WikiConflict: A New Dataset for Conflicting Data Reconciliation in Knowledge Graph Construction

Lucas Jarnac
Orange Research, Université de Lorraine, CNRS, LORIA
Belfort, Nancy, France
lucas.jarnac@orange.com

Yoan Chabot
Orange Research
Belfort, France
yoan.chabot@orange.com

Miguel Couceiro
INESC-ID, Instituto Superior Técnico, Universidade de Lisboa
Lisbon, Portugal
miguel.couceiro@inesc-id.pt

Abstract

The construction of a knowledge graph (KG) can be performed manually. Nevertheless, ensuring minimal coverage of a KG often requires the automatic data extraction from multiple sources. However, sources and extraction algorithms often vary in quality, may provide conflicting data with different levels of specificity or even contradict each other for the same entity. To reconcile these conflicting data and integrate them consistently within the KG, numerous fusion models can be adopted that simultaneously evaluate both the quality of the sources and the data provided. However, most of these models are usually evaluated on datasets that do not specifically represent differences in specificity, the heterogeneity of data types, or the presence of long-tail entities. These three challenges are frequently encountered in KG construction, making the data fusion process more complex. In this paper, we propose to overcome these limitations by introducing WikiConflict, a dataset built from the Wikidata revision history and designed for KG construction.

CCS Concepts

• **Information systems → Information integration**; • **Computing methodologies → Knowledge representation and reasoning**.

Keywords

Knowledge graph construction, Reconciliation, Conflicting data, Uncertain data sources, Data fusion

ACM Reference Format:
Lucas Jarnac, Yoan Chabot, and Miguel Couceiro. 2025. WikiConflict: A New Dataset for Conflicting Data Reconciliation in Knowledge Graph Construction. 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/10.1145/3731443.3771371

1 Introduction

Every day, activities of organizations produce huge amounts of data through documents in both structured and unstructured formats. A knowledge graph (KG) is an ideal tool for unifying heterogeneous data efficiently. It enables multiple downstream applications such as recommendation systems, question-answering systems, and improved information retrieval [1]. A KG can be constructed either manually or automatically. Automatic or semi-automatic approaches require methods for extracting knowledge from heterogeneous sources. However, extracted knowledge is prone to errors due to the extraction methods and knowledge deltas can be observed between data sources or within the same source, leading to conflicts such as contradictions or differences in specificity [6]. Addressing these conflicts is essential for maintaining a consistent KG. To this aim, many methods deal with the fusion of data from multiple sources using probabilistic models, optimization techniques, or machine learning (ML) approaches [6]. While these methods often reach high accuracy, they are evaluated on datasets that fail to capture all the scenarios and challenges encountered when constructing large-scale and real-world KGs. These challenges include acquiring knowledge about long-tail entities, handling heterogeneous data types, and managing differences in specificity. Data scarcity in long-tail entities complicates the fusion of conflicting values. Data type heterogeneity is another important aspect to consider since a KG should contain various data types. Finally, differences in specificity encompass three levels of specificity: 1) hierarchical specificity, which can be inferred through relations such as "instance of", "subclass of", or "part of"; 2) informational specificity, where a textual description provides more detailed information than another on the same topic; and 3) specificity completeness, *i.e.,* whether all values for a given (entity, property) pair are provided. In this paper, we propose to address these limitations by leveraging the full revision history of Wikidata [3] to build WikiConflict to simulate the dynamic construction of a KG.

Our contributions are twofold: (i) we provide a new evolving dataset built from the revisions of Wikidata entities to experiment with fusion approaches in a KG construction context, (ii) we propose a testbed to experiment with fusion approaches along with preprocessing methods, evaluation metrics, and visualization tools. The remainder of this paper is organized as follows. We present the data fusion task in Section 2. We review the works that leverage the Wikidata revision history in Section 3. We describe step by step the construction of WikiConflict in Section 4, and we evaluate some fusion models on WikiConflict and datasets of the literature in Section 5. Finally, we provide perspectives and conclude in Section 6.

