSESS is that it implements a bridge between RDF, SPARQL and natural language (in this case English). Therewith, it makes RDF amenable to be a source of content (and not only of metadata or descriptions) for educational applications. Still, the education material generated by content can be consumed by users with all possible range of expertise in Semantic Web technologies. In the following, we present how ASSESS addresses the challenge of providing useful educational content using Linked Data:

- **Innovation in Education:** ASSESS addresses the automatic generation of tests out of structured data, especially RDF. Its main innovation w.r.t. to education software pertains to the generation of the test questions directly out of RDF data. So far, Linked Data has been most commonly used to describe educational data. To the best of our knowledge, ASSESS is the first approach that uses state-of-the-art technology to extract and verbalize questions directly out of RDF. The resulting tests are available in the form of different types of quizzes, allowing users to prepare for specific types of tests. The learning process is improved in several ways. First, the users can tailor the tests to exactly the domain in which they are interested by describing

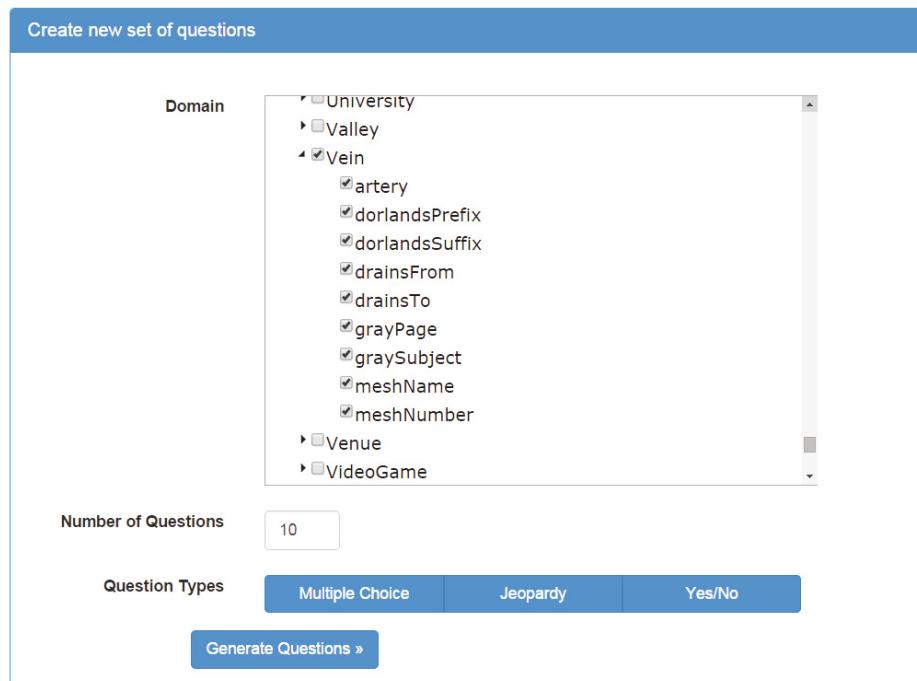


Fig. 4. Configuration window of ASSESS

both the type of resources as well as the properties of these resources for which questions are to be generated. Moreover, the users can specify the type of questions they wish to answer as well as the number of questions. Therewith, users can easily tailor the assessment to exactly their needs. The online availability of the tests via a simple browser furthers ubiquitous learning, making the learning process more efficient. The interface supports a design-for-all paradigm by remaining consistent across devices and by being simple and yet powerful.

- **Audience:** Our framework is domain-independent and addresses all users who are interested in assessing their own knowledge of a particular domain or the knowledge of others. Consequently, it is suitable for learners and teachers of all age as long as they are able to read written natural language. Note that while the current natural-language generation layer only supports English as natural language, the verbalizer is currently being extended to support French. Learners can use ASSESS for the sake of self-assessment by generating tests that pertain to the areas in which they need to evaluate their knowledge. Teachers can use our tool for several purposes, including the following two: First, they can generate tests and make these available to students for the sake of assessing them. Moreover, they can make the learning material available as RDF and support the learners during the preparation

