# Detecting Knowledge Inconsistencies Across Text, Tables, and Knowledge Graphs

Fanfu Wei[1 [0009−0008−4397−0492]] ⋆, Thibault Ehrhart[1 [0000−0003−1377−8279]], and Raphaël Troncy[1 [0000−0003−0457−1436]]

EURECOM, France
{fanfu.wei, thibault.ehrhart, raphael.troncy}@eurecom.fr

**Abstract.** Wikipedia and Wikidata are widely used for information access, LLM pre-training, and retrieval-augmented generation. Their knowledge is deeply connected but scattered across text, tables, and knowledge graphs. This raises a practical question: when these modalities disagree, how can we detect and explain the conflict? We study this problem as *modality-level inconsistency detection*. We first introduce a taxonomy of cross-modal knowledge inconsistencies, covering information granularity differences, direct conflicts, temporal changes, and KG incompleteness. We then present KONTRAST, an automatic framework that uses Text-to-SPARQL and LLM reasoning to compare table-based answers with KG evidence and categorize the resulting inconsistencies. Experiments on various Table-QA datasets show that cross-modal inconsistencies are common and informative. They reveal not only true knowledge conflicts, but also missing KG structure and temporal mismatches while being limited by Text-to-SPARQL errors and noise. Our analysis shows that text, tables, and KGs can complement and correct one another through systematic comparison. KONTRAST provides a practical tool for large-scale knowledge auditing and establishes a benchmark for future work on cross-modal knowledge consistency. Code and data are available at `https://github.com/ECLADATTA/KONTRAST`.

**Keywords:** Knowledge Graphs · Knowledge Inconsistencies · Text-to-SPARQL

## 1 Introduction

Wikipedia and Wikidata are two of the most widely used open knowledge resources on the Web. They support human information access and serve as important knowledge sources for open-domain question answering [17], retrieval-augmented generation [19], large language model pre-training [3,12], and other knowledge-intensive applications. Their value comes from scale, openness, and constant updates. These same properties, however, make consistency difficult to maintain.

⋆ Corresponding author.

This difficulty is amplified by the way knowledge is represented. The same fact may appear in a sentence, an infobox, or a table row of a Wikipedia article, and in a Wikidata statement. These representations are connected, but they are edited under different schemas, workflows, and update cycles. As a result, the same question can receive different answers depending on whether we consult text, tables, or the KG. A Wikipedia table may list the complete casting of a movie while Wikidata contains only the lead actors; Wikidata may store country-specific release dates for the movie while a table reports only a single year; an elevation value may differ between an infobox and Wikidata because the sources refer to different measurement points. Such cases are not merely retrieval errors, they reveal *modality-level knowledge inconsistencies*.

Recent work has begun to study inconsistencies within Wikipedia. For example, WikiContradict constructs a benchmark of conflicting textual claims and evaluates whether LLMs can recognize incompatible answers [14]. WikiCollide/-CLAIRE detects corpus-level inconsistencies by surfacing conflicting Wikipedia passages for human review [29]. These studies show that inconsistencies are common and that automatic systems can help finding them. However, they mostly focus on textual evidence which leaves an important gap: how to systematically detect and categorize inconsistencies across text, tables, and KGs.

To address this gap, we need a setting where the same information needs can be grounded in multiple modalities. Table-based question answering provides such a hinge point. Its questions are grounded in Wikipedia tables or tables and table-relevant text, while recent Text-to-SPARQL models allow the same questions to be queried against Wikidata. This creates a direct comparison: the table-based answer represents Wikipedia table-text evidence, and the SPARQL result retrieves KG evidence. The consistent agreement provides mutual support. Inconsistent disagreement provides a signal for possible knowledge conflicts, missing KG structure, or temporal drift.

In this paper, we introduce the task of *modality-level inconsistency detection*: given a table-based question, its Wikipedia-grounded answer, and a KG answer obtained through Text-to-SPARQL querying, the goal is to determine whether the modalities agree and to characterize the mismatch when they do not.

We introduce KONTRAST (**K**nowledge **Contrast**), an automatic system for detecting and categorizing inconsistencies across text, tables, and KGs. Given a table-based QA instance, KONTRAST retrieves Wikidata KG evidence with Text-to-SPARQL and compares it with the Wikipedia-table-grounded answer using rule-based matching and LLM-based reasoning. Its central idea is to treat cross-modal disagreement not as noise, but as a signal for knowledge reconciliation. Figure 1 depicts the overview of our system. Our contributions are:

- We introduce modality-level inconsistency detection across text, tables, and KGs.
- We propose a taxonomy of cross-modal knowledge inconsistencies, covering direct conflicts, granularity differences, temporal changes, and KG incompleteness.

- We present KONTRAST, a system for detecting and categorizing inconsistencies across text, tables, and KGs.
- We generate a new table-based QA dataset suitable for evaluating KONTRAST and we show that cross-modal inconsistencies are measurable at scale and can reveal where modalities complement or correct each other.

## 2 Preliminaries and Related Work

### 2.1 Knowledge Conflict

Knowledge conflict can be divided into three main directions: (i) *Uncertainty in KG construction* studies uncertainty arising from extracting, aligning, and fusing heterogeneous data into a KG. These discrepancies, termed knowledge deltas, reflect the semantic nature of knowledge conflicts such as invalidity, ambiguity, vagueness, fuzziness, timeliness, and incompleteness [15,9]. In parallel, WikiConflict offers a Wikidata revision-based dataset for conflicting data reconciliation in KG construction [16]. (ii) *Corpus-level inconsistency* studies contradictions inside large knowledge corpora such as Wikipedia. WikiContradict [14] and CLAIRE [29] show that real Wikipedia passages can support incompatible answers and that LLM-based systems can help automate surfacing such conflicts for human review. However, these works mainly treat conflict as disagreement among textual claims or retrieved passages. (iii) *LLM and RAG conflict* studies disagreement among retrieved sources or between evidence and a model's parametric knowledge. Prior work examines how contextual and parametric knowledge affect QA [22], and analyzes conflicts from outdated knowledge, ambiguity, misinformation, and inconsistent retrieval [34,21,4,35]. Surveys distinguish context–memory, inter-context, and intra-memory conflicts [38], while conflict-aware QA benchmarks test whether systems can separate consensus from conflicting answers [23].

