Automatic Verification of References of Wikidata Statements

Elena Simperl
elena.simperl@kcl.ac.uk

Odinaldo Rodrigues
odinaldo.rodrigues@kcl.ac.uk

Albert Meroño-Peñuela
albert.merono@kcl.ac.uk

Kholoud Saad Alghamdi
kholoud.alghamdi@kcl.ac.uk

Gabriel Maia Rocha Amaral
gabriel.amaral@kcl.ac.uk

Nathan Schneider Gavenski
nathan.schneider_gavenski@kcl.ac.uk

Jongmo Kim
jongmo.kim@kcl.ac.uk

Miriam Redi
mredi@wikimedia.org

Yiwen Xing
yiwen.xing@eng.ox.ac.uk

Bohui Zhang
bohui.zhang@kcl.ac.uk

Yihang Zhao
yihang.zhao@kcl.ac.uk

Abstract

Wikidata is one of the world's most important data assets. It is used by search engines, virtual assistants, fact checkers, and in over 800 Wikimedia projects. Wikidata contains 1.65 billion statements about over 116 million data items, edited by nearly 25 thousand editors. Manually checking whether Wikidata statements are supported by references is a slow process that does not scale with size. Given the overall number of statements to check, the collaborative nature of Wikidata, and the fact that referenced documents can change over time, preserving the quality of the references is an onerous process requiring continuous intervention. This paper present ProVe – a tool to assist in the automatic verification and assessment of the quality of the references of Wikidata items.

Keywords: reference verification, quality assurance, tools, Wikidata

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, and voice-activated assistants, amongst others (Ji et al. 2022; Malyshev et al. 2018). In most KGs, information is stored as a set of statements, semantic triples of the form ⟨*subject*, *predicate*, *object*⟩, denoting a property of the subject in 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). For example, if the statement "beer is bitter" is encoded in a KG by the triple *⟨beer, has characteristic, bitterness⟩*, this triple should reference a source which supports the statement.

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; Wang et al. 2021). However, such processes are currently mostly performed manually (McAndrew and Strathmann 2021) and do not scale with size. Yet, on vital KGs such as Wikidata and DBpedia, manual verification is prohibitive due to their sheer size (Piscopo et al. 2017)—Wikidata has currently over 1.65 billion statements—and more support to assist with verification is needed.

ProVe (Provenance Verification) leverages research conducted for and evaluated by the Wikidata community, responding to their data assurance needs (Amaral et al. 2021; Amaral, Rodrigues and Simperl 2022; Amaral, Rodrigues, and Simperl 2023). It consists of an add-on user interface embedded in Wikidata's editing pages (a Wikidata gadget) and web API, which use Natural Language Processing (NLP) models, public datasets on data verbalisation and fact verification, and rule-based methods.

Given a Wikidata statement and an external URL reference (e.g., through the P854 property or an external identifier property that has formattable URLs), ProVe verbalises the statement, automatically retrieves the referenced document and looks for the passages within it that are relevant to the statement's claim. ProVe then evaluates the overall stance of the referenced document with respect to the statement, displaying the stance along with the most relevant passage found as justification. This process is repeated for all statements whose subject is the given item, with the results summarised in a convenient, unobtrusive table.

ProVe is work in progress, but we have an MVP whose functionality is described in this paper, along with important technical details about the reference evaluation process. The paper then concludes with a discussion of limitations and plans for future work.

