Detecting Corpus-Level Knowledge Inconsistencies in Wikipedia with Large Language Models

Sina J. Semnani **Jirayu Burapacheep** **Arpandeep Khatua**
Thanawan Atchariyachanvanit **Zheng Wang** **Monica S. Lam**
Computer Science Department
Stanford University, Stanford, CA
{sinaj,jirayu,akhatua,thanawan,peterwz,lam}@cs.stanford.edu

Abstract

Wikipedia is the largest open knowledge corpus, widely used worldwide and serving as a key resource for training large language models (LLMs) and retrieval-augmented generation (RAG) systems. Ensuring its accuracy is therefore critical. But how accurate is Wikipedia, and how can we improve it?

We focus on *inconsistencies*, a specific type of factual inaccuracy, and introduce the task of *corpus-level inconsistency detection*. We present CLAIRE, an agentic system that combines LLM reasoning with retrieval to surface potentially inconsistent claims along with contextual evidence for human review. In a user study with experienced Wikipedia editors, 87.5% reported higher confidence when using CLAIRE, and participants identified 64.7% more inconsistencies in the same amount of time.

Combining CLAIRE with human annotation, we contribute WIKICOLLIDE, the first benchmark of real Wikipedia inconsistencies. Using random sampling with CLAIRE-assisted analysis, we find that at least 3.3% of English Wikipedia facts contradict another fact, with inconsistencies propagating into 7.3% of FEVEROUS and 4.0% of AmbigQA examples. Benchmarking strong baselines on this dataset reveals substantial headroom: the best fully automated system achieves an AUROC of only 75.1%.

Our results show that contradictions are a measurable component of Wikipedia and that LLM-based systems like CLAIRE can provide a practical tool to help editors improve knowledge consistency at scale. [1]

1 Introduction

Wikipedia is a widely used source of knowledge, attracting billions of monthly visitors (Bianchi, 2024). Although initially criticized for reliability, the English Wikipedia later gained broad acceptance as a reputable source (The Economist, 2021). Beyond public use, it plays a central role in natural language processing (NLP) research: Wikipedia is used to train large language models (LLMs), provide ground truth for retrieval-augmented generation (RAG) systems (Semnani et al., 2023; Lewis et al., 2020; Guu et al., 2020; Zhang et al., 2024a), and supply gold answers for question answering and fact verification.

Given this reliance, ensuring Wikipedia's accuracy is critical. We focus specifically on internal inconsistencies: contradictory facts within Wikipedia that indicate errors requiring correction through consultation of original sources. In a crowdsourced repository, inconsistencies can arise from outdated information, limited awareness of related content during editing, or simple human error.

The corpus's vast scale makes comprehensive verification challenging for both humans and automated tools. While Wikipedia is often used to detect hallucinations in LLMs, we instead leverage LLMs to detect inconsistencies in a human-curated corpus. Our contributions are as follows:

We formalize the task of Corpus-Level Inconsistency Detection (CLID). Given a fact from a corpus, the goal is to identify at least one other fact within the same corpus that contradicts it. While inconsistency detection has been studied at the sentence-pair and document levels, corpus-level detection remains largely unexplored. Recent work examines inconsistencies between retrieved information and the internal knowledge of LLMs (Su et al., 2024; Jin et al., 2024; Xie et al., 2024a; Wang et al., 2025), but often relies on synthetic edits or focuses solely on temporal drift (Marjanovic et al., 2024). Our task differs from traditional knowledge-intensive settings (Petroni et al., 2021), such as question an-

[1] Dataset and code are available at https://github.com/stanford-oval/inconsistency-detection.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 34828–34855
November 4-9, 2025

swering (Kwiatkowski et al., 2019; Joshi et al., 2017) and fact verification (Thorne et al., 2018a; Jiang et al., 2020), which typically assume corpus consistency: finding a single supporting or refuting evidence item is sufficient. This assumption breaks down when the corpus itself contains contradictions. Figure 1 illustrates this distinction using an example from the FEVEROUS dataset (Aly et al., 2021a) and contrasts it with how CLAIRE addresses the same case.

We propose CLAIRE (Corpus-Level Assistant for Inconsistency REcognition), a system for surfacing inconsistencies in large corpora. To support non-expert users, CLAIRE finds and displays not only candidate contradictions but also disambiguating context and explanations of specialized terminology. It features an interactive interface implemented as a browser extension that surfaces potential inconsistencies to Wikipedia visitors. In a user study with eight experienced editors, participants identified 64.7% more inconsistencies within the same amount of time when using CLAIRE than when using search engines.

We provide the first lower bound on the inconsistency rate in the English Wikipedia and in two widely used Wikipedia-based NLP benchmarks. Through manual verification of CLAIRE outputs, we estimate that approximately 3.3% of facts in the English Wikipedia contradict other statements in the corpus. To our knowledge, this is the first systematic attempt to quantify corpus-level inconsistencies in Wikipedia. In AmbigQA (Min et al., 2020), 4.0% of questions have answers that contradict other content in the same Wikipedia dump, challenging the dataset's assumption of unambiguous, unique answers. In FEVEROUS, 7.3% of claims labeled as `Supports` are contradicted by other evidence within Wikipedia, undermining the standard assumption of corpus consistency in fact verification.

We introduce WIKICOLLIDE, a dataset for Corpus-Level Inconsistency Detection on Wikipedia. WIKICOLLIDE contains inconsistencies identified in the English Wikipedia. Unlike synthetic datasets, it captures genuine ambiguities and complex factual relationships arising in real-world content. To ensure meaningful coverage, we target Wikipedia's Level 5 Vital Articles,[2] which are prioritized for improvement by WikiProject Vital Articles and serve as a centralized watchlist of important entries. Is finding inconsistencies in these pages like finding needles in a haystack? With CLAIRE-assisted curation, WIKICOLLIDE comprises 955 facts, 34.7% of which are inconsistent.

We evaluate CLAIRE and establish strong baselines on WIKICOLLIDE. On the WIKICOLLIDE test set, CLAIRE achieves an AUROC of 75.1%, outperforming baselines while leaving substantial headroom for future work.

2 Related Work

Fact Verification. Recent advances in fact verification increasingly leverage LLMs (Luu et al., 2024; Jayaweera et al., 2024), often within retrieval-augmented generation (RAG) frameworks (Malviya and Katsigiannis, 2024; Rothermel et al., 2024; Chern et al., 2023; Xie et al., 2024b). Many systems extend fact verification to large text corpora (Schuster et al., 2022) by retrieving relevant passages (Khattab and Zaharia, 2020) and using language models to assess whether claims align with the retrieved content.

A wide range of Wikipedia-based fact verification datasets has been developed, including FEVER (Thorne et al., 2018b), FEVEROUS (Aly et al., 2021b), TabFact (Chen et al., 2020), HOVER (Jiang et al., 2020), WikiFactCheck-English (Sathe et al., 2020), VitaminC (Schuster et al., 2021), EX-FEVER (Ma et al., 2024), and AveriTeC (Schlichtkrull et al., 2024). These datasets typically create the `Refutes` class by synthetically modifying true statements, whereas our dataset captures contradictions naturally present in the corpus. WikiContradict (Hou et al., 2024) also targets real contradictions but relies on inconsistency tags added by Wikipedia editors. Our analysis shows that many tagged cases have since been resolved, reducing the accuracy of those labels. Moreover, WikiContradict does not explicitly include a corpus-level `Supports` class— facts that are extensively checked to be free of corpus-level inconsistencies. WikiContradiction (Hsu et al., 2021) focuses on contradictions within a single article, whereas our WIKICOLLIDE extends the scope to contradictions across the entire corpus. This corpus-level setting introduces the additional challenge of searching for and aggregating evidence across

[2] https://en.wikipedia.org/wiki/Wikipedia:Vital_articles/Level/5

multiple articles.

Claim Decomposition. The CLID task requires decomposing the corpus into smaller, self-contained facts. Prior work has examined claim extraction and decomposition within fact verification systems (Hu et al., 2024; Wührl and Klinger, 2024; Min et al., 2023a; Song et al., 2024; Cattan et al., 2024; Gunjal and Durrett, 2024; Pham et al., 2025).

3 Corpus-Level Inconsistency Detection (CLID)

We define CLID as a binary classification task over atomic facts. An atomic fact (Min et al., 2023b) is a short, self-contained statement that conveys a single piece of information and can be verified independently (Semnani et al., 2023; Gunjal and Durrett, 2024). An atomic fact from a corpus is *corpus-level inconsistent* if there exists at least one other piece of information within the corpus that contradicts it; otherwise, it is consistent.

Formally, consider a corpus of documents $C = D_1, D_2, \ldots, D_n$. Let f be an atomic fact extracted from some document $D_i \in C$. The objective is to determine whether there exists a subset of documents $E \subseteq C$ containing evidence that contradicts f. We define the function $\text{CLID}(C, f) \mapsto \{\text{True}, \text{False}\}$ as:

$$\text{CLID}(C, f) = \begin{cases} \text{True}, & \text{if } \exists E \subseteq C \text{ such that} \\ & \text{NLI}(E, f) = \texttt{Refutes} \\ \text{False}, & \text{otherwise} \end{cases}$$

where

$$\text{NLI}(E, f) \in \left\{ \begin{array}{l} \texttt{Supports}, \texttt{Refutes}, \\ \texttt{Not Enough Information} \end{array} \right\}$$

denotes the standard three-way Natural Language Inference task (Bowman et al., 2015; Condoravdi et al., 2003).

CLID is closely related to fact verification (Thorne et al., 2018a; Aly et al., 2021a) but differs in a critical assumption. Fact verification typically presumes that the corpus is internally consistent; the goal is therefore to find *any* evidence supporting or refuting a given claim, i.e., $\exists E$ such that $\text{NLI}(E, f) \in$ `Supports`, `Refutes`. In contrast, CLID requires either identifying at least one piece of refuting evidence or *exhaustively* verifying that no contradictory evidence exists anywhere in the corpus. Figure 1 illustrates this distinction with an example from FEVEROUS.

