Wiki Workshop (12th edition) – May 21-22, 2025

Introducing ProVe for Wikidata: Automatic Verification of References

Elena Simperl[1], Odinaldo Rodrigues[1], Albert Meroño-Peñuela[1], Gabriel Maia Rocha Amaral[1], Nathan Gavenski[1], Jongmo Kim[1], Miriam Redi[2], Yihang Zhao[1]

[1]Department of Informatics, King's College London, UK
[2]Wikimedia Foundation, UK

Abstract

This paper describes key aspects of ProVe – a Wikidata gadget and accompanying Web API for automatic verification of references of Wikidata items. ProVe can be integrated into the regular Wikidata editing workflow to help identify statements about items whose references need attention. ProVe automatically verbalises item statements to assess the support of the statement by an external document and provide an indicative score of the item's quality of references.

Keywords: reference verification, fact verification, knowledge graphs, Wikidata tools

Introduction

A Knowledge Graph (KG) is a large network of interconnected entities, encoding their properties and relationships to one another (Krötzsch and Weikum, 2016; Paulheim, 2016). KGs serve as sources of machine-readable and semantically structured data used by several web applications, including Wikipedia infoboxes, search engines, voice-activated assistants, among others (Malyshev et al., 2018; Ji et al., 2022). The information in most KGs is stored as a set of statements – semantic triples of the form ⟨subject, predicate, object⟩, denoting a property of the subject of the triple (Färber et al., 2018). Ensuring KGs are trustworthy depends on well-documented and verifiable provenance to the information they encode (Zaveri et al., 2016), e.g. references. For example, if we have the statement ⟨beer, has characteristic, bitterness⟩, then we expect it to be supported by a reference that states that indeed beer has the characteristic of being bitter.

Mechanisms that help to evaluate and ensure the quality of information provenance are thus crucial to the verifiability of KGs (Zaveri et al., 2016; Piscopo et al., 2017; Xiangyu et al., 2021). However, such processes are currently mostly performed manually (McAndrew and Strathmann, 2021) and do not scale with size. On vital KGs such as Wikidata and DBpedia, manual reference verification is prohibitive due to their sheer size (Piscopo et al., 2017) – Wikidata has currently over 1.65 billion statements.

ProVe (Provenance Verification) uses research findings that have been designed and evaluated by the Wikidata community, responding to their data assurance needs (Amaral et al., 2022; Amaral et al., 2023). It consists of a "gadget" and Wikidata API built upon state-of-the-art NLP models, public datasets on data verbalisation and fact verification, and rule-based methods. Given a statement about an item and reference URL, ProVe verbalises the statement, automatically retrieves the referenced document, and then selects the passages in the document that are most relevant to the statement. ProVe then evaluates the overall stance of the referenced document for the statement, assessing whether it is supported by the reference. This is then aggregated to provide an overall quality score for the item.

In this paper, we give an overall description of ProVe's functionality and how it is being made available to the wider community. We present some important technical details about the reference evaluation process, concluding with a discussion of limitations and plans for future work.

Overview

ProVe is currently hosted at King's College London. It consists of five main components: the main server processing user requests; some ML models used for natural language tasks; a *gadget* that summarises the assessment of individual Wikidata item statements; a Web API which can be used in client applications; and the main database storing results of statement evaluations. Figure 1 gives an overview of these components.

ProVe's Gadget

Most users can easily interact with ProVe via its *gadget*, installed according to the instructions found on `https://www.wikidata.org/wiki/Wikidata:ProVe`. The gadget works at the level of a Wikidata item, systematically checking every statement about the item that is accompanied by an external reference. For each such statement-reference pair, ProVe displays its support stance and the passage found in the external reference as justification for its evaluation. The individual statement assessments are then combined to give an overall quality indicator between −1 and 1 and displayed in the item's Wikidata page (see Technical Details and Figure 1).

Web API

The gadget is intended for simple, item-oriented information about the quality of external references. For programmatic applications that need information about multiple items, ProVe provides a Web API. Through the API, applications can check whether the references of specific items have been previously evaluated, request for items to be (re-)checked, submit requests for item evaluation, and obtain historical verification results.