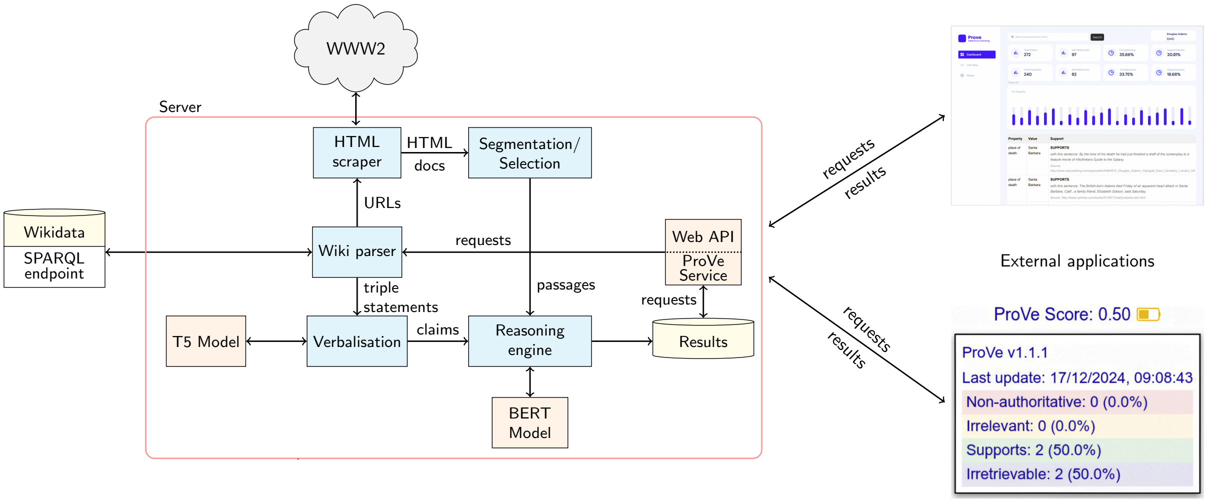
Related work

FEVER (Fact Extraction and VERification) (Thorne et al. 2018) is a large dataset containing over 185K claims, that can be used for fact verification against textual sources. Despite its general applicability, claim verification in KGs faces additional challenges because statements need to be turned into sentences first. This "verbalisation" process needs to consider important assumptions about the way information is represented in the KG (more on this later). ProVe employs FEVER for some natural language tasks (see the Reference Verification Process section).

Several works have previously focused on the verification of the quality of information in KGs (McAndrew and Strathmann 2021; Piscopo et al. 2017; Wang et al. 2021; Zaveri et al. 2016; Amaral, Rodrigues, and Simperl 2023; Amaral, Rodrigues, and Simperl 2022; Amaral et al. 2021). However, to the best of our knowledge, ProVe is the only available tool integrated within Wikidata and based on published research that *automates* the reference verification process. ProVe has been running since Summer 2024, even though we only started collecting detailed usage information since March 2025. Our statistics show we have been receiving around 400 requests a day, coming from 22 countries and all continents.

General Overview

Figure 1 gives an overview of ProVe's architecture. It consists of the main server that processes user requests submitted from the gadget or API; ML models within it used for some natural language tasks; a web service and API which can be used in external applications; a Wikidata *gadget* that interacts with editors and summarises the assessment of references of individual items; and the main database storing the results of evaluations for future analyses.



ProVe Score: 0.50

ProVe v1.1.1
Last update: 17/12/2024, 09:08:43
Non-authoritative: 0 (0.0%)
Irrelevant: 0 (0.0%)
Supports: 2 (50.0%)
Irretrievable: 2 (50.0%)

Figure 1. Overview of ProVe's system architecture

Wikidata Gadget

Most users can easily interact with ProVe via its *gadget*, which can be installed according to the instructions found on Wikidata's project page for the tool. Once installed, the gadget works within Wikidata's item editing page, systematically checking externally referenced statements. For each statement-reference pair, ProVe will then display the computed support stance of the referenced document along the most relevant passage found within the document. This is all conveniently displayed in a sortable table, with links to the editing sections within the page for each of the statements. All individual statement-reference quality assessments are then combined to give a number in the [-1,1] interval—ProVe's quality score for the item. This is displayed within the Wikidata page (see infobox, bottom right of Figure 1), but can also be retrieved programmatically for analysis outside Wikidata (see below).

Web API

The gadget is intended for simple, item-oriented information about the quality of external references of statements. For programmatic applications that need information about multiple items at specific points in time, ProVe provides a Web API too. Through the API, external applications can check whether the references of items have been previously analysed, request them to be (re-)evaluated, obtain summarised and comprehensive evaluation results, as well as historical evaluation data.

Reference Verification Process

The general workflow of ProVe's reference verification process is shown in Figure 2. For each statement and associated external reference, ProVe first verbalises the statement (A), then retrieves the text of the referenced document, removing markup elements and segmenting it into passages (B). Passages are then ranked according to their relevance to the verbalised statement (C), and the support stance of the most relevant passages with respect to the statement analysed. The overall support stance of the referenced document for the statement is computed and provided along with the evidence found (D). This is typically the passage within the document found to be the most relevant to the verbalisation of the statement. Finally, the results for each statement-reference pair are aggregated to give the user the item's ProVe score (E). Further details of each step are given below.

