

# Who Can Fix This? User Recommendation for Knowledge Graph Repair via Embedding-Based Clustering

Nicolas Ferranti

Vienna University of Economics and Business  
Vienna, Austria  
nicolas.ferranti@wu.ac.at

Jairo Francisco de Souza

Universidade Federal de Juiz de Fora  
Juiz de Fora, Brazil  
jairo.souza@ufjf.br

Dayane Guimaraes

Universidade Federal de Juiz de Fora  
Juiz de Fora, Brazil  
02252190698@estudante.ufjf.br

Axel Polleres

Vienna University of Economics and Business  
Complexity Science Hub  
Vienna, Austria  
axel.polleres@wu.ac.at

## Abstract

Maintaining the consistency of large-scale knowledge graphs (KGs) like Wikidata requires both automated methods and human expertise. In this paper, we address the task of recommending users best suited to repair a given inconsistency in a KG. Our approach leverages textual entity abstracts to compute sentence embeddings, which are clustered to identify semantically coherent regions of the KG. We introduce a framework that combines unsupervised clustering with 10-fold evaluation to test user recommendation strategies. Repair histories are linked to users, and test inconsistencies are assigned to clusters using approximate prediction. We evaluate two strategies: (i) frequency-based assignment, which recommends users based on how often they have edited entities in the predicted cluster, and (ii) embedding-based similarity, which compares the test inconsistency to past user-edited items via cosine similarity. Preliminary results show a cluster silhouette  $\geq 0.5$ , membership hit rate of 80%, with the frequency-based approach achieving a Hits@3 of 60%. Our findings suggest that lightweight unsupervised methods can effectively recommend users, showing promise for semi-automated KG maintenance.

## CCS Concepts

• **Information systems** → **Web searching and information discovery**; *Web data description languages*.

## Keywords

Knowledge Representation, Knowledge Graphs, Wikidata, Data Quality, User Recommendation, Cluster Analysis

## ACM Reference Format:

Nicolas Ferranti, Dayane Guimaraes, Jairo Francisco de Souza, and Axel Polleres. 2025. Who Can Fix This? User Recommendation for Knowledge Graph Repair via Embedding-Based Clustering. In *Knowledge Capture Conference 2025 (K-CAP '25)*, December 10–12, 2025, Dayton, OH, USA. ACM, New York, NY, USA, 4 pages. <https://doi.org/10.1145/3731443.3771363>



This work is licensed under a Creative Commons Attribution 4.0 International License. *K-CAP '25, Dayton, OH, USA*

© 2025 Copyright held by the owner/author(s).  
ACM ISBN 979-8-4007-1867-0/2025/12  
<https://doi.org/10.1145/3731443.3771363>

## 1 Introduction

Large-scale knowledge graphs (KGs) like Wikidata (WD) are central to semantic technologies, enabling tasks such as search, integration, and reasoning [3]. Yet, maintaining their *quality* remains challenging due to their scale, openness, and rapid evolution. Incomplete statements, inconsistent hierarchies, and incorrect type assignments can significantly affect downstream applications [4].

WD is a prominent community-driven KG that exemplifies this challenge: collaboratively maintained by thousands of users [5], it provides mechanisms such as *property constraints* and *Entity Schemas* to flag inconsistencies. These are complemented by external tools like WDSHEx [2], which support schema-based validation aligned with the Wikibase model. These tools detect errors but don't help identify the best users to fix them.

This paper addresses the problem of recommending users for KG repair. Given a constraint violation in WD, can we identify users likely to provide quality fixes? We approach this by clustering entities based on semantic content and analyzing user repair histories to detect patterns of topical expertise. This strategy supports WD's collaborative model by suggesting users best suited for maintaining specific parts of the graph.

Our contributions are threefold: (1) We extend an existing dataset of Wikidata constraint repairs by linking each fix to the user who performed it, enabling user-centric analysis; (2) We propose a modular framework for recommending users to repair WD inconsistencies, combining data augmentation, semantic embedding, clustering, and ranking;<sup>1</sup> (3) We empirically evaluate unsupervised strategies that use textual embeddings and semantic clustering to recommend knowledgeable users for repair tasks.