### 2.2 Table-based QA

Tables store rich structured knowledge, and existing table-based QA datasets can again be grouped into three categories: (i) *Table QA* uses a single table as evidence, often through table-to-text generation or question annotation [24,25,26,7]. Examples include FeTaQA that requires free-form reasoning over table cells, ToTTo that focuses on faithful generation from highlighted cells, and NQ-Table that extracts naturally asked questions grounded in Wikipedia tables or infoboxes [13,18]. (ii) *Table-Text QA* combines tables with textual passages. HybridQA and OTT-QA require multi-hop reasoning across both modalities [6,5]. QAMPARI supports multi-answer questions with distributed Wikipedia text paragraph evidence [2], while MoNaCo [37] targets time-consuming questions requiring reasoning chains over dozens of tables and passages. Meanwhile, SportReason focuses on sports-domain reasoning and reconstructs evidence from Table-Text QA datasets, providing a multi-table with multi-text scenario [11]. (iii)

*Heterogeneous QA* goes beyond one table and its passages. CompMix introduces compositional questions over KGs, web tables, and text [8], while TANQ uses sentences, tables, and infoboxes to produce summary-table answers [1]. However, these benchmarks usually assume coherent evidence, leaving inconsistencies across texts, tables, and KGs largely under-explored. The dataset we propose is built from all these benchmarks.

### 2.3 Text-to-SPARQL

KBQA answers natural language questions by translating them into SPARQL queries [27] over knowledge bases such as Wikidata [32]. Earlier systems mainly rely on semantic parsing [41,39], often requiring supervised query annotations. Recent LLM-based methods move toward zero-shot and agentic settings. For instance, GRASP [33] proposes a zero-shot SPARQL generation framework that combines LLM reasoning with dynamic graph exploration, achieving strong generalization across arbitrary RDF knowledge bases without fine-tuning. Similarly, SPINACH [20] proposes a challenging real-world KBQA benchmark derived from complex user queries and an LLM-guided agent that mimics human expert SPARQL construction through iterative navigation. Both show promising results on multiple KBQA benchmarks [30,31,39]. Existing KBQA work focuses on query accuracy and answer retrieval. In contrast, we use Text-to-SPARQL as a bridge between table questions and KG evidence. This setting exposes discrepancies caused by incomplete, outdated, or conflicting knowledge, motivating our framework to detect and categorize inconsistencies across text, tables, and KGs.

## 3 Taxonomy of Knowledge Inconsistency

In our setting, Wikipedia provides table and table-text evidence, while Wikidata provides KG evidence through entities, relations, properties, and qualifiers. Following prior work on knowledge deltas in KG construction, mismatches between these heterogeneous sources may arise from differences in granularity or from contradictions [15]. Since different conflicts require different interpretations and actions [4], we classify answer-level mismatches into a taxonomy of conflicts across text, tables, and KGs.

We derived the taxonomy by inspecting the Wikipedia pages from which the table-based questions were constructed and manually comparing their answers with the corresponding Wikidata item pages. This process revealed recurring inconsistency patterns, which we refined by examining their causes and relating them to prior work [15]. Our proposed taxonomy focuses on answer-level inconsistencies across modalities, rather than on the intrinsic semantic nature of uncertainty.

Overall, our taxonomy distinguishes granularity mismatches from direct contradictions, temporal validity conflicts, and structural incompleteness in the KG, at the schema level (a property or a qualifier does not exist) or at the instance

level (an entity or an edge is missing). This distinction is useful for downstream knowledge maintenance: granularity-related cases indicate which source should be enriched; different-answer and temporal conflicts require verification against context and time; and missing nodes, edges, properties, or qualifiers provide actionable signals for KG completion and schema refinement.

| **Label** | **Question** | **Table-based answer** | **KG Answer** |
|---|---|---|---|
| Same answer | Who was the king of England in 1756? | George II | George II of Great Britain |
| Higher accuracy in KG than in Table | In what movie did Ian Charleson play Eric Liddell, and what year did the movie come out? | Ian Charleson played Eric Liddell in *Chariots of Fire*, in 1981 | Chariots of Fire, 30 Mar 1981; 31 Mar 1981; 15 May 1981; 26 Sep 1981; 9 Apr 1982; 7 May 1982 |
| Higher accuracy in Table than in KG | Who starred in the *pirates of the caribbean* | Johnny Depp \| Geoffrey Rush \| Kevin McNally \| Orlando Bloom \| Keira Knightley \| Jack Davenport \| Jonathan Pryce | Johnny Depp |
| Different answer | What is the elevation of Dakar? | 22 m | 10 m |
| Temporal changes | When is the season finale of designated survivor? | May 16, 2018 | 7 June 2019 |
| Missing edge | Who directed (P57) the TV series *Decoupled (Q110323803)* ? | Hardik Mehta (Q35488479) | - |
| Missing node | What city is the university that taught Angie Barker located in ? | Johnson City | - |
| Missing property/qualifier | Who ran the fastest in the Men's 100 metres in the first semi-final of the 2012 Summer Olympics? | Justin Gatlin (9.82), ahead of Churandy Martina (9.91) and Asafa Powell (9.94). | Usain Bolt \| 9.87 |

**Table 1.** Examples of inconsistencies between Wikipedia table-based answers and Wikidata KG answers. Questions come from existing benchmarks. The symbol "-" indicates that no SPARQL query can be generated and thus no result can be provided from the KG.

**Same answer.** This category occurs when the Wikipedia table-based answer and the Wikidata KG answer refer to the same real-world entity, value, or fact at the same level of accuracy and coverage. These include differences in naming, aliases, language-tagged labels, or minor formatting. For example, the table-based answer *George II* and the KG answer *George II of Great Britain* refer to the same monarch. This category serves as a consistent baseline: the modalities agree, even if they express the answer at slightly different lexical or representational levels.

**Higher accuracy in KG than in Table.** This inconsistency occurs when the KG answer has a higher level of precision or/and completeness than the table-based table-based answer. The two sources are not necessarily contradictory. Instead, they differ in granularity or specificity. For example, in Table 1, the table-based answer states that the movie *Chariots of Fire* came out in 1981, while Wikidata contains multiple release dates, with a day-month-year granularity, depending on countries. The table-based answer derived from a table gives a valid answer, while Wikidata KG provides more detailed information.

**Higher accuracy in Table than in KG.** This inconsistency occurs when the table-based answer has a higher level of precision or/and completeness than the KG answer. In the cast example of Table 1, the table-based answer lists several actors who all exist in Wikidata, while the KG result returns only Johnny Depp. In this case, the KG answer is a subset of the table-based answer.

**Different answer.** This conflict occurs when the table-based answer and the KG answer provide incompatible values for the same question. Unlike specificity-related cases, the mismatch cannot be resolved by adding detail or choosing a more precise answer. For example, a table in Wikipedia reports Dakar's elevation as 22 m, while Wikidata reports 10 m in Table 1. In this case, the two sources may refer to elevations measured at different locations within Dakar.

**Temporal changes.** This conflict occurs when the correct answer depends on the temporal state of the source. In Table 1, the question *When is the season finale of Designated Survivor?* gets the table-based answer of *May 16, 2018*, corresponding to the Season 2 finale. This answer was true when this TV series was expected to end after two seasons. Wikidata returns *June 7, 2019*, corresponding to the Season 3 finale, which is valid today since the TV series was finally renewed and later concluded. Thus, these answers can be considered both valid depending on the temporal snapshot considered. This case is actually frequent, occurring for example with population census, rankings, geographic measurements, or event schedules which are all time dependent.