Technical Details of Statement Evaluation

The general workflow of ProVe's reference verification process is depicted in Figure 2.

As we mentioned, a Wikidata statement is recorded as a triple ⟨subject, predicate, object⟩. Given a Wikidata statement t and a document r used as reference for t, ProVe first verbalises t using a pre-trained transformer[1] to generate a sentence $v(t)$ conveying the claim encoded by t. It then retrieves the document r and segments it into a set of passages (chunks of texts) P, using spaCy's sentence segmenter and the *en_core_web_lb* model (Honnibal et al., 2020). Relevance scores ρ_i for each passage are computed next, using a pre-trained BERT transformer fine-tuned on the FEVER dataset (Thorne et al.,), and the five passages $\{p_1, \ldots, p_5\}$ that are most contextually relevant to $v(t)$ (independently of their stance) are selected. ProVe will then evaluate text entailment for each such sentence-passage pair $(v(t), p_i)$ using a pre-trained BERT model fine-tuned on FEVER for RTE. This ultimately gives each statement t and reference r a stance label $z = S(t, r) \in \{-1, 0, 1\}$, where -1 indicates that r refutes t; 1 indicates that r supports t; and 0 indicates that the support of t by r is inconclusive.[2] For completeness, we define $S(t, r) = 0$, for any irretrievable reference r, since r's support stance for t cannot be determined by ProVe in these cases. A summary with the inferred stances of the pairs $S(t, r)$ for every external reference r is displayed in a table by ProVe's gadget, alongside the most relevant sentence for t found in r, and links to the statements for edition. This facilitates the edition process and gives further context to the editor about the evaluation.

ProVe Score To give an indication of how well-supported (or refuted) an item i is by its *external* references, the support stances of all of its statements are aggregated. Let $T(i)$ be the set of all statements for which i is the subject, and $ER(t)$ be the set all external references for the statement t. The *ProVe score* for i is defined as $PS(i) = \sum_{t \in T(i)} \sum_{r \in ER(t)} \{S(t,r)\} / |\cup_{t \in T(i)} ER(t)|$. It is easy to see that $PS(i)$ is a value in $[-1, 1]$ with the following intended meaning. Positive values indicate that the number of supporting references surpasses the number of refuting references; negative values indicate the opposite; with the proportion of references which are inconclusive or cannot be retrieved, bringing the score closer to 0. Hence, proximity to 1 is associated with good quality of the references of an item, proximity to 0 is associated with references of a mixed nature, and proximity to -1 indicates high levels of disparity between claims and references.

The ProVe score is shown to editors in the gadget when the Wikidata page for the item is loaded (see Figure 1, bottom right). Computation or re-computation of scores can be requested by pressing appropriate buttons. Recalculation can be useful to track progress in the improvement of the references for items under revision.

Limitations

ProVe can take any non-ontological KG triple as long as its components are accompanied by labels in English. This requirement is because the NLP modules so far have only been trained for English. Future extensions to other languages are under consideration. In addition, ProVe works on *external* references only (arguably harder to verify). Visual elements, such as images and charts, can serve as evidence for KG statements. However, the automated extraction of text from such evidence in a format that language models can understand is not trivial and ProVe cannot yet deal with these types of reference. Finally, ProVe employs a combination of techniques, including pre-trained sentence segmenters, to extract passages from various forms of structured text, e.g., in tables, but this is not always effective.

Community Involvement

We welcome suggestions of the community for improvements to ProVe and development of extra functionality. Users can register interest by adding their names to `https://www.wikidata.org/wiki/Wikidata:ProVe`.

Discussion and Future Work

ProVe is limited to the verification of statements presented as triples, not being directly applicable to full sentences, and hence cannot be used in applications such as Wikipedia. However, it is technically possible to skip the verbalisation and continue the evaluation from there. For this to be effective, we need a mechanism to properly extract from the input text the sentence to verify, and ensure the NLP models employed in the relevance and entailment tasks are suitable for the domain of the application. This is left as a potential future improvement.