for exams by deploying ASSESS on top of this learning material. For courses with respect to general knowledge (such as geography, history, etc.), the teachers can easily reuse existing knowledge bases such as DBpedia and simply define the fragment of the knowledge base that contains the resources of interest for their course. Manifold other applications can be built on top of ASSESS. For example, we are currently exploring the use of our framework for schooling employees that need to learn the specifics of novel products that are being marketed by their company. We have evaluated the technology underlying ASSESS with 120+ users. The results of this evaluation are presented in [13]. We are currently planning the integration of ASSESS and SlideWiki [11]⁵ for the generation of questions out of the content of SlideWiki slides. Currently, SlideWiki has 3800+ presentations and more than 27.000 slides. The number of active users is around 1381.

- **Usability:** While designing the client-side of ASSESS, we wanted our end user to be confronted with a simple interface in a well-known look-and-feel and simple instructions. Moreover, we wanted the interface to be easily usable on both mobile and stationary devices. We thus chose to use mobile-optimised Java Script libraries to implement the interface of ASSESS. When using our tool, the user is first confronted with a simple configuration window that allows him/her to setup the test (see Figure 4) Here, descriptions are provided for each of the interface elements, making the configuration windows easy to use. The user is subsequently confronted with a single window per question. The verbalization of each question is displayed on the top of the page while the possible answers are displayed underneath. This approach to displaying quizzes is akin to the way questions are shown in exams. Moreover, it has been used in manifold previous applications and is thus well known and easy to use. Finally, we rely on the widely adopted Twitter Bootstrap for the interface design. These libraries have already been used to create manifold applications and thus has a look-and-feel that is familiar to most internet-affine users. The color coding is the street light coding (green for correct, red for false) and is thus easy to interpret. The summary window expresses statistics in the manner of a dashboard and can thus be easily read. The users are also given the option to export their results for future reference.
- **Performance:** Our approach is based on selecting a set of resources and using CBDs to generate questions about these resources. Hence, the time required to generate an assessment grows linearly with the number of questions. We deployed ASSESS on the large dataset DBpedia to show its scalability. Overall, we need around 5 seconds to initialize the class hierarchy and around 4 seconds/question to generate an assessment. Given that we rely on the SPARQL endpoint which contains the dataset upon which questions are to be generated and by virtue of the caching mechanisms employed in the tool, we scale as well as modern triple stores and can thus easily deploy ASSESS on billions of triples while still generating tests in acceptable times.

⁵ <http://slidewiki.org>, Statistics were collected on April 30th, 2015.

- **Data Usage and Quality:** Our application can be deployed on any set of RDF datasets. We rely directly on the ontology of the dataset to generate quizzes by using state-of-the-art verbalization techniques. Given that our tool is independent of the dataset used, there is no need for a documentation or version control of the underlying datasets. While ASSESS does not store the results of the users, it can be easily extended to do so (for example for the sake of statistical analysis by teachers). Further possible extensions would be user groups for the teachers to assess how well a group of students perform on a particular topic and thus supporting them while deciding on the content and didactics of future teaching plans.
- **Legal and Privacy:** We do not store any user data. Yet, the tool can be easily extended to do so. The terms of use state explicitly that the current version of the tool is free to use, that the tool is provided as-is and that no guarantees are provided. Still, our system does not rely on local client-side data and cannot harm our users' systems in any way. Given that we do not replicate the data contained in the endpoint (we verbalize it), we are immune against any license that prohibits the duplication of portions of the underlying knowledge base. The person who deploys ASSESS on knowledge bases with more restrictive licenses is required to restrict the accessibility of the tool to certain users (for example by using a server with a restricted access). The data generated by the ASSESS is free to use for any purpose.

4 Evaluation

In the following, we begin by presenting an evaluation of the most critical component of ASSESS, i.e., its verbalizer, as well as an usability study of the web-interface. Thereafter, we contrast our approach with related work. Finally, we present current limitations and possible extensions of our tool.

