Check for updates

TrustFuse: A Fusion Testbed for Uncertain Knowledge Reconciliation

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

To build a knowledge graph, knowledge can be extracted from multiple data sources. However, for a given topic, multiple data sources rarely provide a unified view of the data. The data may differ in unit scales, levels of specificity, or even be contradictory. To jointly find the most trustworthy data and evaluate the reliability of the sources, data fusion approaches are usually applied. Although existing tools implement such approaches, they often lack essential functionalities such as a template for developing data fusion approaches, evaluation metrics, or a user-friendly visualization of the fused results. To overcome these limitations, we introduce TrustFuse, a comprehensive testbed that supports experimentation with fusion models, their evaluation, and the visualization of datasets as graphs or tables within a unified user interface.

CCS Concepts

• **Information systems** → **Information integration**; *Graph-based database models.*

Keywords

Conflicting data, Data fusion, Uncertain data sources

ACM Reference Format:
Lucas Jarnac, Yoan Chabot, and Miguel Couceiro. 2025. TrustFuse: A Fusion Testbed for Uncertain Knowledge Reconciliation. 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/10.1145/3731443.3771372

1 Introduction

When searching for information on a specific topic from different sources on the web, we rarely find identical information. Different data sources may use different terms to describe the same object. Additionally, false information can be spread across sources due to various factors such as outdated or erroneous data, fake news, or vandalism by malicious persons [10]. An example of such conflicting data for the *Country* and *Instance of* properties on the Eiffel tower is illustrated in Table 1, where the values come from various revisions of the entity over time in Wikidata. A common approach to automatically finding the truth among uncertain data is the use of data fusion (truth discovery) models. These models automatically find the most reliable values among uncertain data by jointly estimating both the most confident values and the quality of the sources, each influencing the other. Numerous models have been developed for evaluating crowdsourced data or integrating reliable data into a database, including probabilistic, iterative, optimization, and machine learning models. For example, [15] detects the interdependence between data sources that affects the computation of confidence scores associated with the claims made by these sources in the fusion algorithm. [2] deals only with numerical data by using Gaussian distributions. [11] uses various source-related features such as the year of publication or the number of citations for a scientific article. Record Fusion [11] adopts a three-level view (cell, tuple, dataset) in the case of relational data fusion, assuming that these views capture correlations between different data and improve fusion performance with their ML-based model. [5] leverages the prior knowledge of the KG in construction to guide the decisions of the fusion algorithm. These models are presented in more detail in the following surveys [6, 17]. Some papers introducing such methods provide GitHub repositories or applications, but these lack useful features when dealing with knowledge graphs or are no longer maintained. In this paper, we introduce a testbed for experiments with these types of fusion models in the knowledge graph (KG) construction scenario along with a dataset constructed for this setting. This testbed can be used either through the command line or via a user interface (UI). We made the testbed available on GitHub[1]. The remainder of this paper is organized as follows: in Section 2, we define the fusion task, which forms the basis of our tool. In Section 3, we review existing tools designed to experiment with fusion models. Our testbed and its main features are described in Section 4, followed by the presentation of the datasets and models included in TrustFuse in Section 5. Finally, we provide perspectives and a conclusion in Section 6.

2 Data Fusion Task

The aim of the fusion task in KG construction is to combine the different information about an entity provided by multiple data sources into a single consistent representation. Typically, the input to a fusion model consists of a set of four-element tuples:

$$(e, p, v, s) \in O \times P \times V \times S$$

where O is the set of objects (*e.g.,* Eiffel Tower $\in O$), P is the set of properties (*e.g.,* Country $\in P$), V is the set of values that contains literals (*e.g.,* strings, integers, floats, etc.) and objects *i.e.,* $O \subset V$, and S is the set of sources. Each fusion model then estimates the reliability of each triple (e, p, v) while jointly assessing the quality of the data sources, typically assigning them a weight w_s representing

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 979-8-4007-1867-0/25/12
https://doi.org/10.1145/3731443.3771372

[1] https://github.com/Orange-OpenSource/trustfuse

Table 1: Example of conflicting data about the Eiffel Tower (https://www.wikidata.org/wiki/Q243), where multiple values can be attributed to a property (*e.g.,* Instance of).