**Missing edge.** This inconsistency occurs when Wikidata contains the necessary entities and property but lacks the statement needed to connect them. In the *Decoupled* example in Table 1, both the TV series *Decoupled* (Q110323803)[1] and the director Hardik Mehta (Q35488479)[2] exist in Wikidata, but they are not linked with the *director* property (P57).[3] Thus, the KG contains the required entities and schema but lacks the statement needed to derive the table-based answer. This category captures KG incompleteness at the instance level.

**Missing node.** This inconsistency occurs when the KG lacks an entity needed to represent the answer or an intermediate reasoning step. In Table 1, the question *What city is the university that taught Angie Barker located in?* comes from the Southern Conference Hall of Fame Wikipedia page.[4] While most players on the page are linked to Wikipedia entries, Angie Barker has no corresponding

[1] `https://www.wikidata.org/wiki/Q110323803`
[2] `https://www.wikidata.org/wiki/Q35488479`
[3] `https://www.wikidata.org/wiki/Property:P57`
[4] `https://en.wikipedia.org/wiki/Southern_Conference_Hall_of_Fame`

Wikipedia or Wikidata entity. This is another form of incompleteness in the KG at the instance level which lacks the entity/node needed to answer the question.
**Missing property or qualifier.** This inconsistency occurs when the KG contains the relevant entities, but lacks the property or the qualifier needed to express the answer at the required precision. In the Olympic semifinal example in Table 1, the question asks for the fastest runner in the first semifinal, whereas the KG answer returns Usain Bolt who is the winner of the final. Although the Wikidata page[5] includes the *stage reached* property (P2443) with value *semi-final* (Q599999), it does not distinguish the first semifinal from the second one. Thus, the KG schema lacks the fine-grained qualifier needed to represent the answer which is an incompleteness at the schema level of the KG.

## 4 KONTRAST: A Framework for Detecting and Categorizing Inconsistencies Across Modalities

We define *modality-level inconsistency detection* as a task over question–answer pairs grounded in heterogeneous Wikimedia resources. Given a question $q$, Wikipedia provides a table-based answer $a^{\text{table}}$ supported by table or table-text evidence, while Wikidata provides a KG answer $a^{\text{KG}}$ derived from KG evidence. The goal is to determine whether the two answers express the same fact:

$$a^{\text{table}} \equiv a^{\text{KG}}.$$

If they are equivalent, the instance is labeled SAME. Otherwise, it is assigned one of the inconsistency labels in Table 1. Unlike standard QA, where one correct answer is sufficient, our task asks whether two modalities agree on the same information need.
Our approach can be decomposed into three steps:

1. **KG evidence retrieval:** generate and execute a SPARQL query over the KG for each table-text-based question.
2. **Inconsistency detection:** compare the KG answer with the table-based answer to decide whether they are semantically equivalent or not.
3. **Inconsistency categorization:** assign labels describing the inconsistency, distinguishing differences in terms of granularity [15], contradictions [14], temporal mismatches, and KG incompleteness.

### 4.1 KG evidence retrieval

**Table-Text-based QA input.** First, we select table-text based QA datasets whose answers are grounded in Wikipedia text and table evidence and can be verified against KG. Each instance contains a question $q$ and a table-based answer $a^{\text{table}}$, supported by a Wikipedia table, infobox, or table with related text [36,10].

[5] `https://www.wikidata.org/wiki/Q734020`

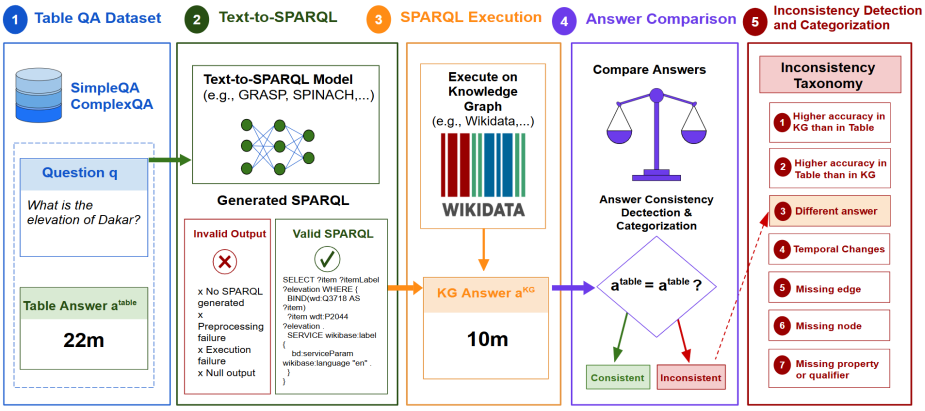



**Fig. 1.** Workflow for detecting and categorizing modality-level inconsistencies by comparing table and KG answers using a knowledge inconsistency taxonomy.

The question defines the information need, and the table-based answer represents the answer derived from Wikipedia tables and text.

**Text-to-SPARQL translation.** A Text-to-SPARQL model translates a natural language question $q$ into a SPARQL query over a KG. This step maps natural language to symbolic KG operations over entities, properties, and qualifiers. We consider two recent systems: GRASP and SPINACH. GRASP is a zero-shot SPARQL generation framework that searches for relevant IRIs, executes intermediate queries, and composes a final executable query. SPINACH is an in-context KBQA agent that simulates expert query construction by navigating entities and properties over multiple steps.

We conducted a pilot study and we observed that GRASP produced more executable queries on average than SPINACH. Its output usually contains a self-contained SPARQL query encoding the full reasoning path, which makes execution and comparison straightforward. By contrast, SPINACH often encodes only the final reasoning step in SPARQL while relying on intermediate exploration outside the final query, making the result harder to interpret. GRASP is also more suitable for scalable experiments. While SPINACH is primarily configured for OpenAI models, GRASP supports OpenAI-compatible endpoints and locally served open-source models such as Qwen [40]. We therefore use GRASP as the Text-to-SPARQL component in our experiments.

### 4.2 Inconsistency detection

**SPARQL execution and normalization.** We separate the outputs generated into *valid* and *invalid* cases. A case is *valid* if the model produces an executable SPARQL query, regardless of whether the query returns a non-empty set or not. For each valid query, we execute it over Wikidata and collect the returned values as $a^{\mathrm{KG}}$. Value-bearing results are passed to comparison and categoriza-

tion. Empty result sets are stored separately. Invalid cases include null outputs, missing queries, execution failures, and preprocessing failures. We retain them because they may reflect not only Text-to-SPARQL errors, but also gaps in KG coverage or schema expressiveness. Before comparison, we normalize the raw KG output by removing table metadata, Wikidata identifiers, and datatype annotations, yielding a compact textual representation of $a^{\mathrm{KG}}$.