The remainder of the paper is organized as follows. Section 2 introduces the WD data model, constraints, and validation mechanisms. Section 3 describes our user recommendation framework, including dataset processing, embeddings, and clustering. Section 4 reports on the experimental setup and results, with a focus on different recommendation strategies and their effectiveness. Section 5 reviews related literature on KG repair, user modeling, and WD quality. Lastly, Section 6 concludes with a summary of findings and directions for future work.

<sup>1</sup>[https://github.com/nicolasferranti/wd\\_user\\_recommendation](https://github.com/nicolasferranti/wd_user_recommendation)

## 2 The Wikidata Data Model

WD is a collaboratively edited KG hosted on the Wikibase platform.<sup>2</sup> While WD is not natively an RDF graph, the Wikibase software provides periodic RDF dumps and a SPARQL endpoint via the WD Query Service (WDQS). These exports enable interoperability with semantic web technologies by mapping WD’s internal model to a standardized RDF representation.

At its core, WD models real-world knowledge using a statement-based data model. The primary elements are items (e.g., Q42 for Douglas Adams) and properties (e.g., P31 for *instance of*), which form statements of the form  $\langle item, property, value \rangle$  [9], where a value can be either an item or a literal. Each item is identified by a unique QID (e.g., Q42), while each property is represented by a PID (e.g., P31). These statements can be enriched with qualifiers and references through a reified representation, allowing WD to express provenance and contextual nuances [1].

**Constraint Mechanisms.** WD ensures data quality through several constraint mechanisms: (1) Property Constraints, declarative rules (e.g., cardinality, type) attached to properties via constraint statements in WD’s reification model; and (2) Entity Schemas, structural patterns defined using Shape Expressions (ShEx).

**Validation.** Property constraints can be validated through various means, including SPARQL queries [1], subtracting satisfying statements from the overall statements [8], querying the edit history [9], and through the WikibaseQualityConstraints extension<sup>3</sup>. These checks are non-enforcing, serving as guidance for editors. SPARQL has also been applied to validate domain-specific constraints [10]. While mappings to SHACL have been proposed [1], SHACL lacks full expressiveness for WD’s semantics and is not yet integrated into the core ecosystem. Entity Schemas are externally validated using tools like the ShEx Simple Validator<sup>4</sup>, but adoption remains limited compared to property constraints.

**Edits/Repairs.** Constraints fall into two categories [9]: consistency constraints, which prohibit certain triples (e.g., conflicting claims), and completeness constraints, which require them (e.g., mandatory P31). Violations are flagged in the UI and can be repaired through instance-level edits (e.g., correcting values), constraint refinements (e.g., modifying rules), or structural changes (e.g., class hierarchy updates). Each edit is logged as a revision, tied to a user or IP. Tanon et al. [9] released a dataset of historical repairs linked to revision IDs. We extend this dataset by attributing each repair to a user and enriching items with semantic descriptions, enabling our user-centered recommendation experiments.

## 3 Approach

This section presents our user recommendation framework for repairing Wikidata inconsistencies, comprising three components: *Data Augmentation*, *Semantic Mapping*, and *Recommendation and Evaluation*. First, we enrich a dataset of historical repairs with user attributions and semantic descriptions. Next, entities are mapped to a vector space and clustered into topical regions. Finally, we generate and evaluate user recommendations based on cluster membership and past repair activity.

<sup>2</sup><https://www.mediawiki.org/wiki/Wikibase>

<sup>3</sup><https://www.mediawiki.org/wiki/Extension:WikibaseQualityConstraints>

<sup>4</sup><https://shex-simple.toolforge.org/wikidata/>

## 3.1 Data Augmentation

We start from a dataset of historical repairs containing revisionID and the repaired  $\langle item, property, value \rangle$  triples. For each revision, we retrieve the corresponding username via the WD API and extract a five-sentence abstract for each QID using DBpedia sitelinks [4]. This yields triplets of the form  $(q, u, a)$ , where  $q$  is the repaired item,  $u$  the user, and  $a$  the abstract.