2 Fusion Task

From conflicting data provided by multiple data sources, the knowledge fusion task aims to derive the most consistent, accurate, and complete representation possible for each entity. These fusion models take as input a set of quadruples (s_i, e_i, p_i, v_i), where $s_i \in S$ is a data source, $e_i \in E$ is an entity, $p_i \in P$ is a property, and $v_i \in V$ is an observed value. The problem can be modeled as illustrated in

K-CAP '25, Dayton, OH, USA

ACM ISBN 979-8-4007-1867-0/25/12
https://doi.org/10.1145/3731443.3771371

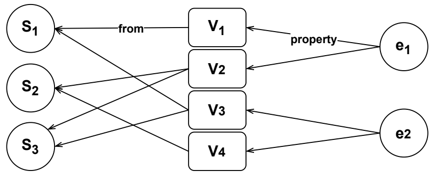

Figure 1: The graph representation of a data fusion task where the aim is to find the correct value(s) for each property and entity among different values provided by multiple sources. S_i, $i \in \llbracket 1, 3 \rrbracket$ stand for the sources; V_j, $j \in \llbracket 1, 4 \rrbracket$ stand for the values; and e_k, $k \in \{1, 2\}$ stand for the entities.

Figure 1. The same value may be provided by multiple data sources for the same property or a different property. However, each source does not necessarily provide a value for every property.

3 Related Work

Data fusion approaches are often evaluated on tabular datasets containing values provided by dozens of data sources, such as the Stock, Flight [13], or Book [15] datasets[1]. Jarnac *et al.* specify other datasets used to evaluate each fusion model presented in the survey [6]. However, these datasets do not include all possible scenarios encountered when constructing a KG. Consequently, the construction history of Wikidata provides a more realistic alternative for evaluating fusion models in such a context. Several datasets derived from Wikidata history have emerged in recent years for similar tasks. For example, in [16], the authors analyze the management of knowledge conflicts in Large Language Models (LLMs). To do this, they construct a dataset called **ConflictBank** from Wikidata containing conflicting facts generated by replacing the object of a triple with another of the same type. LLMs are then evaluated according to three aspects: the knowledge learned, the context, and the interaction between these two aspects. In [5], the authors introduce **WikiFactDiff**, a dataset constructed from Wikidata that captures its evolution between two points in time for updating factual knowledge in post-training language models. They focus on updates at the atomic fact level, where the dataset contains triples divided into three classes based on a set of manual rules: new, obsolete, and static. In [11], the authors propose a SPARQL endpoint for querying Wikidata's revision history. This endpoint enables users to retrieve the differences made after each revision based on a triple pattern. To do this, they define a data model consisting of four graphs: the global state graph, the addition graph, the deletion graph, and the default graph with revision metadata. Revisions are then indexed and stored in a database following this data model. However, the SPARQL endpoint is no longer available. In [7], the authors introduce **Wikidated 1.0**, a dataset constructed from Wikidata's edit history. This dataset captures the addition and deletion of RDF triples, enabling the modeling of Wikidata's evolution over time. It consists of a set of incremental revisions, where each revision is represented as a tuple of four elements: (i) metadata about the Wikidata entity; (ii) metadata about the revision; (iii) a set of RDF triples deleted relative to the previous state of the entity; and (iv) a set of RDF triples added relative to the previous state of the entity. Heindorf *et al.* present **Wikidata Vandalism Corpus WDVC-2015** [9], a dataset used for vandalism detection [10]. Their dataset is based on a dump of the full revision history of Wikidata from its inception (*i.e.,* October, 2012) until November, 2014, with each revision being labeled as vandalism of not. To do this, the authors use the rollback and undo/restore operations used to edit Wikidata and validate their labeling strategy by manually checking a random sample of rollback revisions, undo/restore revisions, and inconspicuous revisions.

4 Dataset Construction

In this section, we describe the WikiConflict construction steps to include the challenges faced in the construction of KG discussed in Section 1 and the data format presented in Section 2.

1) Wikidata Revisions Mechanism. In Wikidata, data are represented as a set of items and properties. Each item is identified by a QID (*e.g.,* Q243 for the *Eiffel Tower* entity), where it is associated with labels and descriptions in different languages. An item can also have properties, which are represented by a PID (*e.g.,* P31 for the *instance of* property). These properties can be used to link items together or to link an item to a literal, for example *(Eiffel Tower, instance of, Tower)* or *(Eiffel Tower, height, 324 meters)*.