**Answer comparison.** We compare $a^{\mathrm{table}}$ and $a^{\mathrm{KG}}$ using semantic equivalence rather than exact string matching. An instance is labeled SAME if the two answers denote the same entity, value, or fact despite surface form variation, such as "George II" and "George II of Great Britain". Otherwise, it is treated as an inconsistency candidate. For example, "22 m" and "10 m" for Dakar's elevation are conflicting values and are passed to categorization.

### 4.3 Inconsistency categorization

**Simple heuristics.** Categorization begins with lightweight automatic heuristics. We normalize $a^{\mathrm{table}}$ and $a^{\mathrm{KG}}$ into sets of answer units, with special handling for dates and multi-value answers. Each table–KG answer pair receives an alignment score based on exact normalized matching, date-aware matching, or SBERT similarity [28] for aliases and minor lexical variation.

We compute recall from table answers to KG answers and precision from KG answers to table answers. An instance is labeled SAME only when the two sets have the same size and both precision and recall are at least 0.95. This threshold was chosen through iterative human inspection of borderline cases, preserving clear matches while separating cases with missing or extra answers. If the KG covers the table answer but adds more specific values, we label it HIGHER ACCURACY IN KG THAN IN TABLE. If the table answer contains additional values missing from the KG, we label it HIGHER ACCURACY IN TABLE THAN IN KG. Cases not confidently assigned by these rules are deferred to a judge LLM.

**LLM-as-a-judge.** The remaining ambiguous cases are categorized with an LLM-as-a-judge protocol. The prompt contains five label definitions and three in-context examples per label, forming a 15-shot prompt. Each input includes the question, the table answer $a^{\mathrm{table}}$, and the KG answer $a^{\mathrm{KG}}$. The model outputs the taxonomy label, severity level, and a concise explanation. This stage is designed to flag meaningful inconsistency labels for human review.

## 5 Experiments and Results

### 5.1 Data Collection

We construct a table-text-based QA collection to evaluate whether questions grounded in Wikipedia tables, infoboxes, and related passages can be translated into executable SPARQL queries over Wikidata KG. Each QA pair serves as a compact representation of the underlying table or semi-structured evidence: the question specifies the information needed, and the table-based answer captures

| Group | Source | Creator | Evidence Type | # Questions |
|---|---|---|---|---|
| SimpleQA | NQ-Table [13,18] | Human | Table or infobox | 966 |
| | CompMix-simple [8] | Human | Table and infobox | 326 |
| | QAMPARI [2] | Template | Table and text | 78 |
| | **Subtotal** | | | **1,370** |
| ComplexQA | CompMix-infobox [8] | Human | Infobox | 300 |
| | CompMix-table [8] | Human | Table | 300 |
| | MoNaCo-time [37] | Human | Table and text | 150 |
| | MoNaCo [37] | Human | Table and text | 150 |
| | OTT-QA [5] | Template | Table and text | 400 |
| | SportReason [11] | LLM-RAG | Table, text, and infobox | 200 |
| | **Subtotal** | | | **1,500** |
| **Total** | | | | **2,870** |

**Table 2.** Dataset sources for Text-to-SPARQL generation.

the answer derived from Wikipedia tables and text. Rather than reusing existing benchmarks directly, we re-organize selected instances into a unified Text-to-SPARQL evaluation collection. We select data according to three criteria:

- **Grounded evidence:** questions must be grounded in Wikipedia tables, infoboxes, or tables with related textual passages.
- **Reasoning diversity:** the collection should cover a broad range of reasoning complexity and skills, including direct lookup, multi-hop reasoning, temporal reasoning, composition, aggregation, and numerical reasoning.
- **Comparable answers:** table-based answers must be comparable with Wikidata query results. We therefore prioritize short strings, keywords, entities, and entity lists over long free-form answers.

Table 2 summarizes the source datasets that have been used. We group them into *SimpleQA* and *ComplexQA*, which differ in the complexity of the information needed and the amount of reasoning required.

*SimpleQA* contains natural or lightly structured questions that can be answered with direct table, infobox, or table-related textual evidence. It combines NQ-Table, simple CompMix questions without temporal, aggregation, or conjunction cues, and QAMPARI multi-answer questions [13,18,8,2].

*ComplexQA* contains questions that require richer evidence integration or more complex reasoning. It includes the longest table- and infobox-grounded questions from CompMix, time-dependent and non-time-dependent questions from MoNaCo, decontextualized OTT-QA development questions, and SportReason questions for numerical reasoning over text, tables, and infoboxes [8,37,5,11,1].

### 5.2 Experiment Setup

We evaluate KONTRAST with three Qwen3 backbones: Qwen3-235B-A22B-Thinking, Qwen3-30B-A3B-Thinking, and Qwen3-4B-Instruct [40]. These models cover different capabilities and inference settings, allowing us to test how model size (4B,

30B and 235B) affects modality-level inconsistency detection. We use GRASP for Text-to-SPARQL generation [33].

For each instance, we provide only the question text to the Text-to-SPARQL model. The generated query is executed against Wikidata, and the resulting KG answer is compared with the table-based answer from the original table-QA datasets.

All SPARQL queries are executed on the QLever Wikidata endpoint,[6] using the fixed 2025-05-10 Wikidata dump released with GRASP.[7] This snapshot makes results reproducible and avoids changes from live Wikidata updates.

For categorization, we first apply rule-based heuristics with SBERT matching using `all-MiniLM-L6-v2`. Remaining cases are classified by a Qwen3-30B-A3B-Thinking judge [40] with a 15-shot taxonomy prompt) (Section 4.3).

For taxonomy analysis, we use the Analysis Set in Table 3, excluding outputs with more than 10 rows, because the Text-to-SPARQL model serializes only the first five and last five rows, and duplicate questions from source datasets. This "row" filter removes 251 cases for Qwen3-4B-Instruct (8.72%), 97 for Qwen3-30B-Thinking (3.37%), and 95 for Qwen3-235B-Thinking (3.30%). Duplicate removal affects 12, 14, and 17 value-bearing cases, respectively.

### 5.3 Results

*SPARQL validity.* We first measure whether generated SPARQL queries execute successfully. As shown in Table 3, Qwen3-30B-Thinking achieves the highest validity, with 2,576 valid outputs out of 2,870 cases (89.8%). It performs best on both SimpleQA (93.3%) and ComplexQA (86.5%). Qwen3-235B-Thinking ranks second overall (79.5%), while Qwen3-4B-Instruct is lowest (71.1%), especially on complex datasets.

*From executable queries to usable answers.* A valid query may still return an empty answer. Therefore, Table 3 separates valid outputs into empty and value-bearing results. Qwen3-30B-Thinking produces the biggest number of executable queries, but the lowest value-bearing rate among valid outputs (49.6%). By comparison, Qwen3-235B-Thinking achieves the highest value-bearing rate among valid outputs (63.7%). Thus, execution validity captures whether a query runs, while the value-bearing rate better indicates whether it yields usable KG evidence.