When a QID lacks a DBpedia abstract, we query WD via SPARQL to retrieve its label, description, and relevant triples, including property labels and the labels/descriptions of their values. We exclude uninformative properties (e.g., external identifiers, media) and URLs, as they do not contribute meaningfully to a descriptive text. The retrieved data is grouped by instance type (e.g., “human”, “film”, “organization”), and properties are ranked by frequency within each group. For items with multiple types, rankings are combined to ensure consistent property selection. We then select up to fifteen top-ranked properties per QID, balancing informativeness and generality for semantic clustering.

This structured representation is passed to the `mixtral:8x7b` language model to generate DBpedia-style abstracts. Outputs with fewer than two sentences are discarded, ensuring a minimum quality standard. This approach provides rich, coherent descriptions when DBpedia abstracts are unavailable.

## 3.2 Semantic Mapping

To represent entities semantically, we embed each textual abstract into a dense vector space using the `all-distilroberta-v1` sentence transformer model. This choice is supported by recent work [4], which indicates that sentence-level embeddings outperform traditional KG embedding models in capturing KG concept similarity.

These embeddings serve as the basis for clustering entities into semantically coherent groups. The framework supports the integration of different dimensionality reduction and clustering techniques, allowing for empirical exploration of their impact on downstream recommendation performance. In our evaluation, we experiment with PCA and UMAP for dimensionality reduction, and KMeans, Gaussian Mixture Models (GMM), Spectral Clustering, and HDBSCAN for clustering. Each resulting cluster is intended to represent a latent topical region in the KG, grouping entities that share similar semantic characteristics based on their descriptions.

To enable test-time inference, new entities can be assigned to clusters using embedding similarity or approximate nearest neighbor techniques. This step ensures that unseen entities can be incorporated into the clustering-based recommendation process.

## 3.3 Recommendation and Evaluation

Once each item is associated with a semantic cluster, we use two strategies to recommend users likely to repair a given test item.

**3.3.1 Frequency-Based Ranking.** The objective is to rank users by how frequently they have interacted with items (QIDs) in a predicted cluster. Given a predicted cluster  $C$ , let  $U_C = \{u_1, u_2, \dots, u_n\}$  be the set of users who have interacted with items in  $C$ . Let  $f(u_i, C) = \sum_{q \in C} 1_{\{(q, u_i) \in D\}}$  be the frequency count of user  $u_i$  in  $C$ , where  $1$  is the indicator function and  $D$  is the training dataset containing  $(q, u)$  interaction pairs.

We define the top- $k$  most frequent users in cluster  $C$  as:

$$\mathbf{Top}_{freq}^k(u, C) = \mathbf{argmax}_{U' \subseteq U_C, |U'|=k} \sum_{u \in U'} f(u, C)$$

For a test query  $(q^*, u^*)$ , where  $q^*$  is an unseen item assigned to cluster  $C_{q^*}$ , we evaluate the recommendation using the Hits@ $k$  metric as follows:

$$\text{Hits@k}_{freq} = \begin{cases} 1, & \text{if } u^* \in \mathbf{Top}_{freq}^k(u, C_{q^*}) \\ 0, & \text{otherwise} \end{cases}$$

**3.3.2 Cosine Similarity-Based Ranking.** Rank users based on how semantically close they are to the query item (QID), using average cosine similarity between embeddings. Let  $\mathbf{v}_{q^*} \in \mathbb{R}^d$  be the embedding of the test item  $q^*$ . For each user  $u \in U_C$ , let  $Q_u \subset C$  be the set of items the user interacted with. Let  $\mathbf{v}_q$  be the embedding of  $q \in Q_u$ . We define the average cosine similarity between  $u$  and  $q^*$  as:

$$\text{sim}(u, q^*) = \frac{1}{|Q_u|} \sum_{q \in Q_u} \cos(\mathbf{v}_{q^*}, \mathbf{v}_q)$$

The top- $k$  users most similar to  $q^*$  are defined as:

$$\mathbf{Top}_{cos.}^k(u, q^*) = \mathbf{argmax}_{U' \subseteq U, |U'|=k} \sum_{u \in U'} \text{sim}(u, q^*)$$

For a test query  $(q^*, u^*)$ , we define the evaluation metric:

$$\text{Hits@k}_{cosine} = \begin{cases} 1, & \text{if } u^* \in \mathbf{Top}_{cos.}^k(u, q^*) \\ 0, & \text{otherwise} \end{cases}$$

**3.3.3 Membership Evaluation Metric.** In addition to ranking-based evaluation, we introduce a membership-based metric to assess whether a target user appears within the predicted semantic cluster of a given entity (QID). This metric captures the fundamental intuition that, if the clustering is semantically meaningful, users who have edited a specific entity should also be present in the cluster containing that entity.