A revision of an entity corresponds to the modification of at least one of its properties, including labels and descriptions. These revisions are detected by applying a difference operation between the revision at the points in time t and $t-1$, then, added to a collection that stores all observed values for each property and entity. We assume that if a source edits a property (*i.e.,* adds, modifies, or deletes a value), then the other existing values of the property have been validated and also provided by that source.

2) Data Retrieval. WikiConflict is constructed from the revision history of Wikidata entities, following a similar approach to Wikidated 1.0 [7] except that we focus on a subset of Wikidata defined by entities belonging to the same Wikipedia category (*e.g.,* Machine Learning or Monuments historiques of Paris[2]). It is important to ensure that the same contributors make multiple revisions to experiment with knowledge fusion approaches that evaluate and incorporate the quality of data sources and contributors in their modeling. Our intuition is that users who edit a Wikidata entity from a specific Wikipedia category are more likely to edit other entities within the same category (*e.g.,* a user is more likely to edit the entities *Eiffel Tower* and *Arc de Triomphe* than *Eiffel Tower* and *Java*). Each edit is transformed into an RDF triple such as ⟨*Eiffel Tower, instance of, tower*⟩.

3) Bucketization. Most data fusion approaches take the entire dataset as input and produce a one-shot evaluation of the reliability of sources and facts. However, in our case, knowledge evolves over time. For example, data provided by a source about an object when Wikidata was created (*i.e.,* in 2012) may now be obsolete, even though it was correct in 2012. A one-shot fusion step applied to the entire dataset would not only consider KG construction as a non-iterative process but also negatively impact the evaluation of source quality. This is why we propose to split the revisions into buckets in chronological order to experiment with iterative bucket-by-bucket fusion approaches and simulate the dynamic construction of a

[1] https://lunadong.com/fusiondatasets

[2] https://en.wikipedia.org/wiki/Category:Monuments_historiques_of_Paris

KG. Figure 2 illustrates the bucketization process. Each revision is represented as a tuple of five elements: the revised entity, the revised property p_i, the new value v_i, the revision timestamp, and the source that made the revision s_i. Each bucket is specific to a single entity and is built in chronological order induced by the revision timestamps.

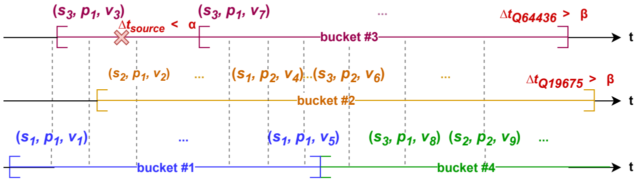

Figure 2: Bucketization of the revisions for Arc de Triomphe, Louvre Museum, and Eiffel Tower entities. By "bucket #i" we denote the i-th bucket in chronological order.

A new bucket is open if: (i) the script encounters the first revision of the entity since the inception of Wikidata, (ii) two revisions of the bucket concern the same property, are made by the same source, and the time delta between the two revisions Δt_{source} is less than the α hyperparameter, and (iii) the previous bucket has just been closed. A bucket is closed if: (i) the script encounters a new value for the same property provided by the same source, or (ii) the time delta Δt_{QID} between the opening of the bucket and the current revision exceeds the β hyperparameter. The parameter α prevents the creation of singleton buckets. Such cases may occur when a user quickly corrects their own revision, for example, to fix a typing error. While the parameter β limits the temporal window of a bucket, preventing large windows (*e.g.,* several years for only one bucket). ***4) Labeling.*** The labeled data can be represented as a set of triples (e, p, V, O), where V can be a singleton if only one correct value is identified or a set of values if the property is multivalued, and O is a set of specificity partial orders (SPO). A SPO can be represented as a tree (or multiple trees for multivalued properties), where the root is the most generic value and the leaves are the most specific values among the correct ones. Labeling is performed in two steps. Firstly, we label static properties whose values are not expected to change over time. Then, we label evolving properties by iterating over the buckets in chronological order. We also perform automatic labeling based on the time intervals associated with values in Wikidata and sources. To do this, given a set of values V for an $(entity, property)$ pair, each value $v \in V$ is considered correct if its time interval in Wikidata exceeds the following threshold:

$$interval\_threshold = \frac{\text{interval}(v)}{\max(\text{interval}(V))} \tag{1}$$