*Inconsistency patterns.* Table 4 summarizes modality-level inconsistencies in the filtered Analysis Set. Across SimpleQA, ComplexQA, and all models, *Different answer* is the largest category, covering (i) true cross-modal knowledge conflicts; (ii) Text-to-SPARQL translation noise where the model selects an incorrect entity or property; and (iii) missing property or qualifier cases (schema level),

[6] `https://qlever.dev/api/wikidata`

[7] `https://ad-publications.cs.uni-freiburg.de/grasp/kg-index-search-index/`

| Dataset | Qwen3-4B | | | Qwen3-30B | | | Qwen3-235B | | |
|---|---|---|---|---|---|---|---|---|---|
| | $n$ | **Exec.** | **Ans.** | $n$ | **Exec.** | **Ans.** | $n$ | **Exec.** | **Ans.** |
| NQ-Table | 966 | 79.6 | 56.8 | 966 | 94.7 | 54.5 | 966 | 91.2 | 67.7 |
| CompMix-simple | 326 | 88.3 | 68.8 | 326 | 90.2 | 57.1 | 326 | 89.0 | 69.0 |
| QAMPARI | 78 | 53.8 | 59.5 | 78 | 88.5 | 47.8 | 78 | 96.2 | 72.0 |
| SimpleQA | 1,370 | 80.2 | 60.1 | 1,370 | **93.3** | 54.8 | 1,370 | 90.9 | **68.2** |
| CompMix-infobox | 300 | 91.3 | 76.6 | 300 | 96.3 | 77.2 | 300 | 67.7 | 84.2 |
| CompMix-table | 300 | 84.7 | 47.2 | 300 | 92.7 | 49.6 | 300 | 89.0 | 66.7 |
| MoNaCo-time | 150 | 70.0 | 44.8 | 150 | 78.0 | 30.8 | 150 | 67.3 | 52.5 |
| MoNaCo | 150 | 70.7 | 47.2 | 150 | 82.0 | 30.9 | 150 | 34.0 | 56.9 |
| OTT-QA | 400 | 35.5 | 37.3 | 400 | 84.2 | 24.6 | 400 | 66.5 | 34.6 |
| SportReason | 200 | 30.5 | 41.0 | 200 | 77.0 | 39.0 | 200 | 73.5 | 54.4 |
| ComplexQA | 1,500 | 62.8 | 53.6 | 1,500 | **86.5** | 44.5 | 1,500 | 69.0 | **58.3** |
| Total | 2,870 | 71.1 | 57.1 | 2,870 | **89.8** | 49.6 | 2,870 | 79.5 | **63.7** |
| Analysis Set | 1,778 | 100.0 | 50.7 | 2,465 | 100.0 | 47.3 | 2,169 | 100.0 | **61.8** |

**Table 3.** Text-to-SPARQL execution quality by dataset and model. $n$ denotes the number of evaluated cases; for **Analysis Set**, it denotes the number of question–answer pairs. **Exec.** is the valid-execution rate, and **Ans.** is the value-bearing rate among valid executions. Bold marks the best result across models for summary rows.

where the Text-to-SPARQL model falls back to the closest available KG entities. Granularity mismatches are also frequent. Across models, *Higher accuracy in KG than in Table* accounts for 11.3–13.5% of cases, while *Higher accuracy in Table than in KG* accounts for 7.3–9.4%. This shows that neither modality is uniformly more complete. Instead, tables, text, and KGs provide complementary evidence for knowledge reconciliation.

*Effect of model scale.* Larger models produce more reliable KG answers and fewer translation-induced mismatches. In the filtered Analysis Set, Qwen3-235B-Thinking has the lowest inconsistency rate among value-bearing cases (61.4%), followed by Qwen3-30B-Thinking (63.8%) and Qwen3-4B-Instruct (68.4%). This trend holds for both SimpleQA and ComplexQA, where Qwen3-235B-Thinking also yields the highest proportion of *Same* cases. Overall, stronger reasoning ability Text-to-SPARQL models improve the quality of downstream inconsistency analysis.

*Interpreting question complexity.* The higher inconsistency rates on SimpleQA should not be interpreted as evidence that simple datasets contain more knowledge conflicts. Rather, SimpleQA questions are translated into SPARQL more reliably, making their KG answers more suitable for inconsistency analysis. ComplexQA questions require more difficult entity linking, relation selection, and compositional reasoning, which introduces additional Text-to-SPARQL noise.

| Taxonomy Label | SimpleQA | ComplexQA | All |
|---|---|---|---|
| | Qwen3-4B / Qwen3-30B / Qwen3-235B | | |
| Same | 29.8 / 35.9 / 38.4 | 34.0 / 36.6 / 38.9 | 31.6 / 36.2 / 38.6 |
| Higher accuracy in KG than in Table | 10.8 / 11.0 / 12.4 | 17.1 / 12.3 / 9.7 | 13.5 / 11.6 / 11.3 |
| Higher accuracy in Table than in KG | 9.7 / 10.3 / 9.9 | 4.2 / 6.7 / 8.7 | 7.3 / 8.7 / 9.4 |
| Different answer | **47.4** / **38.6** / **36.9** | **44.4** / **43.5** / **41.2** | **46.1** / **40.9** / **38.7** |
| Temporal changes | 2.3 / 4.1 / 2.4 | 0.3 / 0.9 / 1.4 | 1.4 / 2.7 / 2.0 |
| Inconsistent rate | 70.2 / 64.1 / **61.6** | 66.0 / 63.4 / **61.1** | 68.4 / 63.8 / **61.4** |

**Table 4.** Distribution of modality-level inconsistency categories in the filtered Analysis Set. Each cell reports percentages in the model order shown in the header. Percentages are computed within each QA group and model. Bold marks the dominant non-*Same* inconsistency category and the lowest inconsistent rate.

Therefore, group-level differences reflect both underlying cross-modal inconsistency and the current limitations of automatic SPARQL generation.

### 5.4 Error Analysis

To analyze Text-to-SPARQL failures, we randomly inspect 15 invalid cases, selecting five from 3 sub categories: *No SPARQL generated*, *Execution failure*, and *Preprocessing failure*. We exclude *Null output*, since it leaves no generation trace to inspect. This inspection reveals four recurring patterns.

*Empty results indicate missing edges or qualifiers.* In 50% of verified errors, `empty_sparql_result` is caused by missing KG structural elements. For example, for the question *Who is the publisher of Writers & Lovers book?*, the Text-to-SPARQL model links the correct book entity, but Wikidata lacks the publisher edge `P123` statement with the answer. Similarly, for *Did John Prine win a Grammy for Fair & Square?*, Wikidata contains the relevant Grammy relation, but lacks the `for work` qualifier `P1686` needed to connect the award to the album.

*No-query cases reflect ambiguous questions.* All inspected cases are due to questions that are under-specified without additional context. For example, *Who is their current manufacturer of their uniform kits?* cannot be grounded reliably because the entity referred to by "their" is not available from the question alone.

