Comparing Research Contributions in a Scholarly Knowledge Graph

Allard Oelen
L3S Research Center, Leibniz
University of Hannover
oelen@l3s.de

Mohamad Yaser Jaradeh
L3S Research Center, Leibniz
University of Hannover
jaradeh@l3s.de

Kheir Eddine Farfar
TIB Leibniz Information Centre for
Science and Technology
kheir.farfar@tib.eu

Markus Stocker
TIB Leibniz Information Centre for
Science and Technology
markus.stocker@tib.eu

Sören Auer
TIB Leibniz Information Centre for
Science and Technology
auer@tib.eu

ABSTRACT

Conducting a scientific literature review is a time consuming activity. This holds for both finding and comparing the related literature. In this paper, we present a workflow and system designed to, among other things, compare research contributions in a scientific knowledge graph. In order to compare contributions, multiple tasks are performed, including finding similar contributions, mapping properties and visualizing the comparison. The presented workflow is implemented in the Open Research Knowledge Graph (ORKG) which enables researchers to find and compare related literature. A preliminary evaluation has been conducted with researchers. Results show that researchers are satisfied with the usability of the user interface, but more importantly, they acknowledge the need and usefulness of contribution comparisons.

KEYWORDS

Scholarly Communication; Scholarly Information Systems; Scholarly Knowledge Comparison; Comparison User Interface; Digital Libraries

1 INTRODUCTION

When conducting scientific research, finding and comparing state-of-the-art literature is an important activity. Mainly due to the unstructured way of publishing scholarly knowledge it is currently time consuming to find and compare related literature. The Open Research Knowledge Graph[1] (ORKG) [9] is a system designed to acquire, publish and process structured scholarly knowledge published in the scholarly literature. One of the main features of the ORKG is the ability to automatically compare related literature.

The benefits of having scholarly knowledge structured include, among others, the ability to easily find, retrieve but also compare such knowledge. Comparing resources (scholarly knowledge or other) can be useful in many contexts [15], for instance resources describing cities for their population, area and other attributes. Comparing structured data is useful particularly when compared resources are described with similar or even same properties. Although knowledge graphs are of course structured, resources—even

of the same type—are often described using different and differently named attributes. Moreover, different hierarchical structures can complicate a comparison of two resources.

In this paper, we present a workflow that describes how to select and compare resources describing scholarly knowledge in graph databases. We implement this workflow in the ORKG, which enables the comparison of related literature, including state-of-the-art overviews. In the ORKG, these resources are specifically called *research contributions*. A research contribution relates the *research problem* addressed by the contribution, the *research method* and (at least one) *research result*. Currently, we do not further constrain the description of these resources. Users can adopt arbitrary third-party vocabularies to describe problems, methods, and results. We thus tackle the following research questions:

- RQ1: *How to compare research contributions in a graph based system?*
- RQ2: *How to effectively specify and visualize research contribution comparisons in a user interface?*

2 RELATED WORK

Resource (or entity) comparison is a well-known task in a variety of information systems, for instance in e-commerce or hotel booking systems. In e-commerce, products can be compared in order to help customers during the decision process [19]. These comparisons are often based on a predefined set of properties to compare (e.g., price and color). This does not apply when comparing resources in a community-created knowledge graph where there is no predefined set of properties. Petrova et al. created a framework to compare heterogeneous entities in RDF graphs using SPARQL queries [15]. In this framework, both the similarities and differences between entities are determined.

The task of comparing research contributions in a graph system can be decomposed into multiple sub tasks. The first sub task is finding suitable contributions to compare. The most suitable comparison candidates are similar resources. The actual retrieval of similar resources can be seen as an information retrieval problem, with techniques such as TF-IDF [12]. Measures to calculate the structural similarity between RDF graphs have been proposed in the literature (e.g., Maillot et al. [11]). A second sub task is matching semantically similar predicates. Determining the similarity of resources is a recurring problem in dataset interlinking [1] or the

---

[1]http://orkg.org

more general task of ontology alignment/matching [17]. For property mapping, techniques of interest include edit distance (e.g., Jaro-Winkler [18] or Levenshtein [10]) and vector distance. Gromann and Declerck evaluated the performance of word vectors for ontology alignment and found that FastText [2] performed best [7].