4.1 Evaluation of the Verbalizer

We have evaluated the technologies underlying ASSESS with 125 users (see Figure 5). Here, we were especially interested in knowing how well our verbalization framework performs. To this end, we used 200 queries of different complexity from the QALD-3 benchmark dataset⁶. Each user was confronted with 10 queries. We used the scale proposed by Doddington [5] to measure the fluency and adequacy of our approach. Overall, we achieved a fluency of 4.56 ± 1.29 (mean \pm standard deviation, see Figure 5), meaning that the language we generate can be understood by the users, even if it sometimes contains grammatical mistakes. Our adequacy results were 5.31 ± 1.08 (see Figure 5), which is a very positive result. This means that the expressions we use when verbalizing the classes and properties match the expected verbalization well. In 62% of the cases, we even achieve a perfect score. Further evaluation details can be found in [13].

⁶ <http://greententacle.techfak.uni-bielefeld.de/~cunger/qald/>

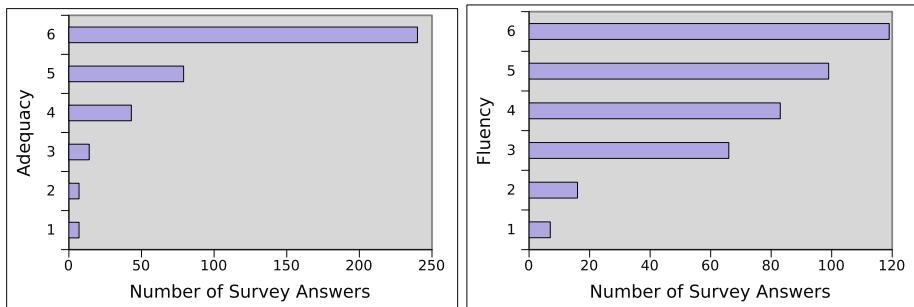


Fig. 5. Adequacy (left) and fluency (right) of ASSESS' verbalizer

4.2 Evaluation of the User Interface

We also performed a system usability study (SUS)⁷ to validate the design of our web interface. 7 users - with a good or no knowledge of natural language processing, language generation or e-learning - answered our survey resulting in a SUS-Score of 85.3. This score assign the mark *S* to the current interface of ASSESS and places it into the 10% best category of interface, meaning that users of the interface are likely to recommend it to a friend. Figure 6 shows the average voting per question and its standard deviation. All users found that they did not need to learn anything to use the tool and that the interface was intuitive. Moreover, none of the users thought that he/she would need the support of a technical person to be able to use this system (Q4) nor need to learn a lot of things before they could get going with this system (Q10). These results suggest that our interface can be deployed on a large number of learning scenarios.

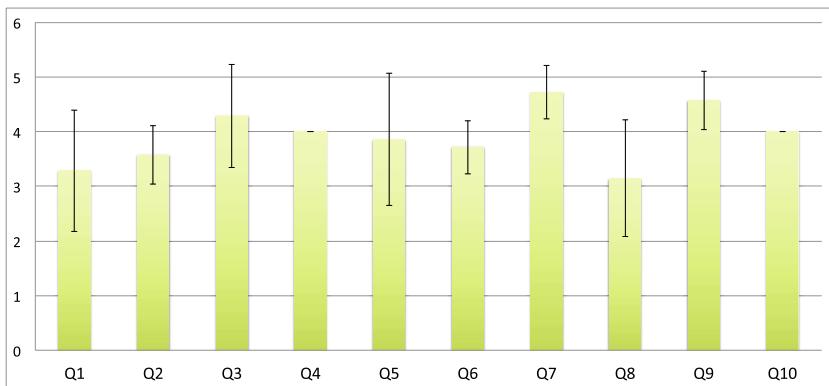


Fig. 6. Average SUS voting per question with standard deviation.

⁷ <http://www.measuringu.com/sus.php>

5 Related Work

The LinkedUp project⁸ (which ran the LinkedUp challenges) can be regarded as a crystallization point for innovative tools and applications for Open Web Data for educational purposes. Within its three consecutive challenges more than 40 submissions proofed the value of Linked Data to the educational community.