4 CLAIRE: A Human-in-the-Loop Assistant for Corpus-Level Inconsistency Detection

The CLID task involves two primary subtasks:

1. **Research:** Gathering a comprehensive set of relevant evidence from a large corpus, as exhaustive manual checking is infeasible.
2. **Verification:** Determining whether any retrieved evidence contradicts the given fact.

While humans generally perform well at verifying inconsistencies, our preliminary studies suggest they struggle to efficiently locate relevant pages that may contradict a given fact. To leverage the strengths of both humans and machines, we propose CLAIRE, an agent based on the ReAct architecture (Yao et al., 2023). In this framework, research and verification steps are interleaved, allowing insights gained during verification to guide subsequent retrieval. This iterative process improves the agent's ability to uncover inconsistencies.

In our experiments, we found that simply presenting retrieval results can confuse users unfamiliar with the domain of the claim. Determining consistency often hinges on nuanced understandings of entities and concepts mentioned in the evidence. We therefore introduce two auxiliary actions to the research subtask and incorporate their outcome into the agent's outputs:

1. `clarify`: Request clarifications to disambiguate entities. To distinguish similarly named entities, the agent identifies ambiguities in the given fact and retrieved evidence, gathers additional context, and produces concise summaries highlighting key differences.
2. `explain`: Request explanations of specialized terminology. When encountering unfamiliar concepts (e.g., "tie-break rules in tennis"), the agent queries an LLM for a brief, accessible explanation.

This structure enables more targeted evidence collection, especially for complex or nuanced claims. Illustrative examples of the benefit of such retrieval appear in Appendix C.2. Implementation details are provided in Appendix E.1.

CLAIRE employs in-context learning with an LLM to assess whether retrieved evidence contradicts the given fact. Preliminary evaluations indicate that current LLMs alone do not reliably verify

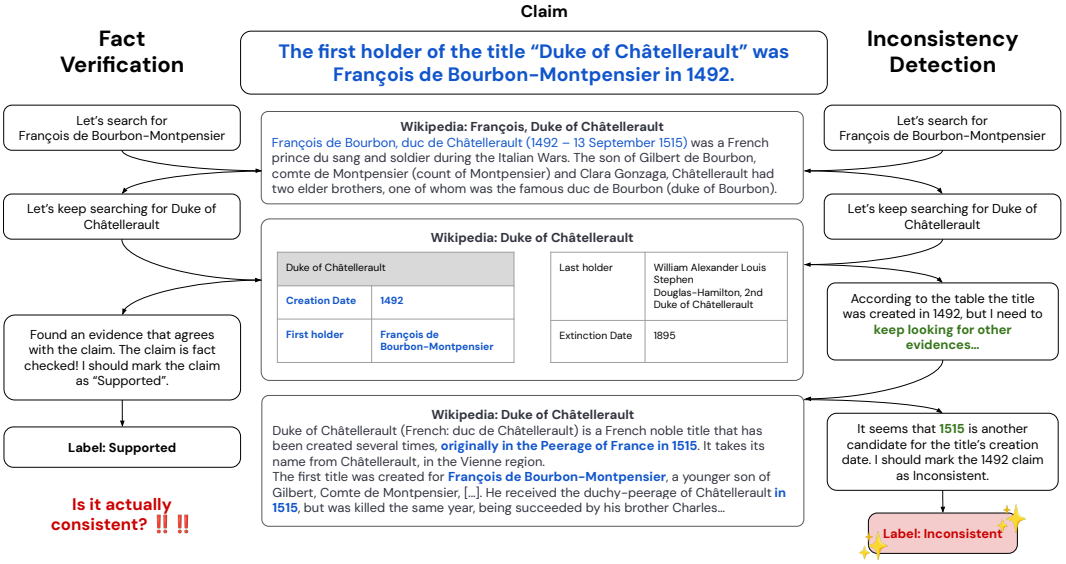

Figure 1: **An example from the FEVEROUS dataset illustrating the difference between fact verification and inconsistency detection.** The claim is shortened for brevity. François de Bourbon-Montpensier was born in 1492 and received the title "duchy-peerage of Châtellerault" in 1515. However, the Wikipedia table "Duke of Châtellerault" incorrectly states that the title was created 23 years earlier. In fact verification, the corpus is assumed to be internally consistent, so the search may stop after finding one supporting piece of evidence. In inconsistency detection, the search continues to identify *any* contradictory evidence within the corpus.

inconsistencies at high accuracy. Therefore, we design CLAIRE to output an inconsistency score in the range $[0, 1]$ to quantify confidence and help users prioritize high-confidence candidates for inspection.

4.1 User Study

We evaluated the effectiveness of CLAIRE in helping users efficiently explore potential inconsistencies by conducting a user study with eight experienced Wikipedia editors (median number of edits: 2,124).

We integrated CLAIRE into a browser extension that highlights potentially inconsistent claims encountered during Wikipedia browsing and editing (Figure 13). The extension analyzes the current page in the background and, when a potential inconsistency is detected, highlights the claim and provides a tooltip with explanations and links to supporting evidence.

For each editor, we selected two Wikipedia articles from a pool of 10 that we had manually verified to contain multiple inconsistencies with the rest of Wikipedia. Each participant completed two 30-minute tasks in randomized order: (1) identifying inconsistencies in one article using our extension without external search, and (2) identifying inconsistencies in a different article without the extension, using any external tools they wish to use (including search engines and LLM chatbots). In both tasks, participants documented all inconsistencies they found within the assigned article.

Participants identified an average of 64.7% more inconsistencies per hour when using CLAIRE.

After the tasks, we collected feedback on perceived usefulness. Participants rated their agreement with several statements on a 5-point Likert scale (*Strongly Disagree* to *Strongly Agree*); Figure 2 shows the response distribution. Editors particularly valued the tool's ability to surface contradictions across article boundaries, information that typically requires extensive manual cross-referencing. Additionally, 87.5% of participants reported increased confidence in identifying inconsistencies when using CLAIRE. These results suggest that AI-assisted inconsistency detection can effectively augment human curation.

Additional details and open-ended responses are provided in Appendix D.

5 Inconsistency Rates in the English Wikipedia and NLP Datasets

We find that **at least 3.3% of Wikipedia facts are inconsistent.** We establish a statistical lower bound on inconsistencies in the November 1, 2024

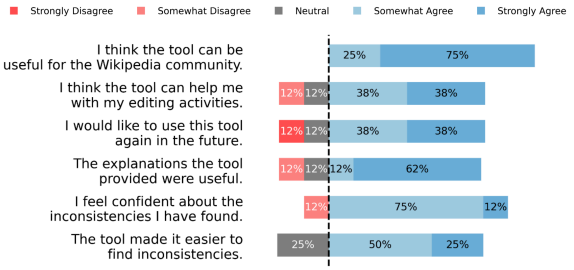

Figure 2: Survey results on the perceived usefulness of our tool ($n = 8$). Responses were collected using a 5-point Likert scale.

Wikipedia dump. Applying CLAIRE to 700 atomic facts uniformly sampled from Wikipedia articles, we identified 44 potentially inconsistent facts, of which 23 were manually confirmed inconsistent. With 99% confidence, we estimate that approximately $3.3\% \pm 1.7\%[1.6\%, 5.0\%]$ of all facts in Wikipedia contradict other information in the corpus. This is a lower bound, as CLAIRE may miss inconsistencies (see Appendix A for further details). Extrapolated to the entire encyclopedia, this corresponds to between 37.6 million and 121.9 million inconsistent facts,[3] underscoring the need for systematic inconsistency detection.

Inconsistency rates vary across article categories. We further analyze frequency by mapping our uniform sample to Wikipedia article categories. Reliability varies substantially across domains, with narrative-heavy subjects particularly prone to inconsistencies. Articles in the history category exhibit the highest inconsistency rate (17.7%), followed by Everyday Life (16.9%) and Society & Social Sciences (14.3%) (Figure 5). The most common error type in history articles is numerical discrepancy. By contrast, categories requiring precise technical knowledge and quantifiable information—such as Mathematics (5.6%) and Technology (9.4%)—show markedly lower rates. See Appendix A.1 for more details.

In the AmbigQA dataset (Min et al., 2020), we find that $4.0\% \pm 1.1\%$ of examples contradict information elsewhere in the corresponding Wikipedia dump, reflecting underlying inconsistencies in Wikipedia. This finding is significant given that AmbigQA is designed to have unique answers at the corpus level. For our investigation, we converted its question-answer pairs into declarative facts and applied the same methodology as before to assess inconsistency.

Applying the same analysis to FEVEROUS (Aly et al., 2021a), we find that $7.3\% \pm 0.5\%$ of claims labeled as `Supports` are involved in corpus-level inconsistencies: their verification outcome depends on which Wikipedia article is chosen as evidence and could have been labeled as `Refutes` instead. This challenges the foundational assumption in fact verification that the corpus provides a consistent source of truth.

6 WIKICOLLIDE: A Dataset for Corpus-Level Inconsistency Detection

With the help of CLAIRE, we create the WIKICOLLIDE dataset, consisting of 955 atomic facts drawn from Wikipedia, each manually labeled as either consistent or inconsistent with the corpus. Whereas prior fact verification datasets often rely on synthetic contradictions that fail to capture real-world nuance, WIKICOLLIDE contains *real, previously unknown* inconsistencies in Wikipedia.

For inconsistent facts, we provide manually verified evidence documents demonstrating the contradiction, detailed reasoning explaining the inconsistency, and a categorization of inconsistency type. For consistent facts, we provide up to 40 evidence passages from Wikipedia that were reviewed during annotation. Note that inconsistent labels represent a gold standard backed by concrete contradictory evidence, whereas consistent labels represent strong verification as exhaustively proving the absence of contradictions across a large corpus is infeasible.

The dataset highlights nuanced challenges such as implicit contradictions, temporal conflicts, and divergent interpretations that might otherwise go undetected. It covers diverse topics including people, history, geography, and science (Figure 11), ensuring broad applicability across domains. Figure 12 shows representative examples from WIKICOLLIDE, illustrating the need for multi-hop reasoning, numerical calculations, nuanced context interpretation, entity disambiguation, and domain expertise.