Input			
Entity	Source	Country	Instance of
Eiffel Tower	S_1	France	{attraction, observation tower, lattice tower}
Eiffel Tower	S_2	Russia	{tower}
Eiffel Tower	S_3	China	{concert tour, architectural landmark}
Eiffel Tower	S_4	France	{tourist destination, landmark, tower}
Results after the fusion step			
Entity		Country	Instance of
Eiffel Tower		France (S_1, S_4)	{observation tower (S_1), lattice tower (S_1), architectural landmark (S_3), tourist destination (S_4)}

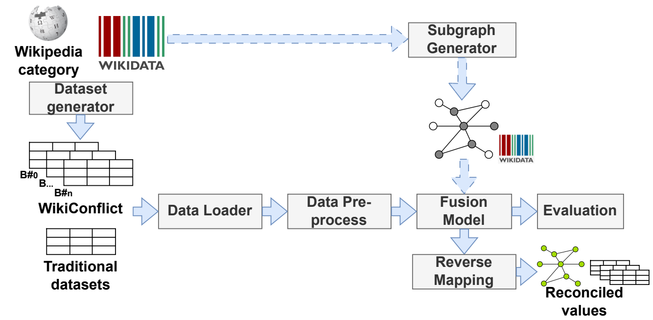

Figure 1: Architecture of TrustFuse.

the confidence in the source. Data fusion models can be categorized into different classes [17]. Some models formulate the task as an optimization problem, where the weight w_s and the confidence score in the triples are iteratively updated until convergence. Probability-based models, often represented as probabilistic graphical models estimate their parameters using techniques such as Expectation Maximization. More recently, machine learning approaches have emerged, offering a wide range of modeling possibilities for fusion.

For downstream applications such as recommendation or question answering engines, it is important to construct a KG with the highest possible level of specificity. Specificity levels are defined by a hierarchy of values. Some data fusion models, such as TDH [13] and ASUMS [12] incorporate this notion of specificity. In our tool, we define this hierarchy through partial orders represented by non-binary trees. For instance, a specificity partial order for the object-property pair *(Eiffel Tower, located in the administrative territorial entity)* could be:

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

where "→" means "more specific than".

3 Related Work

To the best of our knowledge, there is no testbed for experimenting with knowledge fusion with the diversity and complexity imposed by the fusion models. In [3], the authors introduce a fusion system implemented in Java and called AllegatorTrack. It enables users to test, compare, and combine 12 different fusion models and obtain explanations of the results. The back-end of the system includes several components upstream of the fusion task, such as an information extraction module and a data preprocessing module. It also includes a front-end to display the results with the confidence scores attributed to data sources and claims. In [7], Wan *et al.* propose a fusion approach called KDEm, which leverages kernel density estimation with a Gaussian kernel. In their paper, a GitHub repository[2] is provided, which contains different fusion models with evaluation methods and some useful functions. However, both repositories lack documentation on usage and functionality, making the implementation of new methods challenging. In [18], Singleton highlights the limitations of other existing data fusion repositories such as Spectrum[3] and Dafna-ea[4]. The author argues that most repositories provide some pre-built models and evaluation functions with a new method introduced in the corresponding paper, but do not provide a generic framework to experiment with a wider variety of fusion methods. To address this issue, the author provides *truthdiscovery*[5], a Python library that includes tools such as a synthetic data generator, a visualization tool, and some state-of-the-art models. However, their visualization in GIF format does not allow for efficient visualization of large datasets. Furthermore, their modeling of the fusion task requires a specific input data format as a set of triples $(source, attribute, value)$ for a given entity. However, a KG can be seen as a set of triples $\{(entity, predicate, object)\}$ with multiple entities and this specific input data format prevents the fusion of a KG in a single operation. The following section presents the key features of TrustFuse, designed to overcome these limitations by allowing fusion in a KG construction setting.

4 Key Features

Implementing a generic testbed represents a complex challenge. Existing fusion models exhibit different features and the datasets are heterogeneous. Furthermore, each model may process datasets differently, applying different preprocessing steps or considering different data types for the same properties. Following these constraints imposed by the models and the datasets, the TrustFuse architecture is designed as illustrated in Figure 1.