[1] A T5-base model (Raffel et al., 2020) fine-tuned on the WebNLG 2017 dataset (Gardent et al., 2017).

[2] This is a slight simplification. More information comes out of the entailment verification and potentially different aggregation techniques can be used.

Figure 1: Overview of ProVe's system architecture and gadget interface.

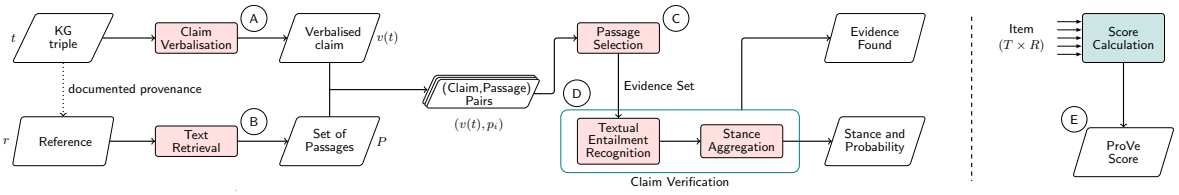

Figure 2: Overview of ProVe's reference verification process.

References

[Amaral et al.2022] G. Amaral, O. Rodrigues, and E. Simperl. 2022. WDV: A broad data verbalisation dataset built from Wikidata. In U. Sattler et. al., editor, *The Semantic Web – ISWC 22*, pages 556–574, Cham. Springer International Publishing.

[Amaral et al.2023] G. Amaral, O. Rodrigues, and E. Simperl. 2023. Prove: A pipeline for automated provenance verification of knowledge graphs against textual sources. *The Semantic Web Journal*.

[Färber et al.2018] M. Färber, F. Bartscherer, C. Menne, and A. Rettinger. 2018. Linked data quality of DBpedia, freebase, opencyc, Wikidata, and yago. *Semantic Web*, 9(1):77–129.

[Gardent et al.2017] Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. The webnlg challenge: Generating text from rdf data. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 124–133.

[Honnibal et al.2020] Ma. Honnibal, I. Montani, S. Van Landeghem, and A. Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

[Ji et al.2022] S. Ji, S. Pan, E. Cambria, P. Marttinen, and P. S. Yu. 2022. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Transactions on Neural Networks and Learning Systems*, 33(2):494–514.

[Krötzsch and Weikum2016] M. Krötzsch and G. Weikum. 2016. JWS special issue on KGs.

[Malyshev et al.2018] S. Malyshev, M. Krötzsch, L. González, J. Gonsior, and A. Bielefeldt. 2018. Getting the most out of Wikidata: semantic technology usage in wikipedia's knowledge graph. In *International Semantic Web Conference*, pages 376–394. Springer.

[McAndrew and Strathmann2021] E. McAndrew and C. Strathmann. 2021. Quality assurance and reliability, Aug.

[Paulheim2016] H. Paulheim. 2016. Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic Web*, 8:489–508.

[Piscopo et al.2017] A. Piscopo, L.-A. Kaffee, C. Phethean, and E. Simperl. 2017. Provenance information in a collaborative knowledge graph: an evaluation of Wikidata external references. In *International semantic web conference*, pages 542–558. Springer.

[Raffel et al.2020] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. Liu, J. Thorne, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

[Thorne et al.] J. Thorne, A. Vlachos, C. Christodoulopoulos, and A. Mittal. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans. ACL.

[Xiangyu et al.2021] W. Xiangyu, C. Lyuzhou, T. Ban, M. Usman, G. Yifeng, L. Shikang, W Tianhao, and C. Huanhuan. 2021. Knowledge graph quality control: A survey. *Fundamental Research*, 1(5):607–626.

[Zaveri et al.2016] A. Zaveri, A. Rula, A. Maurino, R. Pietrobon, J. Lehmann, and S. Auer. 2016. Quality assessment for linked data: A survey. *Semantic Web*, 7(1):63–93.