For example, Didactalia [10] is the worlds largest educational knowledge base with more than 85.000 resources created and used by more than 250.000 people. Its underlying 9 ontologies allow sophisticated information retrieval. However, the provided resources and questionnaires are static. LodStories [4] leverages Linked Open Data to build multimedia stories for art based on person, location and art data. Therewith, this project enables learning processes in the domain of art. MELOD [1] puts its emphasis on the importance of Linked Data in mobile contexts by presenting an mobile application for students visiting another city. To support this endeavor, MELOD provides ad-hoc information based on the user's location acquired from DBpedia, Europeana and Geonames.

To the best of our knowledge, the automatic generation of assessments has been addressed by a very small number of approaches. AutoTool⁹ can generate and evaluate tests that pertain to the domain of theoretical computer science. It uses a generate-and-test approach to test students answers to questions pertaining to formal grammars. Aplusix¹⁰ generates tests for the particular domain of algebra. GEPPET O[12] specializes on generating pen-and-paper tests with the aim of adapting to learners work sequences. The tool that is closest to ours in spirit is the CLAIRE framework [3], which implements a semi-automatic approach for the generation of self-assessment exercises. Yet, CLAIRE uses its own format to represent knowledge, making it more difficult to port than ASSESS. Furthermore, Foulonneau [7] has shown the value of DBpedia and the LOD Cloud to generate educational assessment items. To the best of our knowledge, existing approaches do not generate natural language questions directly out of RDF triples. This is one of the innovations of ASSESS.

6 Current Limitations and Possible Extensions

While our tool can already be used for educational purposes, it still has several limitations. First, ASSESS only generates questions in English. This limitation can be easily dealt with by extending the verbalizer with the grammar of other languages. The SimpleNLG framework¹¹ (on which we rely) has already been extended to generate French and German. These extensions will be added in future work. A further limitation of our tool is that it does not yet used OWL semantics (for example `owl:sameAs` links) to generate questions. This can yet be remedied easily during the CBD generation by (1) checking for `owl:sameAs`

⁸ <http://linkedup-challenge.org/>

⁹ <http://www.imn.htwk-leipzig.de/~waldmann/autotool/>

¹⁰ <http://www.aplusix.com/de/>

¹¹ <https://code.google.com/p/simplenlg/>

links that either go from a resource or point to a resource and (2) merging the CBD of the linked resources with that of the original resource. Our tool can be extended in several other ways. Most importantly, new question generation modules can be added. ASSESS provides a generic interface for question types. Thus, experts in need of other question types can simply implement the interface to their ends. New types of questions can include the following (note that novel user interfaces might be required for these questions):

- Relational questions: What’s the relation between two resources?
- Graphical games: Given a graph of resources and colored edges, color the missing edges correctly (each colour represents a certain relation). The game can be made more difficult by mixing missing resources and edges.
- Story-telling games: The idea here would be to verbalize a portion of the graph and replace a subset of the resources with variables. Then, the task would be to assign a resource to each of the variables in the story.

Especially in less rich countries, ASSESS presents an opportunity to distribute knowledge at low cost as it can support remote learning and online programs for students who cannot afford attending expensive universities. We aim to push towards such a use of ASSESS by combining with free lecture slides (such as SlideWiki for example) and providing means for generating ASSESS tests out of teaching material in natural language.

7 Conclusion

We presented ASSESS, a framework for the automatic generation of questions out of RDF data and SPARQL queries and query fragments. Our framework reuses and extends state-of-the-art verbalization frameworks such as SPARQL2NL. Moreover, it implements fully novel approaches for entity summarization and natural-language generation from RDF triples. We showed that ASSESS can be used on real data by evaluating (1) its kernel with 125 users on DBpedia and DBpedia queries and (2) its usability using the SUS scale. ASSESS is part of a larger agenda, in which we aim to use existing research to make educational material easily and freely available to learners all around the planet. In future work, we thus aim to combine ASSESS with teaching materials in natural language to provide both lecture slides and self-evaluation questions at low cost. Several problems remain to be solved to achieve this goal. Especially, we aim to deploy a generic approach to extract RDF triples from the pseudo-natural language used in slides leverages existing technologies like FOX [14] and BOA [8].