As suggested above, an effective automated related literature comparison relies on scholarly knowledge being structured. There is substantial related work on representing scholarly knowledge in structured form. Building on the work of numerous philosophers of science, Hars [8] proposed a comprehensive scientific knowledge model that includes concepts such as theory, methodology and statement. More recently, ontologies were engineered to describe different aspects of the scholarly communication process. Among them are CiTO[2] and C4O[3] for recording citation related concepts, FaBiO[4] and BiRO[5] for capturing bibliographic data and PRO[6], PSO[7] and PWO[8] for the publication process. Additionally DoCO[9] can be used to describe the structure of a document which can be complemented by DEO[10] to also include rhetorical elements to describe the scientific discourse [4]. Among others, these ontologies are part of the Semantic Publishing and Referencing Ontologies (SPAR), a collection of ontologies that can be used to describe scholarly publishing and referencing of documents [6, 13, 14]. Ruiz Iniesta and Corcho [16] reviewed the state-of-the-art ontologies to describe scholarly articles.

These ontologies are designed to capture primarily metadata about and structure of scholarly articles, not the actual research contributions (scholarly knowledge) communicated in articles. In order to conduct a useful comparison, only comparing article metadata and structure is oftentimes not sufficient. Rather, a comparison should include (structured descriptions of) problem, materials, methods, results and perhaps other aspects of scholarly work. The Open Research Knowledge Graph (ORKG) [9] supports creating such structured descriptions of scholarly knowledge. We thus embed the research contribution comparison presented in this paper in the larger effort of the ORKG project, which aims to advance scholarly communication infrastructure.

3 WORKFLOW

We present a workflow that describes how to perform a comparison of research contributions, thereafter more generally referred to as resources. This workflow consists of four different steps: 1) select comparison candidates, 2) select related statements, 3) map properties and 4) visualize comparison. The workflow is depicted in Figure 1. Section 4 presents the implementation for the individual steps. We now discuss each step of the workflow in more detail.

3.1 Select comparison candidates

To perform a comparison, a starting resource is needed. This resource is called *main resource* and is always manually selected

---

[2]http://purl.org/spar/cito
[3]http://purl.org/spar/c4o
[4]http://purl.org/spar/fabio
[5]http://purl.org/spar/biro
[6]http://purl.org/spar/pro
[7]http://purl.org/spar/pso
[8]http://purl.org/spar/pwo
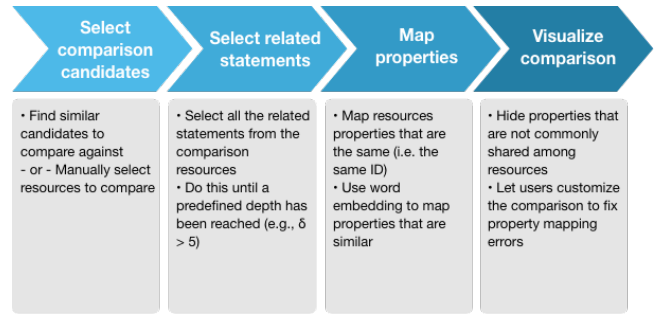[9]http://purl.org/spar/doco
[10]http://purl.org/spar/deo

Figure 1: Resource comparison workflow.

by a user. The main resource is compared against other *comparison resources*. There are two different approaches for selecting the comparison resources. The first approach automatically selects comparison resources based on similarity. The second approach lets users manually select resources.

3.1.1 Find similar resources. Comparing resources makes only sense when resources can sensibly be compared. For example, it does not make (much) sense to compare a city (e.g., *dbpedia:berlin*) to a car brand (e.g., *dbpedia:volkswagen*). This of course does not only apply to comparison in knowledge graphs but also applies to comparison in other kinds of databases. We thus argue that it makes only sense to compare resources that are similar. More specifically, resources that share the same (or a similar set of) properties are good comparison candidates. To illustrate this, consider following resources: *dbpedia:berlin* and *dbpedia:new_york_city*. Both resources share the property *dbo:populationTotal* which makes them suitable for comparison. Finding similar resources is therefore based on finding resources that share the same or similar properties.

To do so, each comparison resource is converted into a string. This string is generated by concatenating all properties of the resource (Algorithm 1). The resulting string is stored. TF-IDF [12] is used to query the store and the string for the main resource is used as query. The search returns the most similar resources. The top-k resources are selected and form a set of resources that is used in the next step.