TrustFuse is a Python application that can be used either on the command line or via a UI built with Gradio and Pyvis. It needs two input files: a mapping file that assigns a data type to each property and a configuration file that lists the preprocessing functions to be applied to the dataset. TrustFuse consists of seven components.

Dataset Generator. This component generates a dynamic dataset from a Wikipedia category and Wikidata [4]. The generated dataset is used to evaluate fusion approaches in a realistic KG construction scenario. It includes a Wikidata retriever, a bucketization script that splits the data into buckets in chronological order, and a labeling script for both silver standard (*i.e.,* automatic labeling) and gold standard (*i.e.,* manual labeling). We also provide a labeling user interface for the constructed dataset.

Data Loader. Loads both the datasets from the literature and the

[2]https://github.com/MengtingWan/KDEm

[3]https://github.com/totucuong/spectrum

[4]https://github.com/daqcri/DAFNA-EA

[5]https://truthdiscovery.readthedocs.io/en/latest/

constructed dataset and prepares the input expected by the fusion module. We represent a dataset as a Pandas DataFrame to facilitate processing and indexing by property. This data structure also enables efficient application of mappings between data before and after preprocessing.

Data Preprocessor. Applies the preprocessing requested in the configuration file. For example, different preprocessing functions are available such as: *scale_unit* to scale the units of values, *transform_date* to format the dates, or *extract_integer* to extract a number from a string. A preprocessing function can be applied by selecting one or more specific properties, to one or more data types, or to the entire dataset.

For additional details on one of the preprocessing options available, consult the TrustFuse documentation.

Subgraph Generator. This module generates Wikidata subgraphs on demand to perform fusion using a KG as prior knowledge. These subgraphs are generated in relation to the selected Wikipedia categories with a chosen depth and rate of entities randomly selected from Wikidata, simulating the enrichment of a KG.

Fusion Model. Applies data fusion according to the selected model. A list of well-known models already implemented is available (see Section 5), and the documentation provides the necessary guidance for implementing a new model by following a provided template.

Evaluator. This component evaluates fusion models on various aspects: the specificity (calculated using partial orders if available), the completeness, and the performances usually evaluated, such as the F1 score or accuracy. It also provides metrics to assess the quality of the fusion of numerical data such as the Mean Normalized Absolute Distance [9] (MNAD).

Visualization. It provides a visualization of the result in a table or graph form constructed with Pyvis as depicted in Figure 2. The graph visualization includes filters to highlight specific properties (*i.e.,* edges) or entities (*i.e.,* nodes) if required. TrustFuse also provides a visualization of partial orders among values for each property if such orders exist in the dataset.

Its UI is divided into two parts: the left side allows for dataset operations such as loading, preprocessing, and data fusion, while the right side displays the dataset and the results of the fusion as shown in Figure 2. To perform the fusion step, we need complete three steps: *1)* dataset loading, *2)* preprocessing the dataset, *3)* selecting the fusion model. Once the dataset is loaded we can navigate between the buckets of the dataset (dataset subsets). TrustFuse supports dynamic navigation between the dataset buckets, *i.e.,* the fusion model provides results progressively, allowing users to observe intermediate buckets without waiting for the full dataset to be processed. To perform the preprocessing step, we can either load a file containing the preprocessing functions or we can select them directly in the UI, where different preprocessing functions are proposed. Finally, we can choose a fusion model and modify its parameters directly in the UI if needed.

5 Fusion Datasets and Models

TrustFuse already includes existing datasets as well as well-known data fusion models from the literature. Regarding datasets, TrustFuse integrates Flight [14], Stock [14], Book [16], and WikiConflict. **Flight** is a dataset that contains data about 1,200 flights on six properties: Scheduled departure, Actual departure, Departure gate, Scheduled arrival, Actual arrival, and Arrival gate A subset of 100 flights are labeled. **Stock** contains data for 1,000 stocks collected from 55 data sources. Each stock has properties such as Symbol, Volume, Last trading price, Dividend and 12 additional properties. As with the Flight dataset, 100 stocks are labeled. **Book** contains titles and authors for 1,265 books identified by their ISBNs collected from 895 data sources. Similarly, 100 books have been labeled. **WikiConflict** is the name of our dataset constructed from a Wikipedia category and Wikidata history by the integrated dataset generation tools in TrustFuse. WikiConflict contains data on 143 properties for 40 entities. The entities have been manually labeled, we also provide a standard silver version with data labeled using manually defined rules. Regarding the models, TrustFuse currently implements the following:

ACCU [15] is a probabilistic model that incorporates the interdependence between data sources. The model uses Bayesian analysis to identify dependent sources: if two sources share many identical incorrect values, then they are likely to be dependent, while sharing correct values is less indicative of dependency.