Data sources are also used as an additional signal to determine whether a triple is correct or not. If the majority of data sources that provide the same triple are identified by IP addresses, the triple is considered incorrect. This assumption is based on the observation that IP addresses are often associated with sources that vandalize entities. Evaluated on 40 manually labeled entities (see Table 1), this automatic approach achieves 96.71% precision and 76.71% recall. We noticed that taking IP addresses into account increased the performance.

Table 1: Statistics of the datasets without preprocessing.

	# entities		# sources		# properties		# triples		# incorrect triples
GT	w/	w/o	w/	w/o	w/	w/o	w/	w/o	w/
Flight	100	2,878	38	38	6	6	106,580	1,354,889	90,303
Stock	100	1,000	55	55	16	16	394,687	3,854,889	367,917
Book	100	1,265	237	895	2	2	965	22,871	965
WikiConflict	40	6,482	459	5,965	143	597	1,330	172,120	381

5 Experiments

In this section, we evaluate several well-known fusion models on WikiConflict dataset and other datasets commonly used.
1) Models. We compare the performance of the TruthFinder [15], CRH [8], SLiMFast [12], and LTM [2]. TruthFinder [15] is an iterative model that uses the notion of implication between different facts to update the confidence score of the facts. CRH [8] is an optimization-based model that addresses data heterogeneity by applying various distance functions to measure the distance of values from the estimated true value. SLiMFast [12] is a statistical inference model that incorporates domain features (*e.g.,* the number of citations of a scientific paper) as additional weights for estimating the quality of a data source. LTM [2] is a probabilistic model that considers multivalued properties. We selected these four popular models because they rely on different methodologies and the aim of these experiments is to provide a concise overview of performance differences across datasets. To experiment with different fusion approaches, we have developed a testbed. WikiConflict and the testbed are available at https://github.com/Orange-OpenSource/trustfuse.
2) Datasets. Concerning the datasets from the literature, we use two datasets available at https://lunadong.com/fusiondatasets, which are widely used to evaluate fusion models. In particular, we use the **Flight** [13] and **Book** [14] datasets. For fair comparison with these datasets, we use a subset of WikiConflict consisting of representative entities: *Eiffel Tower*, *Arc de Triomphe*, and *Napoleon*, which exhibit similar conflict rates to the benchmark datasets. Table 1 presents statistics without preprocessing for these datasets in the literature, including Stock [13]. Before fusion, a preprocessing step is performed by removing extra spaces for string or identifying authors in strings that contain more than one for the Book dataset. For each dataset, we distinguish two parts: one with labeled triples (w/ GT) and a second part where triples are either unlabeled or automatically labeled (w/o GT). While our dataset contains fewer triples labeled as correct or incorrect, it has more distinct sources and properties compared to existing datasets. ***3) Evaluation Metrics.*** To evaluate model performance, we use the following metrics: precision, accuracy, recall, and F1-score. For multivalued properties (*i.e.,* a property that can have multiple values) that have a partial order of specificity, we employ a metric usually used for the semantic annotation of tabular data called Average Hierarchical (AH) score [4], which we have adapted to our problem. In our case, the AH score is computed using the following equation:

$$AH = \frac{\sum_{v \in V} \frac{depth(v,sp)}{depth_{max}(sp)}}{|V|},\ sp \in O \tag{2}$$

where O is a SPO and V is the set of values returned by the fusion model. The specificity of a value for an (entity, property) pair that admits a SPO (O) corresponds to its depth in the partial order divided

Table 2: Data fusion results on datasets Flight, Book, and WikiConflict. P stands for precision, R stands for recall, F1 stands for F1-score, and ACC stand for accuracy. The best results are in bold and we underline the second best result.