Split	Inconsistent	Consistent	Total
Validation	135 (28.3%)	342 (71.7%)	477
Test	196 (41.0%)	282 (59.0%)	478
Total	**331 (34.7%)**	**624 (65.3%)**	**955**

Table 1: Distribution of consistent and inconsistent facts in WIKICOLLIDE.

[3] https://en.wikipedia.org/wiki/Wikipedia:Size_of_Wikipedia

6.1 Dataset Construction

Constructing a corpus-level inconsistency dataset poses significant challenges. Given the rarity of inconsistencies, how can we efficiently identify sufficient and representative examples? Moreover, accurate annotation is difficult for both humans and machines. To address these challenges, we adopt a human-in-the-loop approach with CLAIRE. Figure 6 in Appendix B summarizes the overall process.

Knowledge Corpus. Because Wikipedia changes frequently, we use a frozen snapshot from November 1, 2024 for dataset construction and experiments to ensure reproducibility.

Selection of Facts for the Dataset. We select facts through a three-step procedure:

1. **Sampling Popular Wikipedia Pages.** To ensure broad domain coverage, we sample from Wikipedia's Level 5 Vital Articles. These 50,000 articles are actively maintained by WikiProject Vital Articles and represent diverse topics and quality levels.[4] We extract text blocks delimited by newlines, filtering out passages that are too short (<100 characters) or too verbose (>320 characters) to maintain focused, verifiable content. From this filtered pool, we randomly sample 10,000 blocks while preserving the original category distribution.

2. **Fact Extraction.** Following prior work (Semnani et al., 2023), we use GPT-4o (prompt in Figure 14) to split each text block into atomic facts, yielding 89,300 atomic facts.

3. **Increasing the Proportion of Potentially Inconsistent Facts.** To obtain a relatively balanced dataset under high annotation cost, we prioritize facts more likely to be inconsistent. We apply a simple retrieval and LLM-based filtering method with high recall (Appendix B.1). Facts for which no relevant contradictory information is retrieved are filtered out, reducing the candidate set to 1,880 facts.

Human-in-the-loop Annotation. Annotation is performed by the authors and a small group of high-quality crowdworkers recruited via Prolific (Prolific, 2024). Annotators first verify that extracted facts faithfully reflect their source paragraphs, reducing the candidate set to 955 facts.

An annotation interface presents the findings of CLAIRE to annotators: (1) relevant documents from the corpus, (2) clarifications on ambiguous entities (e.g., people with identical names) and unfamiliar concepts, and (3) two-sided reasoning with both consistent and inconsistent interpretations of the gathered evidence (Appendix B.1). Annotators review this information to determine consistency and provide reasoning with citations. For each fact labeled consistent, annotators reviewed an average of 21 potential evidence passages.

Final Dataset. The annotation effort yields 955 facts, of which 34.7% are inconsistent. Facts are evenly split between validation and test sets in WIKICOLLIDE, with consistent and inconsistent labels randomly distributed. Table 1 summarizes the dataset statistics.

6.2 Analysis of WIKICOLLIDE

We analyze the dataset to understand the sources and types of inconsistencies that appear in Wikipedia. We categorize inconsistencies into seven types; Table 2 reports their proportions.

Numerical discrepancies constitute 54.7% of inconsistencies. Of these, 42% are off-by-one errors, often involving dates or years in historical contexts; the remainder are more substantial and varied. Logical contradictions account for 17.5%, with a smaller fraction requiring inference or indirect reasoning. The remaining 27.8% arise from differing definitions, temporal or spatial conflicts, entity disambiguation errors, and divergent categorizations.

7 Evaluating Automatic Corpus-Level Inconsistency Detectors using WIKICOLLIDE

With the dataset annotated, we evaluate multiple automated systems for CLID.

7.1 Evaluated Systems

CLAIRE. We evaluate CLAIRE directly against the human-corrected labels.

Retrieve-and-Verify. Following established fact-checking methodologies (Thorne et al.,

[4] https://en.wikipedia.org/wiki/Wikipedia:Content_assessment

Inconsistency Type	Description	%
Numerical	Inconsistencies in numerical data, such as quantities, measurements, or percentages	**54.7**
Off-by-One Numerical	Small discrepancy involving a margin of one unit	23.0
Clear Numerical	Significant difference that *cannot* be explained by a margin of one unit	31.7
Logical	The claim and evidence directly or indirectly contradict each other	**17.5**
Direct Logical	Clear negation or alternative to a unique fact	14.8
Indirect Logical	Contradiction inferred or indirectly implied	2.7
Definition	Different definitions or interpretations for the same term or concept	**10.6**
Temporal	Inconsistencies in dates, durations, or event sequences	**7.9**
Named Entity	Inconsistencies identifying specific entities (people, organizations, locations)	**6.0**
Categorical	Differences in categorizing entities, objects, or concepts	**2.1**
Spatial	Inconsistencies in spatial descriptions or geographical information	**1.2**

Table 2: Breakdown of inconsistency types in WIKICOLLIDE validation and test sets (331 inconsistent facts).

2018a), this system separates retrieval and verification. First, relevant passages are retrieved via similarity search. Then, a verification model (a single LLM call) assesses the consistency of the fact against all retrieved evidence and outputs an inconsistency score in [0, 1]. Facts with scores above 0.5 are classified as inconsistent.

NLI Pipeline. This system also follows a retrieve-and-verify approach but evaluates each retrieved passage individually against the fact using an LLM-based Natural Language Inference (NLI) model. Each evidence-fact pair is classified as `Refutes`, `Supports` or `Not Enough Information`. A fact is marked inconsistent if at least one passage is classified as a contradiction.

7.2 Experiment Setup

We experiment with GPT-4o (`gpt-4o-2024-11-20`), the 70B-parameter LLaMA-3.1 (Grattafiori et al., 2024), and `o3-mini` (OpenAI, 2025) as LLM backbones. For retrieval, we embed all Wikipedia passages, tables, and infoboxes using the mGTE embedding model (Zhang et al., 2024b). Unless noted otherwise, all experiments use RankGPT (Sun et al., 2023) for reranking after retrieval.

The CLAIRE agent is allotted 10 steps, with 15 passages retrieved per query. For the retrieve-and-verify and NLI pipeline systems, we retrieve 20 passages per query, yielding a comparable total number of evidence items across methods for fair comparison. Ablation studies on these hyperparameters are provided in Section 7.6. Further implementation details and prompts for each system are provided in Appendix E.

7.3 Evaluation Metrics

A primary use case of CLID systems is flagging potential inconsistencies for human review. False positives waste human effort, while false negatives miss true inconsistencies. Therefore, we report the Area Under the Receiver Operating Characteristic curve (AUROC) as our main metric, alongside accuracy and F1.

For retrieve-and-verify and CLAIRE, we vary the inconsistency score threshold to compute ROC curves. For the NLI pipeline, we vary the number of contradictory passages required to classify a fact as inconsistent.

7.4 Results

Table 3 reports performance using GPT-4o as the LLM backbone on the WIKICOLLIDE validation and test sets. CLAIRE achieves the best validation performance across all metrics. On the test set, CLAIRE outperforms other systems in Accuracy and AUROC by at least 0.3 and 2.1 points, respectively.

7.5 Error Analysis

All evaluated systems frequently conflate distinct entities that share the same name, leading to incorrect inconsistency flags.

A key challenge is context-dependent false positives: systems often detect discrepancies between a fact and retrieved evidence but misunderstand cases where those discrepancies are contextually acceptable. Below we detail cases where CLAIRE superficially and incorrectly flags inconsistencies due to limited contextual understanding:

Numerical context. Minor differences due to acceptable rounding or precision should not be flagged as inconsistent.

System	Accuracy	F1	AUROC
Validation set			
CLAIRE	**76.5**	**67.4**	**80.9**
Retrieve-and-verify	73.6	65.2	78.5
NLI-based pipeline	74.0	66.5	78.4
Test set			
CLAIRE	**69.3**	69.6	**75.1**
Retrieve and verify	69.0	69.7	73.0
NLI pipeline	67.0	**70.2**	72.2

Table 3: Overall performance of different systems using GPT-4o on the WIKICOLLIDE validation and test sets. The best score for each metric and split is shown in bold. For validation, we use a fixed threshold of 0.5. For test, thresholds are chosen to maximize validation F1: 0.6 for retrieve-and-verify, 0.5 for CLAIRE, and 1 contradictory passage for the NLI pipeline.

Language context. Articles sometimes include non-English terms whose translated forms differ for named entities; such translation variants should not be treated as inconsistencies. For example, the Japanese album title "DoriMusu 1" and its English equivalent "Dreams 1" are acceptable variants of the same entity name.

Temporal context. The system sometimes compares facts from different time periods and incorrectly flags inconsistencies when atomic facts lack explicit temporal qualifiers.

System	RR	Acc.	F1	AUROC
Retrieve+verify	✗	71.9	62.8	76.5
	✓	73.6 (+1.7)	65.2 (+2.4)	78.5 (+2.0)
CLAIRE	✗	74.6	64.9	78.1
	✓	76.5 (+1.9)	67.4 (+2.5)	80.9 (+2.8)

Table 4: Ablation study showing the impact of reranking (RR) on system performance using GPT-4o on the WIKICOLLIDE validation set. Green values indicate improvements when reranking is applied. All systems use the same configurations as in Table 3.

Perspective and belief context. The system occasionally fails to distinguish differences in viewpoint, belief versus truth, or intention versus action. For example, it may incorrectly flag "Alice believes Earth is flat" as inconsistent with "Bob believes Earth is round."

Legitimate variation in scholarly interpretation. Apparent contradictions about historical events or scientific classifications may reflect evolving consensus rather than true inconsistencies.

7.6 Ablation Studies

Impact of tools available to the CLAIRE agent. Figure 3 shows that the agent achieves the highest F1 when using both `explain` and `clarify`.

Impact of hyperparameters. We vary (1) the number of thought-action-observation steps and (2) the number of documents returned per query. Figure 4 shows that CLAIRE generally performs better with more retrieved documents, but the effect is within 3%.