CATD [8] is a model adapted to long-tail entities (*i.e.,* when limited information is extracted about an entity), which complicates the fusion process. It models source errors using a Gaussian distribution with mean zero and variance representing source reliability. To cope with limited data from sources, CATD leverages the upper bound of the confidence interval of the variance rather than a sample variance alone.

CRH [9] addresses fusion on heterogeneous values by jointly estimating source and triples reliability across all value types simultaneously. CRH formulates the fusion task as an optimization problem with the objective of minimizing the deviation between truths and source observations.

GTM [2] is a probabilistic graphical model that focus on numerical data by leveraging the relative distance of claimed values from the truth. The truth for each entity is modeled as a random variable, which is used as the mean parameter of a Gaussian distribution where the quality of data sources is modeled by the variance of the distribution.

KDEm [7] uses kernel density estimation with Gaussian kernels, extending it by incorporating source reliability weights. To identify trustworthy values, KDEm analyzes the resulting probability distribution where high density regions indicate trustworthy values. This approach allows multiple correct values for multivalued properties.

LTM [1] models each data source as a classifier defined by its sensitivity and its specificity. These quality dimensions and truth prior probabilities are modeled using Beta distributions with hyperparameters representing prior counts, while truth values follow Bernoulli distributions. LTM uses Collapsed Gibbs Sampling to infer truths and source quality.

SLiMFast [11] considers the fusion as a statistical learning problem. It incorporates domain-specific features of the sources into the logistic function parameters to estimate their reliability in the fusion process (*e.g.,* the number of citations and the year of publication for scientific papers). SLiMFast uses either the Expectation Maximization algorithm when no labeled data are available or Empirical Risk Minimization when labeled data are provided.