Model	Dataset	P	R	F1	ACC	AH
CRH	Flight	**65.53**	**69.87**	**67.63**	**89.73**	_
	Book	62.00	35.43	45.09	87.54	_
	WikiConflict	59.83	49.50	54.18	77.07	34.15
TruthFinder	Flight	**65.07**	**69.38**	**67.16**	**89.58**	_
	Book	32.00	18.29	23.27	82.59	_
	WikiConflict	59.48	49.22	53.86	76.91	23.08
SLiMFast	Flight	**77.25**	**82.37**	**79.73**	**93.57**	_
	Book	9.00	5.14	6.55	78.80	_
	WikiConflict	44.66	36.95	40.44	70.20	31.67
LTM	Flight	**55.51**	**94.51**	**69.94**	**87.52**	_
	Book	38.05	93.71	54.13	77.06	_
	WikiConflict	53.03	87.45	66.02	75.35	13.41

by the maximum depth of this order. This definition ensures that the AH score for an (entity, property) pair is always between 0 and 1. For example, if the value "Paris" is provided for a given property with the following SPO:

$$\text{Europe} \leftarrow \text{France} \leftarrow \text{Ile-de-France} \leftarrow \text{Paris} \leftarrow 7^{\text{th}} \text{ arr. of Paris}$$

where "$x \leftarrow y$" means "y more specific than x", since the value of $depth(Paris, sp) = 3$ and $depth_{max}(sp) = 4$, the AH score for this single value is 0.75.

4) Results. The results are presented in Table 2. The table highlights significant differences in fusion model performance between the two literature datasets and the WikiConflict dataset. In particular, the models achieve the best performance on the Flight dataset, while the lowest results are observed on the Book dataset. For the Flight dataset, preprocessing involves converting dates to minutes and removing extra spaces for gates. Conversely, for the Book dataset a preprocessing is required to extract the first names and last names of authors, which are expressed in a wide variety of formats, requiring more complex preprocessing than for the Flight dataset. In contrast, the WikiConflict dataset does not require any preprocessing steps except for numerical data, where unit scaling may be necessary. The average number of triples per entity shows a significant difference between existing dataset and WikiConflict, which contains fewer values for an (entity, property) pair, reflecting the long-tail entity phenomenon of the dataset. This phenomenon may partly explain the better results on Flight, as fewer triples are provided for an given entity in WikiConflict, making it more challenging to identify patterns among the conflicting values. Finally, the results show that the models do not achieve high AH scores, highlighting the need for the development of fusion models that would provide the most specific values.

6 Conclusion and Perspectives

In this paper, we introduced WikiConflict, a new dataset constructed from the revision history of Wikidata entities to address the task of conflicting data fusion, particularly in the context of KG construction. The experiments on existing datasets and a subset of WikiConflict show significant performance differences. Therefore, we advocate for the development of new fusion models. Ideally, these models should adopt a holistic approach, considering all aspects of knowledge, including long-tail entities, data heterogeneity, and differences in data specificity. Value specificity is an essential aspect when building a domain-specific KG, ensuring the highest possible level of specificity in data is crucial for applications such as recommendation or question-answering systems, to provide precise answers. In future work, we aim to include qualifiers into the dataset and an automatic labeling process to get specificity partial orders for the entire dataset and continue manual labeling efforts.

References

[1] Aidan Hogan et al. 2021. *Knowledge Graphs.* Number 22 in Synthesis Lectures on Data, Semantics, and Knowledge. Springer.

[2] Bo Zhao et al. 2012. A Bayesian Approach to Discovering Truth from Conflicting Sources for Data Integration. *Proc. VLDB Endow.* 5, 6 (2012), 550–561.

[3] Denny Vrandecic et al. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM* 57, 10 (2014), 78–85.

[4] Ernesto Jiménez-Ruiz et al. 2020. SemTab 2019: Resources to Benchmark Tabular Data to Knowledge Graph Matching Systems. In *The Semantic Web - 17th International Conference, ESWC 2020, Heraklion, Crete, Greece, May 31-June 4, 2020, Proceedings (Lecture Notes in Computer Science, Vol. 12123)*, Andreas Harth, Sabrina Kirrane, Axel-Cyrille Ngonga Ngomo, Heiko Paulheim, Anisa Rula, Anna Lisa Gentile, Peter Haase, and Michael Cochez (Eds.). Springer, 514–530.