Given a test query  $(q^*, u^*)$ , let  $C$  denote the predicted cluster assignment for  $q^*$ , and let  $U_C$  be the set of users associated with cluster  $C$ . As  $u^*$  represents the user who edited  $q^*$ , we define a binary membership hit as:

$$\text{MembershipHit}(q^*) = \begin{cases} 1, & \text{if } u^* \in U_C \\ 0, & \text{otherwise} \end{cases}$$

The final Membership Hit Rate is computed as the average over all test entities:

$$\text{HitRate}_{membership} = \frac{1}{|Q_{test}|} \sum_{q \in Q_{test}} \text{MembershipHit}(q)$$

## 4 Experiments & Discussion

We base our evaluation on the dataset of one-of constraint repairs compiled by Tanon et al. [9], comprising approximately 13,700 historical repairs on WD entities. Among these, 7,720 entities have an associated DBpedia abstract, while 5,989 lack a corresponding abstract, enabling a direct comparison between text-enriched and structure-only embeddings.

Table 1 reports the top clustering configurations ranked by silhouette score. We empirically tested both PCA and UMAP for dimensionality reduction and found that UMAP consistently produced better-separated clusters, hence, it is used in all reported

experiments. Across all datasets, HDBSCAN outperformed fixed- $k$  methods (KMeans, GMM), with no improvement in silhouette even when increasing the requested number of clusters up to 40. The *noAbstracts + HDBSCAN* configuration achieved the highest silhouette score (0.58) with 69 clusters on average, reflecting well-separated topical regions of QIDs. Combining all embeddings (*all + HDBSCAN*) yielded more granular clusters (114 on average) with only a slight drop in silhouette (0.57), while improving recommendation accuracy ( $\text{Hit@1}_{freq} = 0.46$ ).

As shown in Table 1, cosine-based Hits@ $k$  were consistently lower than frequency-based ones (e.g., 0.10 vs. 0.44 for Hit@1 in the best configuration). This suggests that edit frequency within a cluster is a stronger signal of future repair activity than semantic embedding proximity, likely because many users repeatedly edit specific subsets of entities, making frequency a more reliable indicator of expertise than global similarity.

Finally, we analyzed user distribution across clusters to understand the recommendation challenge better. For each user, we measured how their historical edits were spread across clusters: if all edits were concentrated in one cluster, the user's distribution was highly focused; if edits were spread evenly across multiple clusters, the distribution was broader. We quantified this spread using entropy, a measure of uncertainty that increases when edits are evenly distributed across clusters and decreases when activity is concentrated in just one or very few clusters. A low entropy value therefore indicates a specialist user focused on a narrow topical area, while a higher value points to a generalist user contributing across multiple clusters.

In our dataset, a large proportion of users ( $\geq 70\%$ ) contributed edits to only a single cluster, and the mean entropy values remained low (0.43–0.56), confirming that most contributors are specialists. Despite this strong skew, the membership hit rates show that our clustering frequently placed the correct user in the same cluster as the test entity (up to 0.90 for fixed- $k$  methods and 0.82–0.87 for HDBSCAN). This highlights that recommending the right user is a non-trivial task: clusters often contain many potential editors with similar topical focus, making intra-cluster ranking challenging. Nevertheless, our approach often retrieves the correct user among the top candidates, showing the promise of clustering-based strategies for user recommendation in WD repair.