Impact of reranking in retrieval. Embedding-based retrieval may miss deeper semantic relations. Adding a context-aware reranker prioritizes semantically relevant documents. As shown in Table 4, RankGPT reranking consistently improves all systems and metrics.

System	Model	Accuracy	F1	AUROC
Retrieve and Verify	GPT-4o	73.6	65.2	78.5
	o3-mini	75.7	65.7	77.0
	Llama-3.1-70B	67.9	52.3	70.9
NLI Pipeline	GPT-4o	74.0	66.5	78.4
	o3-mini	65.4	59.5	77.0
	Llama-3.1-70B	63.1	53.9	65.6
CLAIRE (ours)	GPT-4o	**76.5**	**67.4**	**80.9**
	o3-mini	76.3	54.6	68.1
	Llama-3.1-70B	69.0	43.9	69.5

Table 5: Ablation study of different LLMs on the WIKICOLLIDE validation set. The best score for each metric is shown in bold.

Impact of the LLM used. We compare GPT-4o, o3-mini (medium reasoning), and Llama-3.1-70B on the validation set. As shown in Table 5, GPT-4o consistently achieves the highest scores. o3-mini is competitive, with generally higher precision; however, using the same prompt, it rarely outputs inconsistency scores in the intermediate range (0.1–0.9), instead clustering at extremes. Llama-3.1-70B underperforms relative to the other two.

8 Conclusion

We introduce Corpus-Level Inconsistency Detection (CLID), addressing the challenge of identifying contradictory information within large knowledge repositories. To tackle this problem, we present CLAIRE, an agent-based system that combines retrieval with LLM reasoning to detect and contextualize potential contradictions for human review.

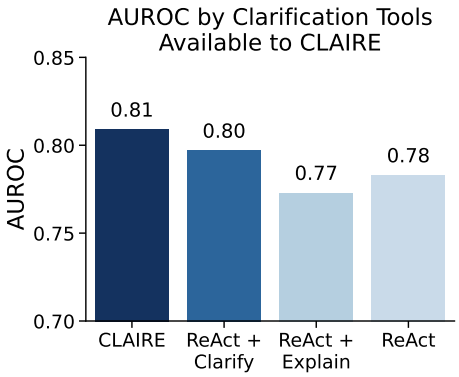

Figure 3: Ablation of tools available to the CLAIRE agent (GPT-4o) on the WIKICOLLIDE validation set. ReAct + Clarify and ReAct + Explain denote the ReAct agent with only one tool enabled.

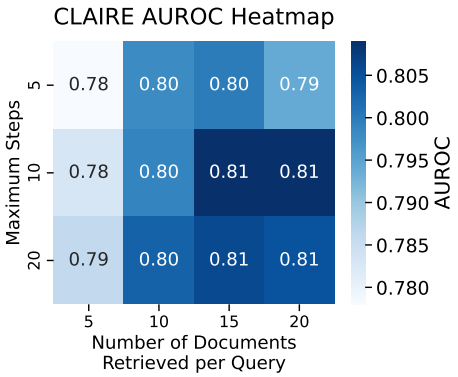

Figure 4: Ablation of retrieval hyperparameters for CLAIRE. Heatmap of AUROC as the number of steps and retrieved documents vary on the WIKICOLLIDE validation set.

We also release WIKICOLLIDE, a benchmark capturing real inconsistencies that synthetic datasets often miss. Our experiments show that, while retrieval and verification are challenging, CLAIRE enables humans to uncover substantially more inconsistencies. Applied to Wikipedia, this framework reveals that approximately 3.3% of facts conflict with other information in the corpus, amounting to millions of contradictory statements across the encyclopedia.

These results demonstrate that corpus-level inconsistencies are a measurable phenomenon in large-scale knowledge corpora. Although automated systems still exhibit systematic errors, they can aid in maintaining knowledge consistency at scale. More broadly, this work suggests a virtuous cycle: LLMs help curate cleaner, more reliable corpora, which in turn improve both human knowledge access and the AI systems built on top of them.

Limitations

This paper focuses exclusively on Wikipedia, the largest open text corpus. As a result, we do not explore other potentially valuable applications of corpus-level inconsistency detection, such as technical texts (e.g., academic, medical, or legal documents) or structured data sources like databases and knowledge graphs. We also leave detecting cross-lingual inconsistencies across different language versions of Wikipedia to future work.

Ethical Considerations

We do not anticipate risks or ethical concerns arising from the publication of WIKICOLLIDE.

For crowdsourcing, we compensated annotators per task, with an overall rate of at least $16 per hour. Participants in our user study were compensated at $20 per hour. The study was approved by our institution's IRB, and participants provided informed consent. No personally identifiable information was collected during annotation or the user study. We release WIKICOLLIDE under the Apache 2.0 License, which is compatible with Wikipedia's license.

Regarding computational resources, we used a CPU-based machine to serve the Wikipedia index and relied on commercial LLM APIs, making direct estimation of carbon footprint difficult. As a proxy, the total experimental cost did not exceed $4,000.

Acknowledgment

This work is supported in part by the Verdant Foundation, the Alfred P. Sloan Foundation, Microsoft Azure AI credits, and the NAIRR Pilot program.

References

Rami Aly, Zhijiang Guo, Michael Sejr Schlichtkrull, James Thorne, Andreas Vlachos, Christos Christodoulopoulos, Oana Cocarascu, and Arpit Mittal. 2021a. The fact extraction and VERification over unstructured and structured information (FEVEROUS) shared task. In *Proceedings of the Fourth Workshop on Fact Extraction and VERification (FEVER)*, pages 1–13, Dominican Republic. Association for Computational Linguistics.

Rami Aly, Zhijiang Guo, Michael Sejr Schlichtkrull, James Thorne, Andreas Vlachos, Christos Christodoulopoulos, Oana Cocarascu, and Arpit Mittal. 2021b. FEVEROUS: Fact extraction and

VERification over unstructured and structured information. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*.

Tiago Bianchi. 2024. Total global visitor traffic to wikipedia.org 2024. Accessed: 2025-03-28.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Arie Cattan, Paul Roit, Shiyue Zhang, David Wan, Roee Aharoni, Idan Szpektor, Mohit Bansal, and Ido Dagan. 2024. Localizing factual inconsistencies in attributable text generation.

Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. 2020. Tabfact: A large-scale dataset for table-based fact verification. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

I-Chun Chern, Steffi Chern, Shiqi Chen, Weizhe Yuan, Kehua Feng, Chunting Zhou, Junxian He, Graham Neubig, Pengfei Liu, and 1 others. 2023. Factool: Factuality detection in generative ai–a tool augmented framework for multi-task and multi-domain scenarios. *ArXiv preprint*, abs/2307.13528.

William G. Cochran. 1953. *Sampling Techniques*. John Wiley & Sons, New York.

Cleo Condoravdi, Dick Crouch, Valeria de Paiva, Reinhard Stolle, and Daniel G. Bobrow. 2003. Entailment, intensionality and text understanding. In *Proceedings of the HLT-NAACL 2003 Workshop on Text Meaning*, pages 38–45.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *ArXiv preprint*, abs/2407.21783.

Anisha Gunjal and Greg Durrett. 2024. Molecular facts: Desiderata for decontextualization in LLM fact verification. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 3751–3768, Miami, Florida, USA. Association for Computational Linguistics.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Realm: Retrieval-augmented language model pre-training.

Yufang Hou, Alessandra Pascale, Javier Carnerero-Cano, Tigran T. Tchrakian, Radu Marinescu, Elizabeth Daly, Inkit Padhi, and Prasanna Sattigeri. 2024. Wikicontradict: A benchmark for evaluating llms on real-world knowledge conflicts from wikipedia. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Cheng Hsu, Cheng-Te Li, Diego Saez-Trumper, and Yi-Zhan Hsu. 2021. Wikicontradiction: Detecting self-contradiction articles on wikipedia. *Preprint*, arXiv:2111.08543.

Qisheng Hu, Quanyu Long, and Wenya Wang. 2024. Decomposition dilemmas: Does claim decomposition boost or burden fact-checking performance?

Chathuri Jayaweera, Sangpil Youm, and Bonnie J Dorr. 2024. AMREx: AMR for explainable fact verification. In *Proceedings of the Seventh Fact Extraction and VERification Workshop (FEVER)*, pages 234–244, Miami, Florida, USA. Association for Computational Linguistics.

Yichen Jiang, Shikha Bordia, Zheng Zhong, Charles Dognin, Maneesh Singh, and Mohit Bansal. 2020. HoVer: A dataset for many-hop fact extraction and claim verification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3441–3460, Online. Association for Computational Linguistics.

Zhuoran Jin, Pengfei Cao, Yubo Chen, Kang Liu, Xiaojian Jiang, Jiexin Xu, Li Qiuxia, and Jun Zhao. 2024. Tug-of-war between knowledge: Exploring and resolving knowledge conflicts in retrieval-augmented language models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 16867–16878, Torino, Italia. ELRA and ICCL.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over BERT. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pages 39–48. ACM.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Son T Luu, Hiep Nguyen, Trung Vo, and Le-Minh Nguyen. 2024. Zefav: Boosting large language models for zero-shot fact verification. In *Pacific Rim International Conference on Artificial Intelligence*, pages 288–295. Springer.

Huanhuan Ma, Weizhi Xu, Yifan Wei, Liuji Chen, Liang Wang, Qiang Liu, Shu Wu, and Liang Wang. 2024. EX-FEVER: A dataset for multi-hop explainable fact verification. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 9340–9353, Bangkok, Thailand. Association for Computational Linguistics.

Shrikant Malviya and Stamos Katsigiannis. 2024. Evidence retrieval for fact verification using multi-stage reranking. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 7295–7308, Miami, Florida, USA. Association for Computational Linguistics.

Sara Vera Marjanovic, Haeun Yu, Pepa Atanasova, Maria Maistro, Christina Lioma, and Isabelle Augenstein. 2024. DYNAMICQA: Tracing internal knowledge conflicts in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 14346–14360, Miami, Florida, USA. Association for Computational Linguistics.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023a. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100, Singapore. Association for Computational Linguistics.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023b. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100, Singapore. Association for Computational Linguistics.

Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. AmbigQA: Answering ambiguous open-domain questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5783–5797, Online. Association for Computational Linguistics.

OpenAI. 2024. Hello gpt-4o. https://openai.com/index/hello-gpt-4o/.

OpenAI. 2025. Introducing o3-mini. https://openai.com/index/openai-o3-mini/. Accessed: Jan 2025.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. 2021. KILT: a benchmark for knowledge intensive language tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2523–2544, Online. Association for Computational Linguistics.

Hoang Pham, Thanh-Do Nguyen, and Khac-Hoai Nam Bui. 2025. Verify-in-the-graph: Entity disambiguation enhancement for complex claim verification with interactive graph representation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5181–5197, Albuquerque, New Mexico. Association for Computational Linguistics.

Prolific. 2024. Prolific.

Mark Rothermel, Tobias Braun, Marcus Rohrbach, and Anna Rohrbach. 2024. InFact: A strong baseline for automated fact-checking. In *Proceedings of the Seventh Fact Extraction and VERification Workshop (FEVER)*, pages 108–112, Miami, Florida, USA. Association for Computational Linguistics.

Aalok Sathe, Salar Ather, Tuan Manh Le, Nathan Perry, and Joonsuk Park. 2020. Automated fact-checking of claims from Wikipedia. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 6874–6882, Marseille, France. European Language Resources Association.

Michael Schlichtkrull, Yulong Chen, Chenxi Whitehouse, Zhenyun Deng, Mubashara Akhtar, Rami Aly, Zhijiang Guo, Christos Christodoulopoulos, Oana Cocarascu, Arpit Mittal, James Thorne, and Andreas Vlachos. 2024. The automated verification of textual claims (AVeriTeC) shared task. In *Proceedings of the Seventh Fact Extraction and VERification Workshop (FEVER)*, pages 1–26, Miami, Florida, USA. Association for Computational Linguistics.

Tal Schuster, Sihao Chen, Senaka Buthpitiya, Alex Fabrikant, and Donald Metzler. 2022. Stretching sentence-pair NLI models to reason over long documents and clusters. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 394–412, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Tal Schuster, Adam Fisch, and Regina Barzilay. 2021. Get your vitamin C! robust fact verification with contrastive evidence. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 624–643, Online. Association for Computational Linguistics.

Sina Semnani, Violet Yao, Heidi Zhang, and Monica Lam. 2023. WikiChat: Stopping the hallucination of large language model chatbots by few-shot grounding on Wikipedia. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2387–2413, Singapore. Association for Computational Linguistics.

Yixiao Song, Yekyung Kim, and Mohit Iyyer. 2024. VeriScore: Evaluating the factuality of verifiable claims in long-form text generation. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 9447–9474, Miami, Florida, USA. Association for Computational Linguistics.

Zhaochen Su, Jun Zhang, Xiaoye Qu, Tong Zhu, Yanshu Li, Jiashuo Sun, Juntao Li, Min Zhang, and Yu Cheng. 2024. Conflictbank: A benchmark for evaluating the influence of knowledge conflicts in llms. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. 2023. Is ChatGPT good at search? investigating large language models as re-ranking agents. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14918–14937, Singapore. Association for Computational Linguistics.

The Economist. 2021. Wikipedia is 20, and its reputation has never been higher. *The Economist*. Accessed: 2025-03-28.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018a. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana. Association for Computational Linguistics.

James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2018b. The fact extraction and VERification (FEVER) shared task. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 1–9, Brussels, Belgium. Association for Computational Linguistics.

Han Wang, Archiki Prasad, Elias Stengel-Eskin, and Mohit Bansal. 2025. Retrieval-augmented generation with conflicting evidence. *Preprint*, arXiv:2504.13079.

Amelie Wührl and Roman Klinger. 2024. Self-adaptive paraphrasing and preference learning for improved claim verifiability.

Jian Xie, Kai Zhang, Jiangjie Chen, Renze Lou, and Yu Su. 2024a. Adaptive chameleon or stubborn sloth: Revealing the behavior of large language models in knowledge conflicts. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Yiqing Xie, Wenxuan Zhou, Pradyot Prakash, Di Jin, Yuning Mao, Quintin Fettes, Arya Talebzadeh, Sinong Wang, Han Fang, Carolyn Rose, Daniel Fried, and Hejia Zhang. 2024b. Improving model factuality with fine-grained critique-based evaluator.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Heidi Zhang, Sina Semnani, Farhad Ghassemi, Jialiang Xu, Shicheng Liu, and Monica Lam. 2024a. SPAGHETTI: Open-domain question answering from heterogeneous data sources with retrieval and semantic parsing. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 1663–1678, Bangkok, Thailand. Association for Computational Linguistics.

Xin Zhang, Yanzhao Zhang, Dingkun Long, Wen Xie, Ziqi Dai, Jialong Tang, Huan Lin, Baosong Yang, Pengjun Xie, Fei Huang, Meishan Zhang, Wenjie Li, and Min Zhang. 2024b. mGTE: Generalized long-context text representation and reranking models for multilingual text retrieval. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 1393–1412, Miami, Florida, US. Association for Computational Linguistics.

A Establishing Lower Bounds for Inconsistency Rates

To determine an appropriate sample size for estimating the proportion of inconsistencies in Wikipedia, we compute the minimum number of claims required to achieve a 99% confidence level with a 5% margin of error using the Cochran formula (Cochran, 1953):

$$n = \frac{z^2 \times p(1-p)}{E^2}$$
$$= \frac{2.576^2 \times 0.5 \times 0.5}{0.05^2} = 664$$

where $z = 2.576$ corresponds to a 99% confidence level, $p = 0.5$ assumes maximum variance (yielding the largest required sample size), and $E = 0.05$ is the desired margin of error. This calculation indicates that we must examine at least 664 claims. We apply the same statistical approach to determine sample sizes for our analyses of the AmbigQA and FEVEROUS datasets.

A.1 Inconsistencies Per Wikipedia Article Category

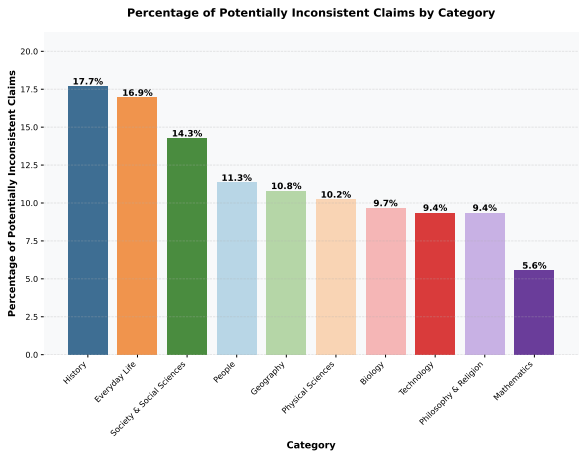

Figure 5: Distribution of inconsistencies in Wikipedia across topics.

The distribution of potentially inconsistent claims across Wikipedia categories reveals notable patterns (Figure 5). History exhibits the highest rate of inconsistencies (17.7%), followed by Everyday Life (16.9%) and Society & Social Sciences (14.3%). These trends are reflected in concrete cases from our analysis. For example, the claim that "The Ottoman Empire first developed the technique of using explosive shells in naval warfare in 1640" is inconsistent with historical records documenting earlier use by other naval powers. Similarly, the assertion that "London's population doubled between 1800 and 1820" oversimplifies gradual demographic change; empirical population estimates for that period do not support a doubling.

In contrast, categories requiring precise technical knowledge, such as Mathematics (5.6%) and Technology (9.4%), show markedly lower inconsistency rates, suggesting that factual precision is better maintained in domains with more quantifiable information. Overall, these results indicate that Wikipedia's reliability varies across knowledge domains, with narrative-heavy subjects being particularly susceptible to inconsistencies.

B More Details on the Annotation Process for WIKICOLLIDE

B.1 Annotation Tool

Figure 6 provides an overview of the dataset construction process.

Filter to Balance Inconsistent Labels in the Dataset. We implement a weak baseline to provide a permissive standard for inconsistency detection and filter out obviously consistent claims during WIKICOLLIDE construction. This baseline is a simplified version of the retrieve-and-verify system described in Section7, where the verifier outputs a binary inconsistency decision rather than a confidence score. We use GPT-4o mini (OpenAI, 2024) as the language model for these binary decisions.

Report Generation. We develop a report generation system that produces a detailed report for each fact in the dataset. This system mirrors the setup in Section 7 but replaces the verifier with a report generation stage. The report stage takes all retrieved evidence and the clarifications made by the tools agent and generates a two-sided analysis via two GPT-4o calls: one soliciting reasoning that the fact is inconsistent and another soliciting reasoning that it is consistent. This provides annotators with balanced information for final judgment. The final report shown to annotators includes both lines of reasoning and the agent's trace. See Figure 8 through 10 for illustrations.

Annotation Portal. We built a web-based annotation platform to streamline the workflow. Annotators first check the extracted fact for any extraction issues, then review the detailed inconsistency analysis report, and finally assign a label of either "consistent" or "inconsistent". Screenshots of the interface are shown in Figure 7 through 10.