Algorithm 1 Paper indexing

1: **procedure** INDEXPAPER(paper)
2: 　　**for each** *property* in *paper* **do**
3: 　　　　*propertyString* ← *propertyString* + *property*
4: 　　**save** *propertyString*

3.1.2 Manual selection. There are scenarios where comparison based on similarity is not suitable. For example, a user may want to compare Germany and France to see which country has the highest GDP. In this case, there is no need to automatically select resources to be compared because they are determined by the user. Therefore, manual selection of resources should also be supported. How to manually select resources is an implementation detail and a proposal for how to do this is presented in Section 4. The result of manual selection is a set of resources used for the comparison.

3.2 Select related statements

This step selects the statements related to the resources to be compared returned in the previous step. Statements are selected transitively to match resources in subject or object position. This search is performed until a predefined maximum transitive depth δ has been reached. The intuition is that the deeper a property is nested the less likely is its relevance for the comparison.

3.3 Map properties

As described in the first step, comparisons are built using shared or similar properties of resources. In case the same property has been used between resources, these properties are grouped and form one *comparison row*. However, often different properties are used to describe the same concept. This occurs for various reasons. The most obvious reason is when two different ontologies are used to describe the same property. For example, for describing the population of a city, DBpedia uses *dbo:populationTotal* while WikiData uses *WikiData:population* (actually the property identifier is P1082; for the purpose here we use the label). When comparing resources, these properties should be considered as equivalent. Especially for community-created knowledge graphs, differently identified properties likely exist that are, in fact, equivalent.

To overcome this problem, we use FastText [2] word embeddings to determine the similarity of properties. If the similarity is higher than a predetermined threshold τ, the properties are considered same and are grouped. In the end, each group of predicates will be visualized as one row in the comparison table. The result of this step is a list of statements for each comparison resource, where similar predicates are grouped.

The similarity matrix γ is generated

$$\gamma = \left[cos(\overrightarrow{p_i}, \overrightarrow{p_j}) \right] \quad (1)$$

with $cos(.)$ as the cosine similarity of vector embeddings for predicate pairs $(p_i, p_j) \in \mathcal{P}$, whereby $\mathcal{P}$ is the set of all resources.

Furthermore, we create a mask matrix Φ that selects predicates of resources $c_i \in C$, whereby C is the set of resources to be compared. Formally,

$$\Phi_{i,j} = \begin{cases} 1 & \text{if } p_j \in c_i \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

Next, for each selected predicate p we create the matrix φ that slices Φ to include only similar predicates. Formally,

$$\varphi_{i,j} = (\Phi_{i,j}) \begin{array}{l} c_i \in C \\ p_j \in sim(p) \end{array} \quad (3)$$

where $sim(p)$ is the set of predicates with similarity values $\gamma[p] \geq \tau$ with predicate p. Finally, φ is used to efficiently compute the common set of predicates [9]. This process is displayed in Algorithm 2.

3.4 Visualize comparison

The final step of the workflow is to visualize the comparison and present the data in a human understandable format. Tabular format is often appropriate for visualizing comparisons. Another aspect of the visualization is determining which properties should be

---

Algorithm 2 Property mapping

1: **procedure** MAPPROPERTIES(properties, threshold)
2:     **for each** property $p_1 \in properties$ **do**
3:         **for each** property $p_2 \in properties$ **do**
4:             $similarity \leftarrow cos(\text{FastText}(p_1), \text{FastText}(p_2))$
5:             **if** $similarity > threshold$ **then**
6:                 $similarProps \leftarrow similarProps \cup \{p_1, p_2\}$
    return $similarProps$

---

displayed and which ones should be hidden. A property is displayed when it is shared among a predetermined amount τ of papers, where τ mainly depends on comparison use and can be determined based on the total amount of resources in the comparison.

Another aspect of comparison visualization is the possibility to customize the resulting table. This is needed because of the similarity-based matching of properties and the use of predetermined thresholds. For example, users should be able to enable or disable properties. They should also get feedback on property provenance (i.e., the property path). Ultimately, this contributes to a better user experience, with the possibility to manually correct mistakes made by the system.