*Execution failures mix timeouts, hallucination, and missing entities.* Among the inspected cases, 40% are endpoint temporary timeouts that succeed after re-running. Another 20% are LLM hallucination, where the Text-to-SPARQL model produces an answer from parametric knowledge rather than executable KG evidence. The remaining 40% are due to missing entities, such as *Who played the character of Sevika in Arcane?*, where the character Sevika from *Arcane*[8] is not available as a required Wikidata entity.

[8] `https://en.wikipedia.org/wiki/Arcane_(TV_series)`

*Preprocessing failures are malformed SPARQL.* All inspected cases are caused by invalid SPARQL syntax. These are generation-quality errors rather than KG incompleteness. For example, some generated queries miss the predicate, producing an incomplete subject–predicate–object triple pattern.

### 5.5 Human Evaluation

We conduct a human evaluation to check the reliability of the two automatic labeling stages. The evaluation focuses on answer-level categorization: one annotator verifies whether the assigned taxonomy label correctly describes the relation between the table-based answer and the KG answer.

For the heuristic stage, we inspect 30 cases. For each of SAME, HIGHER ACCURACY IN TABLE THAN KG, and HIGHER ACCURACY IN KG THAN TABLE, we sample the five highest-scoring and five lowest-scoring instances by alignment score, covering both clear and borderline cases. All 30 labels are judged correctly.

For the LLM-as-a-judge stage, we inspect 50 cases: the first 10 instances for each of its five output labels, SAME, HIGHER ACCURACY IN TABLE THAN KG, HIGHER ACCURACY IN KG THAN TABLE, DIFFERENT ANSWER, and TEMPORAL CHANGES. All 50 labels are judged correctly revealing a perfect alignment between human evaluation and the LLM judge.

These results provide evidence that both stages produce reliable taxonomy labels on the inspected samples. They are intended as a quality check of the labeling pipeline, rather than a full annotation study.

### 5.6 Limitations

*Question naturalness.* Our Text-to-SPARQL pipeline is sensitive to question naturalness. Human-written questions usually yield higher SPARQL validity, while template- or answer-derived questions can be harder to parse. For example, OTT-QA asks: *What is the capacity of the mosque that is on the list of largest mosques, and that was opened to the public 22 February 1978?*. Such unnatural constraints make entity and property selection harder. Thus, a low validity may reflect a weakness regarding the formulation of the question rather than a weakness of the Text-to-SPARQL model or a KG incompleteness.

*Limits of automatic structural diagnosis.* Our taxonomy includes three structural KG incompleteness: *Missing edge*, *Missing node*, and *Missing property or qualifier*. These categories are difficult to detect automatically from SPARQL execution results alone. They can appear in both valid and invalid generations, and their surface forms often overlap with ordinary Text-to-SPARQL errors.

*Missing property or qualifier.* A missing property or qualifier may still produce a valid query if the model falls back to a related KG fact. For example, for the question about the most passing yards in a single NFL game, the table-based answer is 554 yards,[9], whereas the KG answer is Norm Van Brocklin, the player

[9] `https://en.wikipedia.org/wiki/List_of_500-yard_passing_games_in_the_NFL`

associated with that record. The query is executable, but the mismatch arises because Wikidata does not encode the record using *record held* (P1000).[10] In invalid cases, the model may instead state that the required fact is not available in the KG. However, this cannot be accepted directly as evidence of KG incompleteness, because the failure may also come from selecting the wrong entity or property.

*Missing edges and missing nodes.* The same ambiguity holds for missing edges and missing nodes. A missing edge may lead to an empty valid result, a fallback to a semantically close property, or an invalid generation claiming that the relation is unavailable. A missing node may cause the model to select a related but incorrect entity, or to report that no matching entity exists. In all cases, automatic execution signals are insufficient to distinguish true KG incompleteness from semantic parsing errors.

## 6 Conclusion and Future Work

We introduced modality-level inconsistency detection, a task for identifying and categorizing knowledge mismatches across text, tables, and KGs. We proposed a taxonomy of cross-modal inconsistency types and presented KONTRAST, an automatic system that uses Text-to-SPARQL and LLM reasoning to detect and categorize cross-modal knowledge inconsistencies.

Our analysis shows that modality-level inconsistencies are measurable at scale. With Qwen3-235B-Thinking, 61.4% of value-bearing cases are not fully aligned with the table answer. Excluding the mixed *Different answer* category, 22.7% of cases reflect interpretable differences, including Higher accuracy in KG than in Table cases, and Higher accuracy in Table than in KG cases, and temporal shifts. These cases provide actionable signals for correction, enrichment, and temporal verification across texts, tables and KGs.

These results show that text, tables, and KGs should not be treated as isolated knowledge sources. Their disagreements can reveal where one modality complements or corrects another. KONTRAST provides a practical way to discover and classify cross-modal inconsistencies for human review, supporting knowledge correction and enrichment at scale. By formalizing the task, taxonomy, and evaluation setting, this work provides a basis for future research on cross-modal knowledge consistency.

Future work should extend human annotation to both value-bearing and invalid cases for the structural labels *Missing edge*, *Missing node*, and *Missing property or qualifier*. This would separate KG incompleteness from Text-to-SPARQL errors and support automatic diagnosis. A further direction is reconciliation: deciding when to update the KG, refine table evidence, add qualifiers, or keep both answers under different temporal or contextual conditions.

[10] `https://www.wikidata.org/wiki/Property:P1000`

## Declaration of Use of Generative AI

GPT-5.2 has been used to draft the workflow figure. All AI-generated content was reviewed and edited by the authors, who take full responsibility for the final manuscript.

## Acknowledgments

This work was partially supported by the French National Research Agency (Agence Nationale de la Recherche - ANR) under the ECLADATTA project, grant number ANR-22-CE23-0020.