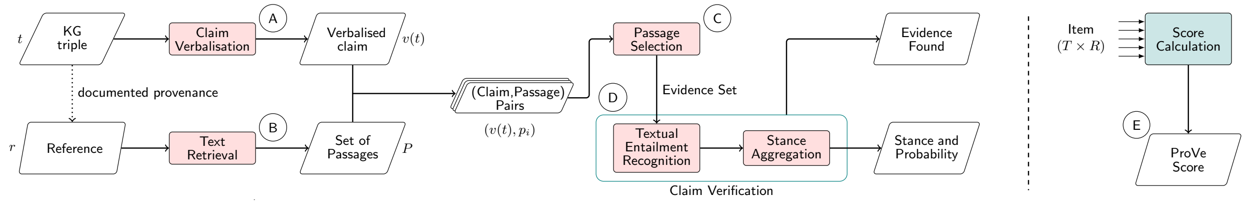


Figure 2. Overview of the reference quality evaluation process

Verbalisation

Although the verbalisation of a statement such as *⟨beer, has characteristic, bitterness⟩* is arguably simple, in general there are several hurdles to overcome. Firstly, components of a statement <*subject, predicate, object*> are provided with a set of alternative labels describing the component, and a judicious choice for a suitable label needs to be made. Secondly, there are implicit assumptions about the roles played by the *subject* and *object* of the relationship, as well as what the relationship is used for. For example, in the triple <*william, child, george*> the intended meaning is that the subject of the statement is the parent, and its object is the offspring. Although just conventions, these assumptions need to be considered to produce faithful verbalisations. To generate verbalisations that are fluent and resemble natural text, ProVe employs a T5-base model (Raffel et al. 2020) fine-tuned on the WebNLG 2017 dataset (Gardent et al. 2017). As part of original research done to underpin ProVe, a dataset with verbalised Wikidata statements was generated and made publicly available (Amaral, Rodrigues and Simperl 2022).

Text Retrieval

Layout and other structural information embedded into referenced documents make the extraction and meaningful re-combination of text non-trivial. In addition, the text itself can contain semantical constructs spread across sentences making ad-hoc segmentation not suitable for subsequent entailment evaluation of the passages. ProVe employs several custom rules to transform and remove HTML markup, after which the text is segmented using spaCy's sentence segmenter with the *en_core_web_lb* model (Honnibal et al. 2020). Pairs of consecutive segments are generated to cater for constructs such as pronominal anaphora, and single and combined segments are sent for subsequent passage selection.

Passage Selection

Once the claim has been verbalised and the referenced text segmented into passages, passages are ranked according to their relevance to the claim (independently of their support stance, which is analysed later). ProVe uses a pre-trained BERT transformer (Soleimani, Monz, and Worring 2020) fine-tuned on the FEVER dataset to give each passage a relevance score in the interval [-1,1]. Since some of the passages overlap (due the combinations described in the previous step), ProVe only keeps the five highest ranked passages that do not overlap along with their relevance scores. The scores are used in the claim verification step below.

Claim Verification

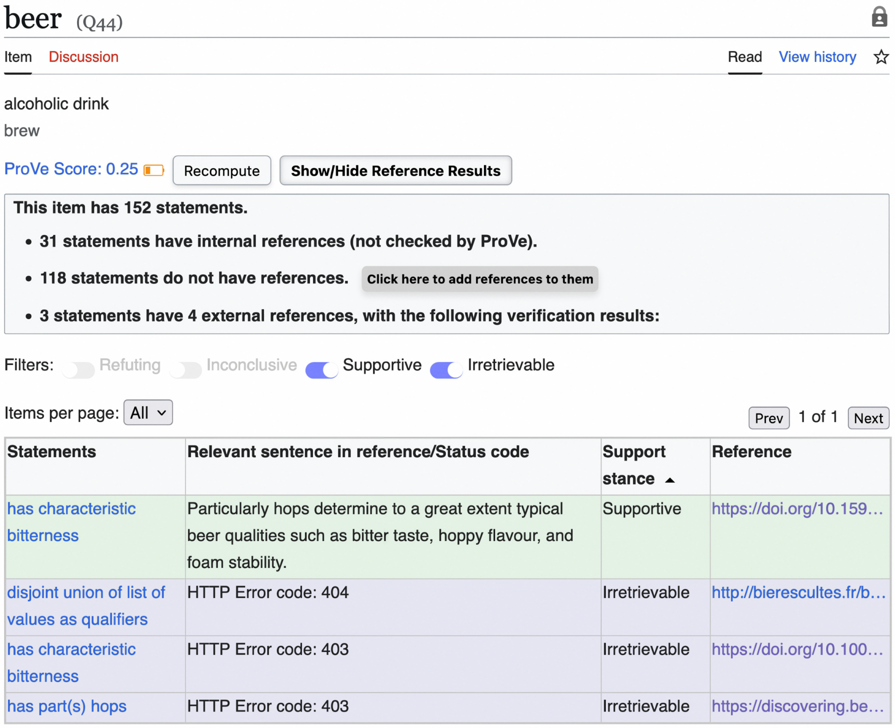
Evaluating the support stance of the referenced document for the statement as whole is performed in two stages. First, the stance of each of the most relevant passages is evaluated by a pre-trained BERT model (also fine-tuned on FEVER) for recognising textual entailment yielding a probability distribution for the classes *supportive, refuting,* and *not enough information* (i.e., inconclusive). At a second stage, the relevance scores of all passages selected along with their support stance probability distributions are aggregated to give an overall strength and support stance of the document for the statement.