4 IMPLEMENTATION

The presented four-step workflow for comparing resources is implemented in the Open Research Knowledge Graph (ORKG), specifically to compare research contributions as a special type of resource. As a companion to the description here, an online video summarizes and demonstrates the ORKG comparison feature[11].

The user interface of the comparison feature is seamlessly integrated with the ORKG front end, which is written in JavaScript using the React framework[12] and is publicly available online[13]. The back end of the comparison feature is written as a service separate from the ORKG back end, is written in Python and is available online[14].

In the ORKG, each paper consists of at least one *research contribution* which addresses at least one *research problem* and is further described with contribution data including for instance materials, methods, implementation, results or other aspects. In the ORKG, it is *research contributions* that are compared rather than papers.

We will now discuss each step of the presented workflow to illustrate how it is implemented in the ORKG.

4.1 Select comparison candidates

Both approaches presented, namely find similar resources and manual selection, are implemented in the ORKG. The reason for implementing both is that they complement each other. Conducting a comparison based on similarity is useful when a user wants to compare a certain contribution with other (automatically determined similar) contributions (for example, addressing the same problem), while manual contribution selection can be helpful to compare a user defined set of contributions. Figure 2 shows both approaches. As depicted, three similar contributions are suggested

---

[11]https://youtu.be/mbe-cVyW_us
[12]https://reactjs.org/
[13]https://gitlab.com/TIBHannover/orkg/orkg-frontend
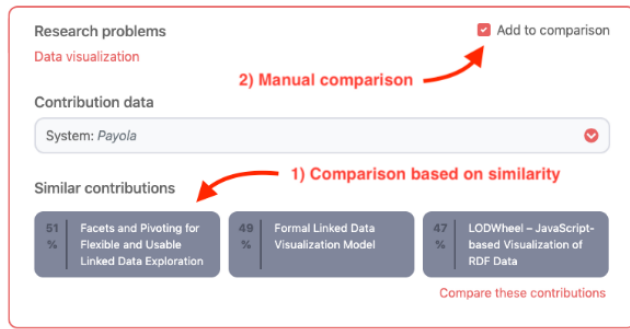[14]https://gitlab.com/TIBHannover/orkg/orkg-similarity

Figure 2: Implementation of workflow step 1, select comparison candidates, showing both the similarity-based and the manual selection approaches.

Figure 3: Box showing the manually selected contributions.

to the user. (The corresponding similarity percentage is displayed next to paper title.) These suggested contributions can be directly compared. In contrast, the manual approach works similarly to an online shopping cart. When the "Add to comparison" checkbox is checked, a box is displayed at the bottom of the page. This box shows the manually selected contributions that will be used for the comparison (Figure 3).

To retrieve contributions that are similar to a given contribution, we developed an API endpoint. This endpoint takes the given contribution as input and returns five similar contributions (of which three are displayed). For performance reasons, each contribution is indexed by concatenating the properties to a string (Section 3.1). This string is stored inside a document-oriented database. The indexing happens as soon as a contribution is added. The result of this step is a set of contribution IDs used to perform the comparison.

4.2 Select related statements

An additional API endpoint was developed for the comparison. This endpoint takes the set of contribution IDs as input and returns the data used to display the comparison. The comparison endpoint is responsible for steps two and three of the workflow: selecting the related statements and mapping the properties. For each listed contribution, an ORKG back end query selects all related statements. This is done as described in Section 3.2. The process of selecting statements is repeated until depth $\delta = 5$. This number is chosen to include statements that are not directly related to the resource, but to exclude statements that are less relevant because they are nested too deep.

4.3 Map properties

Using the API of the previous step, the properties of the selected statements are mapped. As described in the workflow, for each property pair the similarity is calculated using word embeddings. In case the similarity threshold $\tau \geq 0.9$, the properties are considered to be equivalent and are grouped. The threshold is determined by a trial and error method. Then the results from the API are returned to the UI where they are displayed.
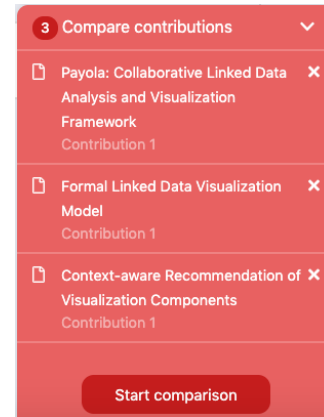
4.4 Visualize comparison

Because the comparisons are made for humans, visualizing them effectively is essential and therefore we invested considerable effort on this aspect. Figure 4 displays a comparison for research contributions related to visualization tools published in the literature. In this example, four properties are displayed. Literals are displayed as plain text while resources are displayed as links. When a resource link is selected, a popup is displayed showing the statements related to this resource. By default, only properties that are common to at least two contributions ($\tau \geq 2$) are displayed. The UI implements some additional features that are particularly useful to compare research contributions. We will now discuss these features is more detail.