## References

1. Akhtar, M., Pang, C., Marzoca, A., Altun, Y., Eisenschlos, J.M.: TANQ: An open domain dataset of table answered questions. Transactions of the Association for Computational Linguistics **13**, 461–480 (2025). https://doi.org/10.1162/tacl_a_00749, https://aclanthology.org/2025.tacl-1.23/
2. Amouyal, S., Wolfson, T., Rubin, O., Yoran, O., Herzig, J., Berant, J.: QAMPARI: A benchmark for open-domain questions with many answers. In: 3rd Workshop on Natural Language Generation, Evaluation, and Metrics. pp. 97–110. Association for Computational Linguistics, Singapore (Dec 2023), https://aclanthology.org/2023.gem-1.9/
3. Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D.: Language models are few-shot learners. In: Advances in Neural Information Processing Systems (NeurIPS). vol. 33, pp. 1877–1901 (2020), https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html
4. Cattan, A., Jacovi, A., Ram, O., Herzig, J., Aharoni, R., Goldshtein, S., Ofek, E., Szpektor, I., Caciularu, A.: DRAGged into conflicts: Detecting and addressing conflicting sources in search-augmented LLMs (2025), https://arxiv.org/abs/2506.08500
5. Chen, W., Chang, M.W., Schlinger, E., Wang, W., Cohen, W.: Open question answering over tables and text. In: 9th International Conference on Learning Representations (ICLR) (2021)
6. Chen, W., Zha, H., Chen, Z., Xiong, W., Wang, H., Wang, W.Y.: HybridQA: A dataset of multi-hop question answering over tabular and textual data. In: Findings of the Association for Computational Linguistics: EMNLP. pp. 1026–1036. Association for Computational Linguistics (2020). https://doi.org/10.18653/v1/2020.findings-emnlp.91, https://aclanthology.org/2020.findings-emnlp.91/
7. Cheng, Z., Dong, H., Wang, Z., Jia, R., Guo, J., Gao, Y., Han, S., Lou, J.G., Zhang, D.: HiTab: A hierarchical table dataset for question answering and natural language generation. In: 60th Annual Meeting of the Association for Computational Linguistics (ACL). pp. 1094–1110. Association for Computational Linguistics, Dublin, Ireland (May 2022). https://doi.org/10.18653/v1/2022.acl-long.78, https://aclanthology.org/2022.acl-long.78/

8. Christmann, P., Saha Roy, R., Weikum, G.: CompMix: A benchmark for heterogeneous question answering. In: Companion Proceedings of the ACM Web Conference 2024. pp. 1091–1094. Association for Computing Machinery, New York, NY, USA (2024). https://doi.org/10.1145/3589335.3651444, https://doi.org/10.1145/3589335.3651444
9. Djebri, A.E.A., Tettamanzi, A.G.B., Gandon, F.: Publishing uncertainty on the semantic web: Blurring the LOD bubbles. In: 24th International Conference on Conceptual Structures (ICCS). pp. 42–56. Springer, Marburg, Germany (2019). https://doi.org/10.1007/978-3-030-23182-8_4, https://doi.org/10.1007/978-3-030-23182-8_4
10. Ettaleb, M., Ehrhart, T., Aussenac-Gilles, N., Chabot, Y., Kamel, M., Moriceau, V., Troncy, R., Wei, F.: ReTaT: A Unified Benchmark for Relation Extraction across Text and Table. In: International Conference on Language Resources and Evaluation (LREC). Mallorca, Spain (2026)
11. Feng, K., Zhang, S., Chen, B., Zhao, Y., Zhao, C.: SportReason: Evaluating retrieval-augmented reasoning across tables and text for sports question answering. In: Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 649–662. Association for Computational Linguistics, Suzhou, China (2025). https://doi.org/10.18653/v1/2025.emnlp-main.34, https://aclanthology.org/2025.emnlp-main.34/
12. Gao, L., Biderman, S., Black, S., Golding, L., Hoppe, T., Foster, C., Phang, J., He, H., Thite, A., Nabeshima, N., Prescod-Weinstein, C., Leahy, C.: The Pile: An 800gb dataset of diverse text for language modeling (2020), https://arxiv.org/abs/2101.00027
13. Herzig, J., Müller, T., Krichene, S., Eisenschlos, J.: Open domain question answering over tables via dense retrieval. In: Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 512–519. Association for Computational Linguistics (2021). https://doi.org/10.18653/v1/2021.naacl-main.43, https://aclanthology.org/2021.naacl-main.43/
14. Hou, Y., Pascale, A., Carnerero-Cano, J., Tchrakian, T., Marinescu, R., Daly, E., Padhi, I., Sattigeri, P.: WikiContradict: A benchmark for evaluating LLMs on real-world knowledge conflicts from Wikipedia. In: Advances in Neural Information Processing Systems 37 (2024). https://doi.org/10.52202/079017-3481, https://proceedings.neurips.cc/paper_files/paper/2024/hash/c63819755591ea972f8570beffca6b1b-Abstract-Datasets_and_Benchmarks_Track.html
15. Jarnac, L., Chabot, Y., Couceiro, M.: Uncertainty Management in the Construction of Knowledge Graphs: A Survey. Transactions on Graph Data and Knowledge **3**(1), 3:1–3:48 (2025). https://doi.org/10.4230/TGDK.3.1.3, https://drops.dagstuhl.de/entities/document/10.4230/TGDK.3.1.3
16. Jarnac, L., et al.: Wikiconflict: A new dataset for conflicting data reconciliation in knowledge graph construction. In: 13th Knowledge Capture Conference (K-CAP). pp. 215–218. Association for Computing Machinery (2025). https://doi.org/10.1145/3731443.3771371, https://doi.org/10.1145/3731443.3771371
17. Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., Yih, W.t.: Dense passage retrieval for open-domain question answering. In: Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 6769–6781. Association for Computational Linguistics, Online (2020).

https://doi.org/10.18653/v1/2020.emnlp-main.550, https://aclanthology.org/2020.emnlp-main.550/
18. Kwiatkowski, T., Palomaki, J., Redfield, O., Collins, M., Parikh, A., Alberti, C., Epstein, D., Polosukhin, I., Devlin, J., Lee, K., Toutanova, K., Jones, L., Kelcey, M., Chang, M.W., Dai, A.M., Uszkoreit, J., Le, Q., Petrov, S.: Natural questions: A benchmark for question answering research. Transactions of the Association for Computational Linguistics **7**, 452–466 (2019). https://doi.org/10.1162/tacl_a_00276, https://aclanthology.org/Q19-1026/
19. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.t., Rocktäschel, T., Riedel, S., Kiela, D.: Retrieval-augmented generation for knowledge-intensive NLP tasks. In: Advances in Neural Information Processing Systems. vol. 33, pp. 9459–9474 (2020), https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html
20. Liu, S., Semnani, S., Triedman, H., Xu, J., Zhao, I.D., Lam, M.: SPINACH: SPARQL-based information navigation for challenging real-world questions. In: Findings of the Association for Computational Linguistics: EMNLP. pp. 15977–16001. Association for Computational Linguistics, Miami, Florida, USA (2024). https://doi.org/10.18653/v1/2024.findings-emnlp.938, https://aclanthology.org/2024.findings-emnlp.938/
21. Liu, S., Ning, Q., Halder, K., Qi, Z., Xiao, W., Htut, P.M., Zhang, Y., Anna John, N., Min, B., Benajiba, Y., Roth, D.: Open domain question answering with conflicting contexts. In: Findings of the Association for Computational Linguistics: NAACL. pp. 1838–1854. Association for Computational Linguistics, Albuquerque, New Mexico (2025). https://doi.org/10.18653/v1/2025.findings-naacl.99, https://aclanthology.org/2025.findings-naacl.99/
22. Longpre, S., Perisetla, K., Chen, A., Ramesh, N., DuBois, C., Singh, S.: Entity-based knowledge conflicts in question answering. In: Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 7052–7063. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic (2021). https://doi.org/10.18653/v1/2021.emnlp-main.565, https://aclanthology.org/2021.emnlp-main.565/
23. Nachshoni, E., Cattan, A., Amar, S., Shapira, O., Dagan, I.: Consensus or conflict? fine-grained evaluation of conflicting answers in question-answering. In: 2nd Workshop on Uncertainty-Aware NLP. Association for Computational Linguistics, Vienna, Austria (2025), https://aclanthology.org/2025.uncertainlp-main.13/
24. Nan, L., Hsieh, C., Mao, Z., Lin, X.V., Verma, N., Zhang, R., Kryściński, W., Schoelkopf, H., Kong, R., Tang, X., Mutuma, M., Rosand, B., Trindade, I., Bandaru, R., Cunningham, J., Xiong, C., Radev, D., Radev, D.: FeTaQA: Free-form table question answering. Transactions of the Association for Computational Linguistics **10**, 35–49 (2022). https://doi.org/10.1162/tacl_a_00446, https://aclanthology.org/2022.tacl-1.3/
25. Parikh, A., Wang, X., Gehrmann, S., Faruqui, M., Dhingra, B., Yang, D., Das, D.: ToTTo: A controlled table-to-text generation dataset. In: Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 1173–1186. Association for Computational Linguistics, Online (2020). https://doi.org/10.18653/v1/2020.emnlp-main.89, https://aclanthology.org/2020.emnlp-main.89/
26. Pasupat, P., Liang, P.: Compositional semantic parsing on semi-structured tables. In: 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing. pp. 1470–