B.2 Annotators

The dataset is annotated partly by the authors and partly by annotators recruited via Prolific (Prolific, 2024). To recruit external annotators, we conducted an initial qualification test using 10 randomly sampled facts from a subset previously labeled by the authors. Candidates were evaluated on both labeling accuracy and the quality of their written justifications. We selected the top 17 candidates who demonstrated strong analytical skills

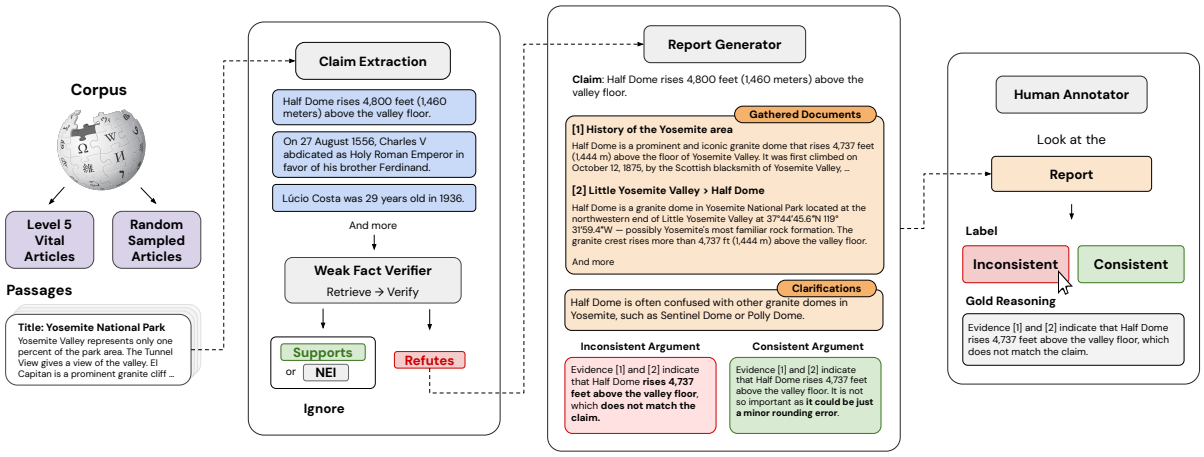

Figure 6: Overview of the WIKICOLLIDE construction process: diverse passage sampling from Wikipedia's Vital Articles, adversarial claims collection using GPT-4o and a weak baseline filter, and human verification with detailed evidence analysis. Randomly sampled articles are used to estimate the prevalence of inconsistencies in the entire English Wikipedia.

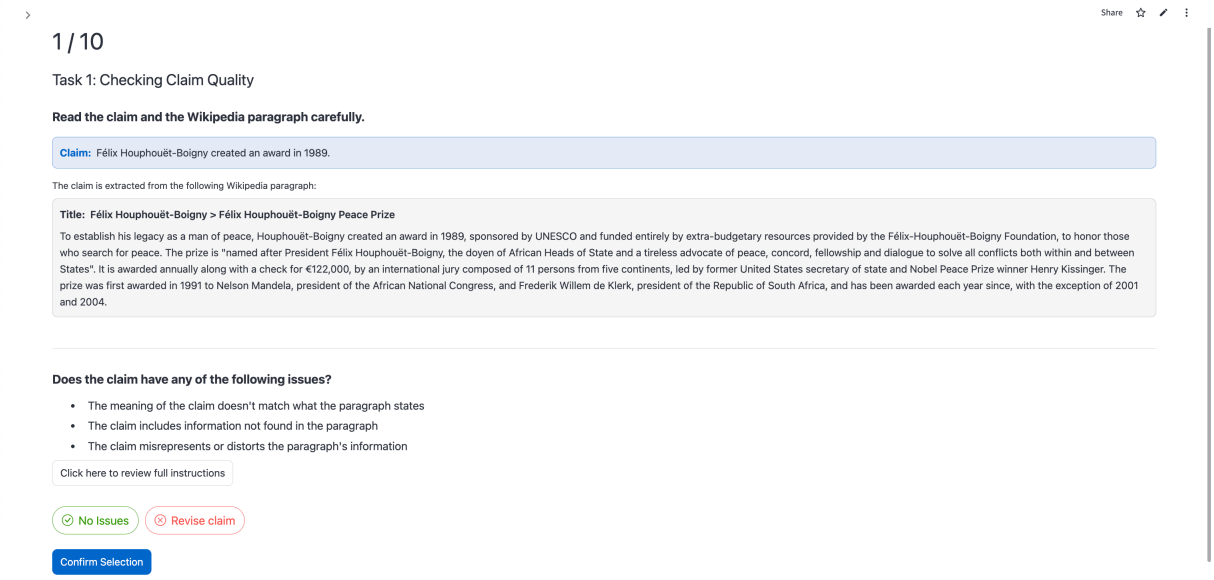

Figure 7: Screenshot 1 of the annotation tool showing the main interface for claim verification and inconsistency labeling.

and high accuracy in identifying inconsistencies. Because inconsistency detection is nuanced, we required annotators to be native English speakers from the US or UK, hold a graduate degree (Masters or PhD), and maintain a Prolific approval rate of at least 95%. The final pool of 17 annotators spent an average of 6.5 minutes evaluating each fact.

B.3 Annotation Guidelines

The task is inherently complex, which increases the risk of labeling errors. Determining whether a claim is inconsistent with a set of evidence is substantially more challenging than many standard annotation tasks. The definition of inconsistency can be context-dependent. For example, if a claim states a population of 4.8 million while the evidence reports 5 million, the case could be labeled "inconsistent" under exact matching or "consistent" if rounding is deemed acceptable. Likewise, comparing an imprecise expression such as "a few years" to a specific value is nontrivial, and the threshold for inconsistency is not always clear.

To mitigate ambiguity, we established explicit guidelines for such cases and instructed annotators to follow them closely.

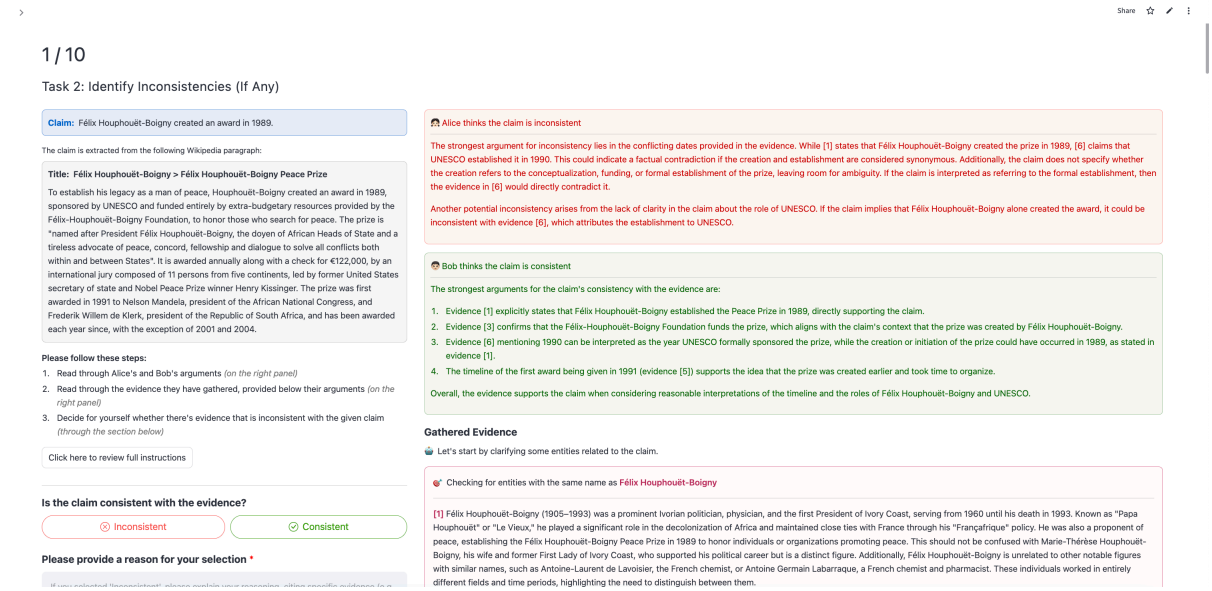

Figure 8: Screenshot 2 of the annotation tool showing the main interface for claim verification and inconsistency labeling.

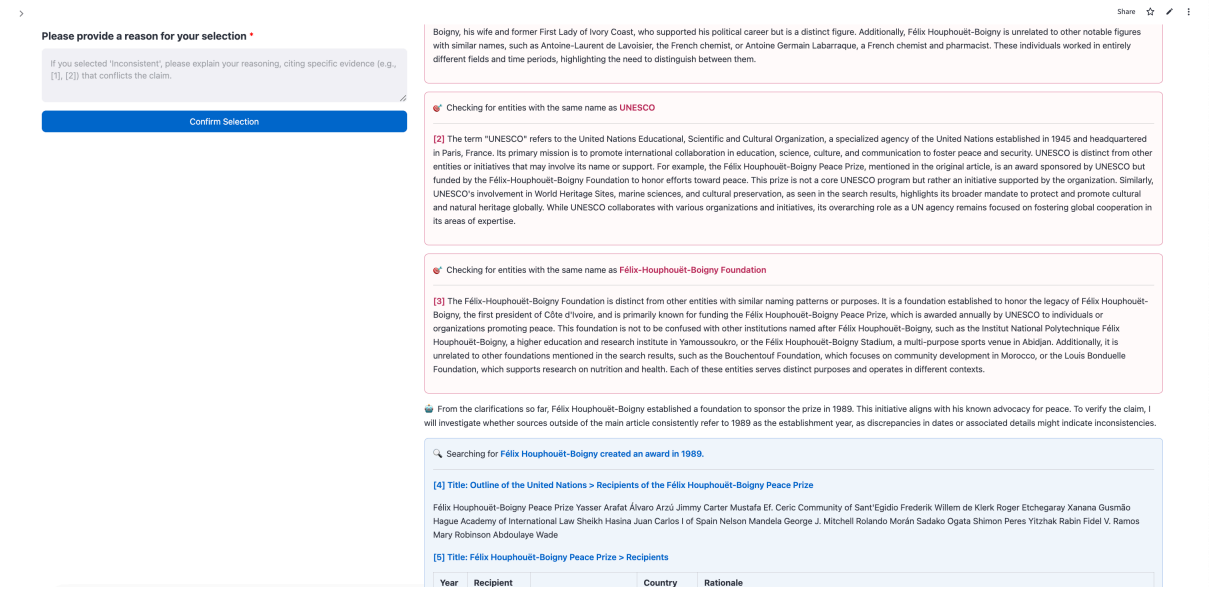

Figure 9: Screenshot 3 of the annotation tool showing the main interface for claim verification and inconsistency labeling.