4.4.1 Customization. Users can customize comparisons according to their needs. The customization includes transposing the table and customizing the properties. The properties can be enabled/disabled and they can be sorted. Especially the option to disable properties is helpful when resources with many statements are compared. Only properties considered relevant to the user can be selected to display. Customizing the comparison table can be useful before exporting or sharing the comparison.

4.4.2 Sharing and persistence. The comparison can be shared using a link. For sharing the comparison, a persistence mechanism has been built in. Especially when sharing the comparison for research purposes, it is important to share the original comparison. Since resource descriptions may change over time comparisons may also change. To support persistency, the whole state of the comparison is stored in a document-based database.

4.4.3 Export. It is possible to export comparisons in formats PDF, CSV and LaTeX. Especially the LaTeX export is useful for the ORKG, since the export be directly used in research papers. In addition to the generated LaTeX table, a BibTeX file is generated containing the bibliographic information of the papers used in the comparison. Also, a link referring back to the comparison by the ORKG is showed as footnote. Just like the shareable link, this link is persistent and is therefore suitable for use in articles.
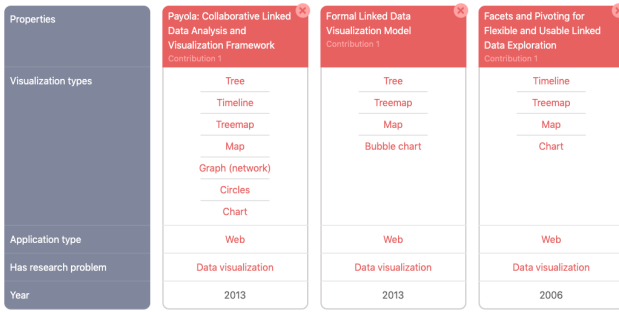
Figure 4: Comparison of research contributions related to visualization tools.

5 PRELIMINARY EVALUATION

In this section we present a preliminary evaluation of the implemented comparison functionality.

5.1 User evaluation

A qualitative evaluation is conducted to determine the usability of our implementation. Additionally, the evaluation is used to determine the usefulness of the comparison functionality in general. The usability is determined using the System Usability Scale (SUS) [3]. In total, five participants were part of the evaluation. All participants are researchers. At the start of the evaluation, each participant was asked to watch a video that explained the basic concepts of the comparison functionality. Afterwards an instructor asked the participant to perform certain tasks in the system, specifically creating a comparison (based on similarity and manually), customizing this comparison and exporting the comparison. The tasks were chosen to include all main functionalities of the comparison functionality. In case a participant was not able to complete the task he was allowed to ask an instructor for help. After interacting with the system, users were asked to fill out an online questionnaire[15]. The questionnaire contained ten questions from the SUS, each questions could be answered on a scale from 1 (strong disagree) to 5 (strongly agree). Afterwards, a short interview was conducted to get the opinions of the participants on the usefulness of the comparison feature.

The SUS score ranges from 1 to 100. In our evaluation, the SUS score is 81, which is considered excellent [5]. Figure 5 depicts the score per question. This indicates that participants did not have problems with using the user interface to create, customize and export their related work comparisons. This is in line with the positive verbal feedback that was provided to the instructor during the evaluation. In addition to the usability questions, three questions were asked related to the usefulness of the related literature comparison functionality. All participants agreed that such a functionality is useful and can potentially save them time while conducting research. Finally, participants were asked to give additional feedback. Among others, participant #1 remarked "*It would be nice if it is explained how similarity of papers is determined*"; participant #3 suggested "*Show text labels next to properties, explaining what this*
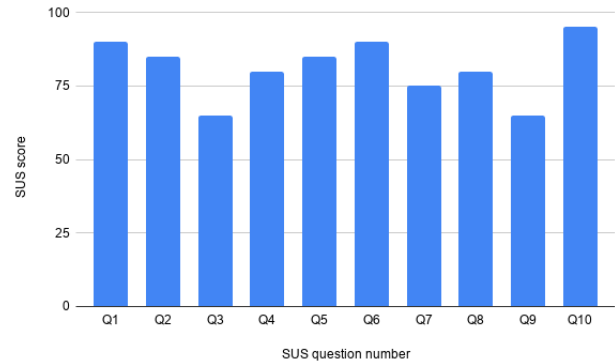