## 5 Related Works

This work intersects WD quality, repair, and user modeling. On the repair side, Tanon et al. [9] develop a dataset of WD constraint violations paired with their fixes, enabling analyses of historical repairs; we extend this line by linking repairs to users to support recommendation and evaluation at the user level. Xue and Zou [11] review knowledge-graph quality management across assessment, detection, and correction methods, highlighting human-in-the-loop and rule-based repair frameworks that validate system-generated repairs with users rather than recommending users themselves. This gap motivates our focus on recommending suitable users to propose repairs. For semantic representations, Ilievski et al. [4] show that sentence embeddings capture concept similarity in WD more effectively than traditional KG embeddings, motivating our

Experiment	Algo.	Sil.	# Clusters	Membership	Hit@1,3,10 <sub>freq</sub>	Hit@1,3,10 <sub>cos</sub>	Users ≤ 1	Entropy
noAbstracts	HDBSCAN	<b>0.58±0.03</b>	69.3±3.95	0.82	<b>0.44/0.66/0.77</b>	<b>0.10/0.17/0.42</b>	<b>0.76</b>	0.43
all	HDBSCAN	<b>0.57±0.04</b>	113.6±8.81	0.83	<b>0.46/0.64/0.76</b>	<b>0.07/0.20/0.39</b>	0.70	0.56
onlyAbstracts	HDBSCAN	0.55±0.05	49.8±21.64	0.87	0.41/0.56/0.72	0.06/0.18/0.31	0.72	0.46
onlyAbstracts	KMeans <sub>k=10</sub>	0.49±0.02	10.0	<b>0.90</b>	0.35/0.50/0.69	0.04/0.12/0.16	0.68	0.48
onlyAbstracts	GMM <sub>k=10</sub>	0.48±0.02	10.0	<b>0.90</b>	0.35/0.50/0.70	0.04/0.12/0.21	0.68	0.48
onlyAbstracts	KMeans <sub>k=20</sub>	0.48±0.02	20.0	0.87	0.37/0.53/0.72	0.04/0.13/0.22	0.66	0.60
onlyAbstracts	GMM <sub>k=20</sub>	0.46±0.02	20.0	0.87	0.36/0.53/0.73	0.04/0.13/0.23	0.66	0.60
onlyAbstracts	GMM <sub>k=5</sub>	0.46±0.04	5.0	<b>0.92</b>	0.32/0.49/0.65	0.04/0.12/0.20	0.74	0.27
all	KMeans <sub>k=20</sub>	0.45±0.02	20.0	0.89	0.35/0.54/0.71	0.03/0.09/0.18	0.67	0.54
noAbstracts	KMeans <sub>k=10</sub>	0.45±0.02	10.0	0.89	0.33/0.57/0.74	0.06/0.08/0.17	<b>0.76</b>	0.32

**Table 1: 10-fold results: silhouette, membership, recommendation (Hit@k), and user distribution.**

use of abstract-based sentence representations for clustering. Complementary quality frameworks detect low-quality statements via reverts, deprecation, and constraints [8], while Ferranti et al. [1] formalize WD’s property constraints with SHACL/SPARQL, emphasizing validation rather than user assignment.

On the user side, prior work characterizes roles and editing dynamics in WD without addressing user recommendation [5, 6]; studies on bots in collaborative projects underline their substantial footprint in maintenance workflows, which we treat uniformly with humans in our modeling [12]. More broadly, surveys of evolving/open KGs synthesize dimensions of evolution, observability, and collaboration (including the roles of registered users and bots), offering useful context for our clustering-based view of topical regions and editor behavior [7]. We align with this by leveraging edit histories and semantically coherent clusters for downstream recommendation, rather than analyzing evolution per se.

To our knowledge, no prior work combines semantic clustering of entities with historical repair activity to recommend users for new inconsistencies in WD, while evaluating frequency- and embedding-based strategies under 10-fold cross-validation.

## 6 Conclusions

This paper introduced a lightweight, unsupervised framework for recommending WD users likely to repair newly detected inconsistencies, based on semantic clustering of historically repaired entities. By combining sentence-level embeddings, UMAP dimensionality reduction, and clustering, we linked past repairs to the users who performed them and explored frequency- and similarity-based recommendation strategies.

Our evaluation on the dataset of one-of constraint repairs by Tanon et al. [9] demonstrated that HDBSCAN with UMAP produced the best clustering quality, achieving a silhouette score of up to 0.58, high membership hit rates (0.82–0.87), and Hits@3 of 60% using frequency-based ranking. These results indicate that topic-based clusters can meaningfully group users with relevant expertise, providing a strong basis for recommending editors for new constraint violations.