1480. Association for Computational Linguistics, Beijing, China (2015). https://doi.org/10.3115/v1/P15-1142, https://aclanthology.org/P15-1142/
27. Pérez, J., Arenas, M., Gutierrez, C.: Semantics and complexity of SPARQL. In: International Semantic Web Conference (ISWC). pp. 30–43. Springer (2006)
28. Reimers, N., Gurevych, I.: Sentence-bert: Sentence embeddings using siamese bert-networks. In: Conference on Empirical Methods in Natural Language Processing (EMNLP) (2019)
29. Semnani, S., Burapacheep, J., Khatua, A., Atchariyachanvanit, T., Wang, Z., Lam, M.: Detecting corpus-level knowledge inconsistencies in Wikipedia with large language models. In: Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 34839–34866. Association for Computational Linguistics, Suzhou, China (2025). https://doi.org/10.18653/v1/2025.emnlp-main.1765, https://aclanthology.org/2025.emnlp-main.1765/
30. Usbeck, R., Ngonga Ngomo, A.C., Haarmann, B., Krithara, A., Röder, M., Napolitano, G.: 7th open challenge on question answering over linked data (QALD-7). In: Semantic Web Evaluation Challenge. pp. 59–69. Springer (2017)
31. Usbeck, R., Yan, X., Perevalov, A., Jiang, L., Schulz, J., Kraft, A., Möller, C., Huang, J., Reineke, J., Ngonga Ngomo, A.C., Saleem, M., Both, A.: QALD-10 – the 10th challenge on question answering over linked data. Semantic Web (2023)
32. Vrandečić, D., Krötzsch, M.: Wikidata: A free collaborative knowledgebase. Communications of the ACM **57**(10), 78–85 (2014)
33. Walter, S., Bast, H.: GRASP: Generic reasoning and SPARQL generation across knowledge graphs. In: 24th International Semantic Web Conference (ISWC). pp. 271–289. Springer, Nara, Japan (2025). https://doi.org/10.1007/978-3-032-09527-5_15, https://doi.org/10.1007/978-3-032-09527-5_15
34. Wan, A., Wallace, E., Klein, D.: What evidence do language models find convincing? In: 62nd Annual Meeting of the Association for Computational Linguistics (ACL). pp. 7663–7695. Association for Computational Linguistics, Bangkok, Thailand (2024). https://doi.org/10.18653/v1/2024.acl-long.403, https://aclanthology.org/2024.acl-long.403/
35. Wang, H., Prasad, A., Stengel-Eskin, E., Bansal, M.: Retrieval-augmented generation with conflicting evidence. In: 2nd Conference on Language Modeling (2025), https://openreview.net/forum?id=z1MHB2m3V9
36. Wei, F., Ehrhart, T., Troncy, R.: From Rows to Narratives: Benchmarking Semantic Relatedness Across Tables and Paragraphs. In: International Workshop on Extraction from Triplet Text-Table-Knowledge Graph (TRIPLET) (2026)
37. Wolfson, T., Trivedi, H., Geva, M., Goldberg, Y., Roth, D., Khot, T., Sabharwal, A., Tsarfaty, R.: MoNaCo: More natural and complex questions for reasoning across dozens of documents. Transactions of the Association for Computational Linguistics **14**, 23–46 (2026). https://doi.org/10.1162/tacl.a.64, https://aclanthology.org/2026.tacl-1.2/
38. Xu, R., Qi, Z., Guo, Z., Wang, C., Wang, H., Zhang, Y., Xu, W.: Knowledge conflicts for LLMs: A survey. In: Conference on Empirical Methods in Natural Language Processing (2024). pp. 8541–8565. Association for Computational Linguistics, Miami, Florida, USA (2024). https://doi.org/10.18653/v1/2024.emnlp-main.486, https://aclanthology.org/2024.emnlp-main.486/
39. Xu, S., Liu, S., Culhane, T., Pertseva, E., Wu, M.H., Semnani, S., Lam, M.: Fine-tuned LLMs know more, hallucinate less with few-shot sequence-to-sequence semantic parsing over Wikidata. In: Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 5778–5791. Association for Computational

Linguistics, Singapore (2023). https://doi.org/10.18653/v1/2023.emnlp-main.353, https://aclanthology.org/2023.emnlp-main.353/

40. Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., Zheng, C., Liu, D., Zhou, F., Huang, F., Hu, F., Ge, H., Wei, H., Lin, H., Tang, J., Yang, J., Tu, J., Zhang, J., Yang, J., Yang, J., Zhou, J., Zhou, J., Lin, J., Dang, K., Bao, K., Yang, K., Yu, L., Deng, L., Li, M., Xue, M., Li, M., Zhang, P., Wang, P., Zhu, Q., Men, R., Gao, R., Liu, S., Luo, S., Jiang, T., Tang, T., Yin, W., Ren, X., Wang, X., Zhang, X., Ren, X., Fan, Y., Su, Y., Zhang, Y., Zhang, Y., Wan, Y., Liu, Y., Wang, Z., Cui, Z., Zhang, Z., Zhou, Z., Qiu, Z.: Qwen3 technical report (2025), https://arxiv.org/abs/2505.09388
41. Yih, W.t., Chang, M.W., He, X., Gao, J.: Semantic parsing via staged query graph generation: Question answering with knowledge base. In: 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing. pp. 1321–1331. Association for Computational Linguistics, Beijing, China (2015). https://doi.org/10.3115/v1/P15-1128, https://aclanthology.org/P15-1128/