Figure 5: SUS score by question (higher is better)

Table 1: Time (in seconds) needed to perform comparisons with 2-8 research contributions using the baseline and ORKG approaches.

	Number of compared research contributions						
	2	**3**	**4**	**5**	**6**	**7**	**8**
Baseline	0.00026	0.1714	0.763	4.99	112.74	1772.8	14421
ORKG	0.0035	0.0013	0.01158	0.02	0.0206	0.0189	0.0204

property means"; participant #5 stated "*It should be possible to split properties that are mapped, but are in fact different*".

5.2 Performance evaluation

In order to evaluate the performance of the overall comparison, we compared the implemented ORKG approach to a baseline approach for comparing multiple resources. In Table 1 the time needed for a comparison is displayed for both the baseline and the ORKG approach. In total eight papers are compared with on average ten properties per paper. In the baseline approach, the "Map properties" step is not scaling well. This is because each property is compared against all other properties. If multiple contributions are selected, the amount of property similarity checks grows exponentially. As displayed in the table, the ORKG approach outperforms the baseline approach. The total amount of papers used for the evaluation is limited to eight because the baseline approach does not scale to larger sets.

6 DISCUSSION & FUTURE WORK

The aim of the contribution comparison functionality is to support literature reviews and make it more efficient. To live up to this aim, the knowledge graph should contain more data. As described in Section 2, structured data is needed to perform an effective and accurate comparison. Currently, such a graph containing research contributions does not exist since most existing initiatives focus solely on document metadata. This is why the ORKG focuses on making the actual research contributions machine-readable. Although the amount of papers in the ORKG is growing, it is currently not sufficient for the comparison functionality to be used effectively. The evaluation results suggest that the comparison feature performs well and that users are satisfied with the usability. Additionally,

---

[15]https://forms.gle/x2t7SYkAzCkCekUp8

they see the potential of the functionality. Thus, the technical infrastructure is in place for the related literature comparison but more data is needed for an extensive evaluation and real-world use.

In order to evaluate the usability of the interface, a user evaluation is arguably the most suitable method. In total there were only five participants for the user evaluation presented here. While this is not sufficient to make any definitive conclusions, it helps to understand what users expect from such a system. The individually provided feedback is also helpful to guide further developments. One of the important outcomes of the evaluation is that all participants agreed on the usefulness of the feature. They saw the potential of conducting literature reviews with the ORKG instead of doing it entirely manually.

Future work will focus on a more extensive evaluation of the individual components of the system. This includes the merging of properties and the similarity functionality. In order to perform such an evaluation, more data should be added to the ORKG. Given that automated related literature comparisons is one of the many advantages of structured scholarly knowledge, more functionalities leveraging this structured data will be developed. An example is faceted search, which provides an alternative to the full-text search commonly used to find related literature.

7 CONCLUSION

The presented workflow shows how research contributions in a graph database can be compared, which answers our first research question. The workflow consists of four steps in which comparison candidates are selected, related statements are fetched, properties are mapped and finally the comparison is visualized. We presented, evaluated and discussed an implementation of the workflow in the ORKG. The implementation answers our second research question by showing how the comparisons can be effectively visualized in a user interface. The performance evaluation results show that the system scales well. The user evaluation indicates that users see the potential of a related literature comparison functionality, and that the current implementation is user-friendly.

ACKNOWLEDGMENTS

REFERENCES

[1] Samur Araujo, Jan Hidders, Daniel Schwabe, and Arjen P. De Vries. 2011. SERIMI - Resource description similarity, RDF instance matching and interlinking. *CEUR Workshop Proceedings* 814 (2011), 246–247.