[5] Hichem Ammar Khodja et al. 2024. WikiFactDiff: A Large, Realistic, and Temporally Adaptable Dataset for Atomic Factual Knowledge Update in Causal Language Models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation, LREC/COLING 2024, 20-25 May, 2024, Torino, Italy*, Nicoletta Calzolari, Min-Yen Kan, Véronique Hoste, Alessandro Lenci, Sakriani Sakti, and Nianwen Xue (Eds.). ELRA and ICCL, 17614–17624.

[6] Lucas Jarnac et al. 2025. Uncertainty Management in the Construction of Knowledge Graphs: A Survey. *TGDK* 3, 1 (2025), 3:1–3:48. doi:10.4230/TGDK.3.1.3

[7] Lukas Schmelzeisen et al. 2021. Wikidated 1.0: An Evolving Knowledge Graph Dataset of Wikidata's Revision History. In *Proceedings of the 2nd Wikidata Workshop (Wikidata 2021) co-located with the 20th International Semantic Web Conference (ISWC 2021), Virtual Conference, October 24, 2021 (CEUR Workshop Proceedings, Vol. 2982)*, Lucie-Aimée Kaffee, Simon Razniewski, and Aidan Hogan (Eds.). CEUR-WS.org.

[8] Qi Li et al. 2014. Resolving conflicts in heterogeneous data by truth discovery and source reliability estimation. In *International Conference on Management of Data, SIGMOD 2014, Snowbird, UT, USA, June 22-27, 2014.* ACM, 1187–1198.

[9] Stefan Heindorf et al. 2015. Towards Vandalism Detection in Knowledge Bases: Corpus Construction and Analysis. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, August 9-13, 2015*, Ricardo Baeza-Yates, Mounia Lalmas, Alistair Moffat, and Berthier A. Ribeiro-Neto (Eds.). ACM, 831–834.

[10] Stefan Heindorf et al. 2016. Vandalism Detection in Wikidata. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management, CIKM 2016, Indianapolis, IN, USA, October 24-28, 2016*, Snehasis Mukhopadhyay, ChengXiang Zhai, Elisa Bertino, Fabio Crestani, Javed Mostafa, Jie Tang, Luo Si, Xiaofang Zhou, Yi Chang, Yunyao Li, and Parikshit Sondhi (Eds.). ACM, 327–336.

[11] Thomas Pellissier Tanon et al. 2019. Querying the Edit History of Wikidata. In *The Semantic Web: ESWC 2019 Satellite Events - ESWC 2019 Satellite Events, Portorož, Slovenia, June 2-6, 2019, Revised Selected Papers (Lecture Notes in Computer Science, Vol. 11762)*, Pascal Hitzler, Sabrina Kirrane, Olaf Hartig, Victor de Boer, Maria-Esther Vidal, Maria Maleshkova, Stefan Schlobach, Karl Hammar, Nelia Lasierra, Steffen Stadtmüller, Katja Hose, and Ruben Verborgh (Eds.). Springer, 161–166.

[12] Theodoros Rekatsinas et al. 2017. SLiMFast: Guaranteed Results for Data Fusion and Source Reliability. In *Proceedings of the 2017 ACM International Conference on Management of Data, SIGMOD Conference 2017, Chicago, IL, USA, May 14-19, 2017.* ACM, 1399–1414.

[13] Xian Li et al. 2012. Truth Finding on the Deep Web: Is the Problem Solved? *Proc. VLDB Endow.* 6, 2 (2012), 97–108.

[14] Xin Luna Dong et al. 2009. Integrating Conflicting Data: The Role of Source Dependence. *Proc. VLDB Endow.* 2, 1 (2009), 550–561.

[15] Xiaoxin Yin et al. 2007. Truth discovery with multiple conflicting information providers on the web. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Jose, California, USA, August 12-15, 2007.* ACM, 1048–1052.

[16] Zhaochen Su et al. 2024. ConflictBank: A Benchmark for Evaluating the Influence of Knowledge Conflicts in LLM. *CoRR* abs/2408.12076 (2024). arXiv:2408.12076