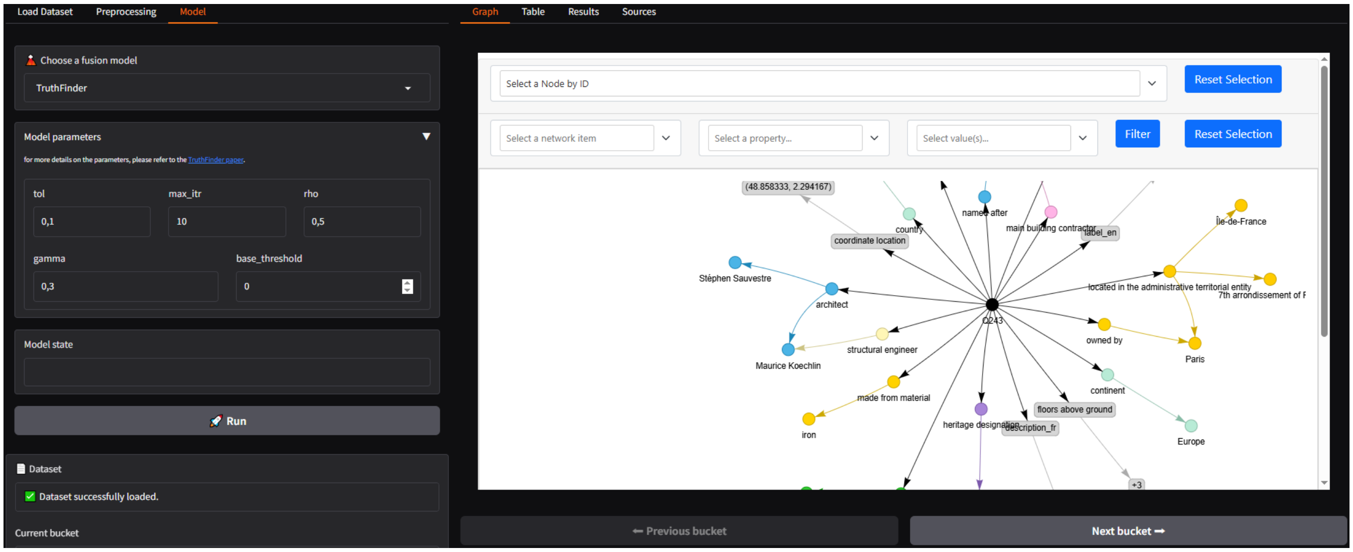

Figure 2: TrustFuse UI. Three steps are needed before the fusion: dataset loading, preprocessing, and selecting the fusion model.

TruthFinder [16] is an iterative method that measures the influence of each triple on other triples and includes the interdependence of the sources through a dampening factor.

6 Conclusion and Future Work

We have introduced TrustFuse, a testbed for developing and experimenting with fusion models in the context of KG construction with multiple conflicting data sources. TrustFuse provides ready-to-use tools such as dataset generator, preprocessing functions, dataset visualization, or evaluation functions. In future work, we aim to enhance the fluidity and interactivity of the UI, particularly for graph visualization, which remains challenging with large datasets. We have implemented this testbed to facilitate the comparison of fusion approaches, but also to advocate the development of holistic fusion models to handle the different scenarios we face when constructing a KG from multiple heterogeneous data sources such as multi-valued properties, differences in specificity, and long-tail entities fusion.

References

[1] 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. doi:10.14778/2168651.2168656

[2] Bo Zhao et al. 2012. A probabilistic model for estimating real-valued truth from conflicting sources. *Proc. of QDB* 1817 (2012).

[3] Dalia Attia Waguih et al. 2015. AllegatorTrack: Combining and reporting results of truth discovery from multi-source data. In *31st IEEE International Conference on Data Engineering, ICDE 2015, Seoul, South Korea, April 13-17, 2015*, Johannes Gehrke, Wolfgang Lehner, Kyuseok Shim, Sang Kyun Cha, and Guy M. Lohman (Eds.). IEEE Computer Society, 1440–1443.

[4] Denny Vrandecic et al. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM* 57, 10 (2014), 78–85. doi:10.1145/2629489

[5] Jiacheng Huang et al. 2022. Trustworthy Knowledge Graph Completion Based on Multi-sourced Noisy Data. In *WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022.* ACM, 956–965. doi:10.1145/3485447.3511938

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

[7] Mengting Wan et al. 2016. From Truth Discovery to Trustworthy Opinion Discovery: An Uncertainty-Aware Quantitative Modeling Approach. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, Balaji Krishnapuram, Mohak Shah, Alexander J. Smola, Charu C. Aggarwal, Dou Shen, and Rajeev Rastogi (Eds.). ACM, 1885–1894.

[8] Qi Li et al. 2014. A Confidence-Aware Approach for Truth Discovery on Long-Tail Data. *Proc. VLDB Endow.* 8, 4 (2014), 425–436.

[9] 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.

[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] 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. doi:10.1145/3035918.3035951

[12] Valentina Beretta et al. 2016. How Can Ontologies Give You Clue for Truth-Discovery? An Exploratory Study. In *Proceedings of the 6th International Conference on Web Intelligence, Mining and Semantics, WIMS 2016, Nîmes, France, June 13-15, 2016.* ACM, 15:1–15:12.

[13] Woohwan Jung et al. 2019. Crowdsourced Truth Discovery in the Presence of Hierarchies for Knowledge Fusion. In *Advances in Database Technology - 22nd International Conference on Extending Database Technology, EDBT 2019, Lisbon, Portugal, March 26-29, 2019.* OpenProceedings.org, 205–216.

[14] Xian Li et al. 2012. Truth Finding on the Deep Web: Is the Problem Solved? *Proc. VLDB Endow.* 6, 2 (2012), 97–108. doi:10.14778/2535568.2448943

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

[16] 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.

[17] Yaliang Li et al. 2015. A Survey on Truth Discovery. *SIGKDD Explor.* 17, 2 (2015), 1–16.

[18] Joe Singleton. 2019. Implementation and Analysis of Truth Discovery Algorithms. (2019).