C Dataset Details

C.1 Distribution of the Topics

Figure 11 shows the distribution of topics covered by the facts.

C.2 Dataset Examples

Figure 12 presents detailed examples of claims with accompanying evidence from the dataset.

D More Details on the User Study

D.1 Browser Extension Implementation

We implement the browser extension using JavaScript for the frontend and Python for the backend server. The frontend is a lightweight Chrome extension that injects content scripts to highlight potentially inconsistent claims on Wikipedia pages and provide a potential explanation for the inconsistency in the form of a side panel. For fact extraction, we use GPT-4o to parse Wikipedia page content into atomic claims, follow-

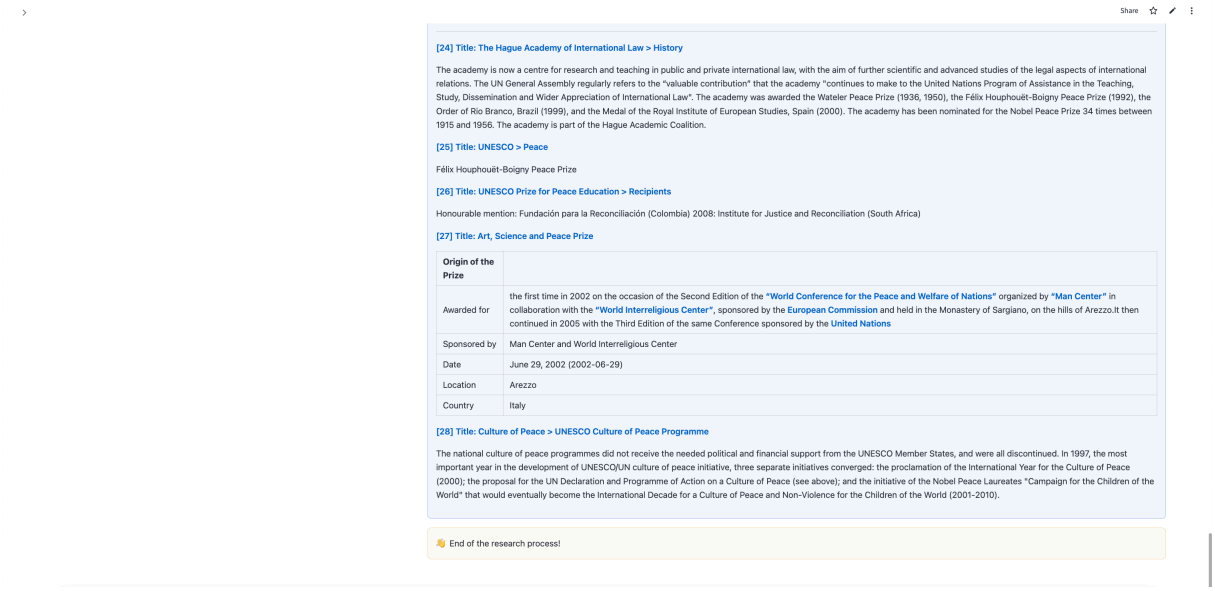

Figure 10: Screenshot 4 of the annotation tool showing the main interface for claim verification and inconsistency labeling.

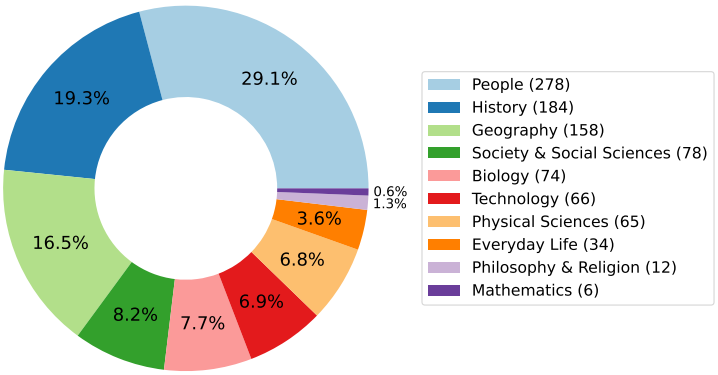

Figure 11: Distribution of topics across the WIKICOLLIDE dataset, showing the diversity of knowledge domains covered.

ing prior work (Semnani et al., 2023; Min et al., 2023b). The extension communicates with the backend via REST API endpoints.

D.2 Recruitment Details

We recruited participants through by posting a research participation call on Meta-Wiki[5]. All recruited editors had made at least 200 edits and been active for over one year. The study protocol was approved by our institution's Institutional Review Board (IRB), and all participants provided informed consent before beginning the study.

D.3 Perceived Usefulness and Qualitative Feecback

In addition to Likert-scale ratings, we collect open-ended feedback about participants' experiences finding inconsistencies with and without our tool. Editors report that they employ diverse manual strategies, such as cross-checking linked articles, search and keyword search, reviewing talk pages, and validating references, but find these approaches time-consuming and cognitively cumbersome. They further express that they value the tool's accessibility for novice editors and utility for verifying AI-generated content. The main concerns center on processing speed, false positive rates, and interface design. These insights reveal a key tradeoff: while our system significantly reduces the editorial burden, its effectiveness ulti-

[5] https://meta.wikimedia.org

Dataset	Examples of Related Evidence	Required Competencies and Reasoning
Claim: Lúcio Costa was 29 years old in 1936. **Title**: Oscar Niemeyer In 1936, at 29, Lúcio Costa was appointed by Education Minister Gustavo Capanema to design the new headquarters of the Ministry of Education and Health in Rio de Janeiro. (...) **Label**: Inconsistent **Type of Inconsistency**: Clear Numerical Discrepancy	[1] **Title**: Lúcio Costa Lúcio Marçal Ferreira Ribeiro Lima Costa (27 February 1902 - 13 June 1998) was a Brazilian architect and urban planner, best known for his plan for Brasília. [2] **Title**: Lúcio Costa > Career (...) Among his major works are also the Ministry of Education and Health, in Rio (1936-43), designed with Niemeyer, Roberto Burle Marx, among others, and consulted by Le Corbusier, and the Pilot Plan of Brasília, a competition winner designed in 1957 and built mostly in 1958-1960. ...	**Calculation** Evidence [1] clearly establishes Costa's birthdate as February 27, 1902, making him 34 years old in 1936. Evidence [2] further supports that this entity is the same person referenced in the claim, as his work information aligns. Therefore, the assertion that he was 29 years old in 1936 is factually inconsistent.
Claim: The first decipherable sentence in the Egyptian language dates to the 28th century BC (Second Dynasty). **Title**: Egyptian hieroglyphs The use of hieroglyphic writing arose from proto-literate symbol systems in the Early Bronze Age c. the 33rd century BC (Naqada III), with the first decipherable sentence written in the Egyptian language dating to the 28th century BC (Second Dynasty). (...) **Label**: Inconsistent **Type of Inconsistency**: Off-by-One Numerical Discrepancy	[1] **Title**: Writing > Egypt (...) The world's oldest deciphered sentence was found on a seal impression found in the tomb of Seth-Peribsen at Abydos, which dates from the Second Dynasty (28th or 27th century BC). (...) [2] **Title**: List of languages by first written account > Before 1000 BC Seal impression from the tomb of Seth-Peribsen, containing the oldest known complete sentence in Egyptian, c. 2690 BC (...) ...	**Multi-hop Reasoning** While neither Evidence [1] nor Evidence [2] individually reveals any inconsistencies, combining the two highlights a contradiction. Evidence [1] informs us that the first decipherable sentence mentioned in the claim appears on a seal impression found in the tomb of Seth-Peribsen. Evidence [2], however, provides the year of origin for this sentence, also based on a seal impression from the same tomb. This specific year contradicts the timeline proposed in the claim.
Claim: The 5-cent fare in 1904 is equivalent to $2 in 2023 dollars. **Title**: New York City Subway (...) Its operation was leased to the Interborough Rapid Transit Company (IRT), and over 150,000 passengers paid the 5-cent fare ($2 in 2023 dollars) to ride it on the first day of operation. **Label**: Not Enough Information	[1] **Title**: New York City transit fares > Token and change From the inauguration of IRT subway services in 1904 until the unified system of 1948 (including predecessor BMT and IND subway services), the fare for a ride on the subway of any length was 5 cents (.05 in 1904 equivalent to 1.7 in 2023; 0.05 in 1948 equivalent to 0.63 in 2023). (...) ...	**Contextual Flexibility** Although there is a discrepancy between the equivalent value of 5 cents from 1904 in 2023, as stated in the claim and the evidence, it is possible that the claim employs a rounding method. Therefore, there is insufficient information to definitively determine an inconsistency.
Claim: Chrysoberyl has the chemical formula Al2BeO4. **Title**: Beryllium Beryllium is found in over 100 minerals, but most are uncommon to rare. The more common beryllium containing minerals include: bertrandite (Be4Si2O7(OH)2), beryl (Al2Be3Si6O18), chrysoberyl (Al2BeO4) and phenakite (Be2SiO4). (...) **Label**: Not Enough Information	[1] **Title**: Chrysoberyl The mineral or gemstone chrysoberyl is an aluminate of beryllium with the formula BeAl2O4. ...	**Domain Expertise** Although there is a discrepancy between the formulas of Chrysoberyl (Al2BeO4 vs. BeAl2O4), both are valid representations based on their oxidation states and molecular structure, where aluminum (Al) and beryllium (Be) form ionic bonds with oxygen (O). In fact, Al2BeO4 is typically used as the alphabetic or published formula, whereas BeAl2O4 is recognized as the standard formula. In contrast, for compounds like sodium nitrate, only the formula NaNO3 is valid. The alternative formula NNaO3 is incorrect because oxygen (O) bonds with nitrogen (N) to form the cohesive polyatomic ion NO3-. This ion then interacts with sodium (Na+) through ionic bonding to create an ionic crystalline structure. Importantly, oxygen (O) does not form a direct bond with sodium (Na) in this arrangement.
Claim: Jonathan Browning was born in 1859. **Title**: John Browning (...) He developed his first rifle, a single-shot falling block action design while he was still his father's apprentice, then, in 1878, in partnership with his younger brother, co-founded John Moses and Matthew Sandefur Browning Company, later renamed Browning Arms Company. The company began producing the brothers' designs and other non-military firearms. By 1882, the company employed John and Matthew's half-brothers Jonathan (1859-1939), Thomas (1860-1943), William (1862-1919), and George (1866-1948). **Label**: Not Enough Information	[1] **Title**: Jonathan Browning Jonathan Browning may refer to: Jonathan Browning (designer), American interior designer and business executive / Jonathan Browning (inventor) (1805-1879), American inventor and gunmaker / Jonathan Browning (UK businessman) (born 1959), president and CEO of Volkswagen Group of America [2] **Title**: John W. Browning John Walker Browning (June 10, 1842 in New York City - 1904) was an American journalist, lawyer and politician from New York. [3] **Title**: John Browning (surveyor) John Samuel Browning (1831 - 24 July 1909), also known as John Spence Browning, was a British-born pioneer surveyor in the South Island of New Zealand. [4] **Title**: Jonathan Browning (inventor) Jonathan Browning (October 22, 1805 - June 21, 1879) was an American inventor and gunsmith. [5] **Title**: Jonathan Browning (inventor) Personal details Born: October 22, 1805 ...: ... Children: 22, including: John M Browning, Matthew S. Browning ...	**Entity Disambiguation** Based on Evidence [1], there are multiple individuals with the name Jonathan or John Browning. Evidence [1]-[3] provide examples of these entities. While the year of birth cited in the claim appears inconsistent with all the years of birth for entities mentioned in Evidence [1]-[3], a closer inspection reveals that most of these entities are distinct from the one referenced in the claim, as they are not related to the gunmaker. Therefore, these discrepancies do not result in inconsistencies. The only Jonathan Browning associated with the Browning Arms Company is Jonathan Browning (the inventor), as identified in Evidence [4]. This evidence also provides a contradictory year of birth. However, further investigation using Evidence [5] reveals that Jonathan Browning (the inventor) is the parent of John M. Browning and Matthew S. Browning. This suggests that Jonathan Browning (the inventor) and the Jonathan Browning mentioned in the claim might be different individuals, as the Jonathan Browning in the claim is described as a half-brother of John M. Browning and Matthew S. Browning, not their father. In fact, the entity referred to in the claim is Jonathan Edmund Browning, the son of Jonathan Browning (the inventor). Therefore, there is insufficient information to conclude an inconsistency, as no evidence mentions the birth year of this specific individual.