References

1. Arrigo, M., Taibi, D., Fulantelli, G.: A mobile environment for learning with linked open data (2013)
2. Bornman, E.: The mobile phone in africa: Has it become a highway to the information society or not. Contemp. Edu. Tech. **3**(4) (2012)

3. Cablé, B., Guin, N., Lefevre, M.: An authoring tool for semi-automatic generation of self-assessment exercises. In: Lane, H.C., Yacef, K., Mostow, J., Pavlik, P. (eds.) AIED 2013. LNCS, vol. 7926, pp. 679–682. Springer, Heidelberg (2013)
4. Chen, J., Liu, Y., Maulik, D., Xu, L., Zhang, H., Knoblock, C.A., Szekely, P., Vander Sande, M.: Lodstories: Learning about art by building multimedia stories1 (2014)
5. Doddington, G.: Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In: Proceedings of the Second International Conference on Human Language Technology Research, pp. 138–145. Morgan Kaufmann Publishers Inc. (2002)
6. Ell, B., Vrandečić, D., Simperl, E.: SPARTIQUULATION: verbalizing SPARQL Queries. In: Simperl, E., Norton, B., Mladenic, D., Valle, E.D., Fundulaki, I., Passant, A., Troncy, A. (eds.) ESWC 2012. LNCS, vol. 7540, pp. 117–131. Springer, Heidelberg (2015)
7. Foulonneau, M.: Generating educational assessment items from linked open data: the case of dbpedia. In: García-Castro, R., Fensel, D., Antoniou, G. (eds.) ESWC 2011. LNCS, vol. 7117, pp. 16–27. Springer, Heidelberg (2012)
8. Gerber, D., Ngonga Ngomo, A.-C.: Bootstrapping the linked data web. In: 1st Workshop on Web Scale Knowledge Extraction @ ISWC 2011 (2011)
9. Gerber, D., Hellmann, S., Bühlmann, L., Soru, T., Usbeck, R., Ngonga Ngomo, A.-C.: Real-time RDF extraction from unstructured data streams. In: Alani, H., Kagal, L., Fokoue, A., Groth, P., Biemann, C., Parreira, J.X., Aroyo, L., Noy, N., Welty, C., Janowicz, K. (eds.) ISWC 2013, Part I. LNCS, vol. 8218, pp. 135–150. Springer, Heidelberg (2013)
10. Gómez, A.A.: Analyzing and producing educational resources for didactalia. net: a pilot project launched at the university of deusto (spain) with students from primary education degree. In: INTED2014 Proceedings, pp. 7186–7191 (2014)
11. Khalili, A., Auer, S., Tarasowa, D., Ermilov, I.: SlideWiki: elicitation and sharing of corporate knowledge using presentations. In: ten Teije, A., Völker, J., Handschuh, S., Stuckenschmidt, H., d'Acquin, M., Nikolov, A., Aussenac-Gilles, N., Hernandez, N. (eds.) EKAW 2012. LNCS, vol. 7603, pp. 302–316. Springer, Heidelberg (2012)
12. Lefevre, M., Jean-Daubias, S., Guin, N.: Generation of pencil and paper exercises to personalize learners work sequences: typology of exercises and meta-architecture for generators. In: World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education, vol. 2009, pp. 2843–2848 (2009)
13. Ngonga Ngomo, A.-C., Bühlmann, L., Unger, C., Lehmann, J., Gerber, D.: Sparql2nl - verbalizing sparql queries. In: Proc. of WWW 2013 Demos, pp. 329–332 (2013)
14. Speck, R., Ngonga Ngomo, A.-C.: Ensemble learning for named entity recognition. In: Mika, P., Tudorache, T., Bernstein, A., Welty, C., Knoblock, C., Vrandečić, D., Groth, P., Noy, N., Janowicz, K., Goble, C. (eds.) ISWC 2014, Part I. LNCS, vol. 8796, pp. 519–534. Springer, Heidelberg (2014)
15. Zaveri, A., Rula, A., Maurino, A., Pietrobon, R., Lehmann, J., Auer, S.: Quality assessment for linked data. A survey. Semantic Web Journal (2015)