While our study focused on one-of constraint repairs, we envision extensions of this framework that address a broader range of constraint types analyzed by Tanon et al. [9], allowing for more comprehensive coverage of quality issues. Furthermore, the repaired property itself could be leveraged as an additional dimension of

user expertise, capturing the users’ “taxonomical” knowledge. A temporal dimension could also be introduced, considering recent edits, as user interests and activity patterns may evolve over time. Finally, a systematic comparison of text-based embeddings versus our fallback structural descriptions on the same dataset would shed light on which approach leads to more coherent clusters and, ultimately, more accurate recommendations.

Overall, our results suggest that embedding-based clustering provides a promising foundation for semi-automated user recommendation in knowledge graph maintenance, opening opportunities for richer user modeling and hybrid approaches that combine semantic, structural, and temporal signals for editor expertise profiling.

**Acknowledgments.** This work was funded by the Austrian Science Fund (FWF) [10.55776/COE12].

## References

- [1] N. Ferranti, J.F. De Souza, S. Ahmetaj, and A. Polleres. 2024. Formalizing and validating Wikidata’s property constraints using SHACL and SPARQL. *Semantic Web* 15, 6 (2024), 2333–2380. doi:10.3233/SW-243611
- [2] J.E. Labra Gayo. 2022. WShEx: A language to describe and validate Wikibase entities. In *3rd Wikidata Workshop 2022 @ ISWC2022, Hangzhou, China, October 2022*, Vol. 3262. CEUR-WS.org. <https://ceur-ws.org/Vol-3262/paper3.pdf>
- [3] A. Hogan, E. Blomqvist, M. Cochez, C. d’Amato, et al. 2022. Knowledge Graphs. *ACM Comput. Surv.* 54, 4 (2022), 71:1–71:37. doi:10.1145/3447772
- [4] F. Ilievski, K. Shenoy, H. Chalupsky, et al. 2024. A study of concept similarity in Wikidata. *Semantic Web* 15, 3 (2024), 877–896. doi:10.3233/SW-233520
- [5] A. Piscopo and E. Simperl. 2018. Who Models the World?: Collaborative Ontology Creation and User Roles in Wikidata. *Proc. ACM Hum. Comput. Interact.* 2, CSCW (2018), 141:1–141:18. doi:10.1145/3274410
- [6] A. Piscopo and E. Simperl. 2019. What we talk about when we talk about wikidata quality: a literature survey. In *Proceedings of the 15th International Symposium on Open Collaboration, OpenSym 2019, Skövde, Sweden, August 20–22, 2019*. ACM, 17:1–17:11. doi:10.1145/3306446.3340822
- [7] A. Polleres, R. Pernisch, A. Bonifati, D. Dell’Aglio, D. Dobriy, S. Dumbrava, L. Etcheverry, N. Ferranti, et al. 2023. How does knowledge evolve in open knowledge graphs? *TGDK* 1, 1 (2023), 11–1. doi:10.4230/TGDK.1.1.11
- [8] K. Shenoy, F. Ilievski, D. Garijo, et al. 2022. A study of the quality of Wikidata. *J. Web Semant.* 72 (2022), 100679. doi:10.1016/J.WEBSEM.2021.100679
- [9] T.P. Tanon, C. Bourgaux, and F. Suchanek. 2019. Learning how to correct a knowledge base from the edit history. In *The World Wide Web Conference*. 1465–1475.
- [10] H. Turki, D. Jemielniak, M.A.H. Taieb, J.E. Labra Gayo, et al. 2022. Using logical constraints to validate statistical information about disease outbreaks in collaborative knowledge graphs: the case of COVID-19 epidemiology in Wikidata. *PeerJ Computer Science* 8 (2022), e1085. doi:10.7717/peerj-cs.1085
- [11] B. Xue and L. Zou. 2023. Knowledge graph quality management: A comprehensive survey. *TGDK* 35, 5 (2023), 4969–4988. doi:10.1109/TKDE.2022.3150080
- [12] L. Zheng, C.M. Albano, N.M. Vora, F. Mai, and J.V. Nickerson. 2019. The Roles Bots Play in Wikipedia. *Proc. ACM Hum. Comput. Interact.* 3, CSCW (2019), 215:1–215:20. doi:10.1145/3359317