[2] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics* 5 (2017), 135–146. https://doi.org/10.1162/tacl_a_00051

[3] John Brooke et al. 1996. SUS-A quick and dirty usability scale. *Usability evaluation in industry* 189, 194 (1996), 4–7.

[4] Alexandru Constantin, Silvio Peroni, Steve Pettifer, David Shotton, and Fabio Vitali. 2016. The Document Components Ontology (DoCO). *Semantic Web* 7, 2 (2016), 167–181. https://doi.org/10.3233/SW-150177

[5] Tim Donovan, Lambert M. Felix, James D. Chalmers, Stephen J. Milan, Alexander G. Mathioudakis, and Sally Spencer. 2018. Continuous versus intermittent antibiotics for bronchiectasis. *Cochrane Database of Systematic Reviews* 2018, 6 (2018), 114–123. https://doi.org/10.1002/14651858.CD012733.pub2

[6] Aldo Gangemi, Silvio Peroni, David Shotton, and Fabio Vitali. 2017. The Publishing Workflow Ontology (PWO). *Semantic Web* 8, 5 (2017), 703–718. https://doi.org/10.3233/SW-160230

[7] Dagmar Gromann and Thierry Declerck. 2019. Comparing pretrained multilingual word embeddings on an ontology alignment task. *LREC 2018 - 11th International Conference on Language Resources and Evaluation* (2019), 230–236.

[8] Alexander Hars. 2001. Designing Scientific Knowledge Infrastructures: The Contribution of Epistemology. *Information Systems Frontiers* 3, 1 (2001), 63–73. https://doi.org/10.1023/A:1011401704862

[9] Mohamad Yaser Jaradeh, Allard Oelen, Manuel Prinz, Jennifer D'Souza, Gábor Kismihók, Markus Stocker, and Sören Auer. 2019. Open Research Knowledge Graph: Next Generation Infrastructure for Semantic Scholarly Knowledge (in press). In *In Proceedings of the 10th International Conference on Knowledge Capture (K-CAP '19)*. ACM. https://doi.org/10.1145/3360901.3364435

[10] Vladimir I Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, Vol. 10. 707–710.

[11] Pierre Maillot, Carlos Bobed, Pierre Maillot, Carlos Bobed, Pierre Maillot, and Carlos Bobed. 2019. Measuring structural similarity between RDF graphs To cite this version : HAL Id : hal-01940449 Measuring Structural Similarity Between RDF Graphs. (2019).

[12] Carme Pinya Medina and Maria Rosa Rosselló Ramon. 2015. Using TF-IDF to Determine Word Relevance in Document Queries Juan. *New Educational Review* 42, 4 (2015), 40–51. https://doi.org/10.15804/tner.2015.42.4.03

[13] Silvio Peroni and David Shotton. 2012. FaBiO and CiTO: Ontologies for describing bibliographic resources and citations. *Journal of Web Semantics* 17 (2012), 33–43. https://doi.org/10.1016/j.websem.2012.08.001

[14] Silvio Peroni and David Shotton. 2018. The SPAR ontologies. In *International Semantic Web Conference*. Springer, 119–136.

[15] Alina Petrova, Evgeny Sherkhonov, Bernardo Cuenca Grau, and Ian Horrocks. 2017. Entity comparison in RDF graphs. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 10587 LNCS (2017), 526–541. https://doi.org/10.1007/978-3-319-68288-4_31

[16] Almudena Ruiz Iniesta and Oscar Corcho. 2014. A review of ontologies for describing scholarly and scientific documents. In *4th Workshop on Semantic Publishing (SePublica) (CEUR Workshop Proceedings)*. http://ceur-ws.org/Vol-1155#paper-07

[17] Pavel Shvaiko and Jérôme Euzenat. 2013. Ontology matching: State of the art and future challenges. *IEEE Transactions on Knowledge and Data Engineering* 25, 1 (2013), 158–176. https://doi.org/10.1109/TKDE.2011.253

[18] William E. Winkler. 1990. String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage. *Proceedings of the Section on Survey Research, American Statistical Association* (1990), 354–359. https://doi.org/10.1007/978-1-4612-2856-1_101

[19] Paweł Ziemba, Jarosław Jankowski, and Jarosław Wątróbski. 2017. Online comparison system with certain and uncertain criteria based on multi-criteria decision analysis method. In *International Conference on Computational Collective Intelligence*. Springer, 579–589.