ProVe Score (E)

Evaluating the support of individual statements by their external references is critical, but for editors working on items, an overall indication of the support for the item is also very important. Since a Wikidata item T can appear as the subject of many statements, each of which can have many references, the support stances of the item's statements need to be aggregated. This is done as follows. Let S(T)=[s1,…,sn] be the support stances for all statement-reference pairs of the item T calculated as described in the claim verification step above, where the value si (i=1,...,n) is either -1 for refuting references, 0 (for inconclusive), or 1 (for supportive).[1] The ProVe score for item T, in symbols PS(T), is calculated as the sum of values in S(T) divided by n.

It should be easy to see that PS(T) 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 and negative values indicate the opposite. In either case, inconclusive references bring the score closer to 0. As a result, proximity to 1 is associated with “good” quality of the references; proximity to 0 is associated with inconclusive or missing references; and proximity to −1 indicates high levels of disparity between claims and references. Of course, there is scope to provide customised scores that consider different types of statements and references.

The ProVe score of an item along with a summary of the total of references in each stance category is shown to editors when the Wikidata page for the item is loaded (see “infobox” at the bottom right of Figure 1). A summary table with sortable columns and filtering capability also provides convenient links to statements for potential correction of references (see Figure 3).



alcoholic drink
brew

This item has 152 statements.

- **31 statements have internal references (not checked by ProVe).**
- **118 statements do not have references.**
- **3 statements have 4 external references, with the following verification results:**

Statements	Relevant sentence in reference/Status code	Support stance ▲	Reference
has characteristic bitterness	Particularly hops determine to a great extent typical beer qualities such as bitter taste, hoppy flavour, and foam stability.	Supportive	https://doi.org/10.159…
disjoint union of list of values as qualifiers	HTTP Error code: 404	Irretrievable	http://bierescultes.fr/b…
has characteristic bitterness	HTTP Error code: 403	Irretrievable	https://doi.org/10.100…
has part(s) hops	HTTP Error code: 403	Irretrievable	https://discovering.be…

Figure 3. ProVe’s main user interface in Wikidata

Users can request the computation or re-computation of scores by pressing appropriate buttons in the interface. The scores allow users to factor in reference information in the

[1] For completeness, ProVe classifies irretrievable references as inconclusive since it cannot evaluate their stances w.r.t. their claims.

prioritisation of items to edit and to perform custom analyses with the extra information provided via the web API, e.g., the progress achieved in reference quality improvement of specific items.

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. Extensions to other languages are under consideration. ProVe focuses on *external* references only, since these are arguably harder to verify. In addition, visual elements, such as pictures and charts, can serve as evidence for KG statements. However, the automated extraction of text from these types of evidence in a format that language models can understand is not trivial and ProVe cannot deal with them yet. ProVe employs a combination of techniques (including pre-trained sentence segmenters) to extract passages from various forms of structured text, e.g., within tables, but this is not always effective. Finally, we are working towards parsing PDF documents too, but this is not yet operational.

Community Involvement

We welcome suggestions of the community for improvements to ProVe and the development of new functionality. Users can register interest by adding their names to the tool's participants list.

Discussion and Future Work

ProVe is limited to the verification of statements presented as triples, not being directly applicable to actual sentences in natural language. Hence, it cannot be used in applications such as Wikipedia. However, it is technically possible to skip the verbalisation phase (step (A) in Figure 2), replace the verbalised statement by a sentence of interest, and continue the evaluation process from there. For this to be effective, we would need a mechanism to properly extract from the input text the sentence we would like to verify and ensure the NLP model(s) employed in the relevance and entailment tasks are suitable for the domain of the text. This is left as a potential future extension.

We have identified particularly challenging types of references that would benefit from special treatment in the text retrieval phase (step (B) in Figure 2). Alternative mechanisms for text segmentation for use in subsequent textual entailment recognition are under consideration. Finally, we would like to make available to end users more of the information obtained during the evaluation process. This is more easily done by modifying the end points of the Web API, although further tweaks to the Wikidata gadget are also possible.

References

Amaral, Gabriel, Alessandro Piscopo, Lucie-Aimée Kaffee, Odinaldo Rodrigues, and Elena Simperl. 2021. “Assessing the Quality of Sources in Wikidata Across Languages: A Hybrid Approach.” *Journal of Data and Information Quality (JDIQ)* 13 (4): 1–35.

Amaral, Gabriel, Odinaldo Rodrigues, and Elena Simperl. 2023. “ProVe: A Pipeline for Automated Provenance Verification of Knowledge Graphs against Textual Sources.” *The Semantic Web Journal*. https://doi.org/https://doi.org/10.3233/SW-233467.

Amaral Gabriel and Rodrigues, Odinaldo and Simperl Elena. 2022. “WDV: A Broad Data Verbalisation Dataset Built from Wikidata.” In *The Semantic Web – ISWC 2022*, edited by Aidan and Keet Maria and Presutti Valentina and Almeida João Paulo A. and Takeda Hideaki and Monnin Pierre and Pirrò Giuseppe and d’Amato Claudia Sattler Ulrike and Hogan, 556–74. Cham: Springer International Publishing.

Färber, Michael, Frederic Bartscherer, Carsten Menne, and Achim Rettinger. 2018. “Linked Data Quality of DBpedia, Freebase, Opencyc, Wikidata, and Yago.” *Semantic Web* 9 (1): 77–129.

Gardent, Claire, 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*, 124–33.

Honnibal, Matthew, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. “SpaCy: Industrial-Strength Natural Language Processing in Python.” https://doi.org/10.5281/zenodo.1212303.

Ji, Shaoxiong, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S Yu. 2022. “A Survey on Knowledge Graphs: Representation, Acquisition, and Applications.” *IEEE Transactions on Neural Networks and Learning Systems* 33 (2): 494–514. https://doi.org/10.1109/TNNLS.2021.3070843.

Krötzsch, Markus, and Gerhard Weikum. 2016. “JWS Special Issue on Knowledge Graphs.” *International Center for Computational Logic*. Journal of Web Semantics. https://iccl.inf.tu-dresden.de/web/JWS_special_issue_on_Knowledge_Graphs/en.

Malyshev, Stanislav, Markus Krötzsch, Larry González, Julius Gonsior, and Adrian Bielefeldt. 2018. “Getting the Most out of Wikidata: Semantic Technology Usage in Wikipedia’s Knowledge Graph.” In *International Semantic Web Conference*, 376–94.

McAndrew, Ewan, and Clea Strathmann. 2021. “Quality Assurance and Reliability.” *Wikidata Quality Assurance and Reliability*. The University of Edinburgh.

https://www.ed.ac.uk/information-services/help-consultancy/is-skills/wikimedia/wikidata/quality-assurance-and-reliability.

Paulheim, Heiko. 2016. “Knowledge Graph Refinement: A Survey of Approaches and Evaluation Methods.” *Semantic Web* 8:489–508.

Piscopo, Alessandro, Lucie-Aimée Kaffee, Chris Phethean, and Elena Simperl. 2017. “Provenance Information in a Collaborative Knowledge Graph: An Evaluation of Wikidata External References.” In *International Semantic Web Conference*, 542–58.

Raffel, Colin, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, and others. 2020. “Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer.” *J. Mach. Learn. Res.* 21 (140): 1–67.

Soleimani, Amir, Christof Monz, and Marcel Worring. 2020. “BERT for Evidence Retrieval and Claim Verification.” In *European Conference on Information Retrieval*, 359–66.

Thorne, James, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. “FEVER: A Large-Scale Dataset for Fact Extraction and VERification.” In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 809–19. New Orleans, Louisiana: Association for Computational Linguistics. https://doi.org/10.18653/v1/N18-1074.

Wang, Xiangyu, Lyuzhou Chen, Taiyu Ban, Muhammad Usman, Yifeng Guan, Shikang Liu, Tianhao Wu, and Huanhuan Chen. 2021. “Knowledge Graph Quality Control: A Survey.” *Fundamental Research* 1 (5): 607–26. https://doi.org/https://doi.org/10.1016/j.fmre.2021.09.003.

Zaveri, Amrapali, Anisa Rula, Andrea Maurino, Ricardo Pietrobon, Jens Lehmann, and Soeren Auer. 2016. “Quality Assessment for Linked Data: A Survey.” *Semantic Web* 7 (1): 63–93.