Figure 12: Examples of claims with evidence from the dataset and required competencies.

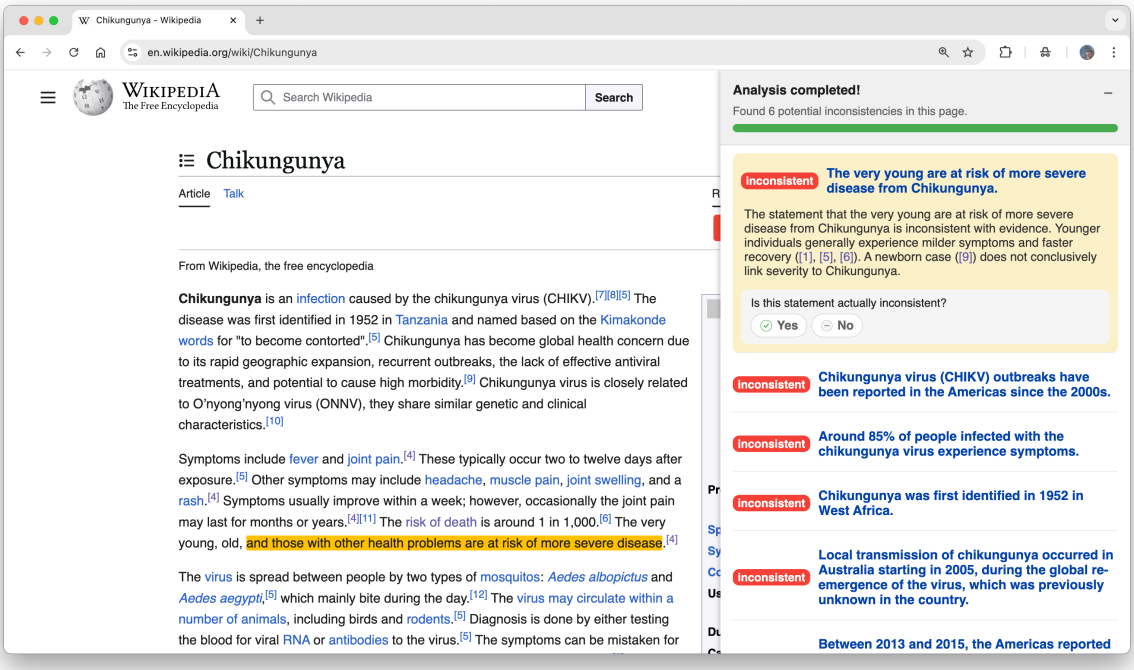

Figure 13: The browser extension is implemented as a button in Wikipedia which the user can click to check for inconsistencies. When a claim is flagged as potentially inconsistent, we highlight the claim on the page and show a side panel with a detailed explanation of the inconsistency and links to the evidence documents.

mately depends on balancing detection sensitivity with efficiency and usability. See Table 6 for examples of user feedback.

E Implementation Details

We accessed OpenAI models via Azure OpenAI and accessed LLaMA models through Azure AI Services. For all experiments, greedy decoding (i.e. temperature 0) is used. All numbers are the result of a single run.

E.1 Implementation of Explain and Clarify Tools

The implementation of clarification tools in our system enables the verification agent to gather additional information when evaluating a claim.

- `explain` prompts the LLM to provide background information about the topic query. As shown in Figure 15, this action instructs the LLM to synthesize its existing knowledge about the topic in relation to the claim being evaluated.
- `clarify` disambiguates entities by first retrieving relevant information and then using the language model to explain differences based on that retrieved information. The number of retrieved documents per each clarify is 10. As shown in Figure 16 and 17, we prompt the language model to analyze the retrieved results to resolve ambiguities by explain the differences.

E.2 Prompts for Systems

We list all the prompts used for implementing the agent system below. Figure 14 is our prompt for extracting atomic facts from Wikipedia articles, whereas Figures 15, 16, 17, 18, 19, 20, 21, and 22 are the tools available to the CLAIRE agent. In each prompt, `# input` and `# output` denote the boundaries of few-shot examples used, if any.

Positive Feedback
The fact that I can simply enable the extension and, within a few minutes, see at a glance 126 inconsistencies is impressive. I also appreciate that it directly links me to where each inconsistency occurs in the article and provides a brief summary. Additionally, I like the feature that allows me to validate these inconsistencies by either accepting or rejecting them.
It gave a lot of potential statements and claims that could be inconsistent with other information. One of the harder parts without using the tool was to find relevant articles that would include overlapping information.
The tool is instructive, directive and straight to the point. It identifies inconsistency without rigorous means.
Very good at finding discrepancies.
That each result has an option to give feedback if the statement is actually inconsistent.
Really helpful automation, especially for editors that may not be a subject-matter expert. Like that it could be used to fact-check AI-generated content if/when it comes to Wikipedia.
It clearly demarked which statements were potentially inconsistent and with what, plus it allows for feedback on if the statements are in fact inconsistent.
I really like the intent of the tool! I feel like it holds a lot of promise to help editors fact-check and correct inaccurate articles more quickly. I also love the UI - it's simple, clear, and easy to interpret. I especially appreciated the explanations offered in the panel, which even linked to helpful sources! That part was really impressive. I was able to detect one very clear inconsistency using this tool, and I was able to identify and validate it extremely quickly because of the helpful explanation and sources that the tool provided.
Areas for Improvement
I didn't like how often the tool was wrong. Even if, on paper, it would be better to highlight claims that the tool is unsure is consistent, in reality I personally felt very annoyed every time it was wrong.
It assumed that Georgian, Renaissance Revival, etc. architecture were mutually exclusive with Palladian architecture, and that all the American buildings listed as having been influenced by Palladian architecture were inconsistent due to the architectural style listed in those articles.
UI/UX improvements mostly. Would be nice if it was a native feature that could be turned on in Wikipedia and did not require downloading onto your computer.
There wasn't much to dislike! I think that the model's precision could be improved, but the experience itself was really great.

Table 6: Qualitative User Feedback on CLAIRE

```
# instruction
You are an expert fact extractor tasked with identifying and listing atomic facts
    from a given text. Your goal is to produce a comprehensive list of facts that
    are explicitly stated or directly inferrable from the provided information.

Instructions:
1. Read the title and text carefully.
2. Extract all atomic facts from the information provided. An atomic fact is a
    single, indivisible piece of information that cannot be broken down further
    without losing its meaning or accuracy.
3. Include only facts that are explicitly stated or can be directly and
    unambiguously inferred from the text.
4. Do not add any external knowledge or assumptions not present in the given
    information.
5. Ensure that each fact is self-contained and can be independently fact-checked.

Before providing your final list of facts, break down your fact extraction
    process in <fact_extraction_process> tags. This will help ensure a thorough
    and accurate extraction of facts.

In your fact extraction process, follow these steps:
1. Identify key topics or themes from the title and text.
2. For each topic/theme, list explicit facts from the text.
3. Consider potential inferences that can be directly drawn from the explicit
    facts, and evaluate their validity.
4. Evaluate each fact (explicit and inferred) for atomicity and self-containment.
5. Categorize facts by topic/theme.
6. Cross-reference each fact with the original text to ensure accuracy.
7. Review the list to ensure no redundant or overlapping facts are included.

After your analysis, provide your final list of facts, with each fact on a new
    line.

Example output structure:

<fact_extraction_process>
[Your detailed fact extraction process, following the steps outlined above]
</fact_extraction_process>

<facts>
[Fact 1]
[Fact 2]
[Fact 3]
...
</facts>

# input
Here is the title and text you need to analyze:

<title>
{{ full_title }}
</title>

<text>
{{ text }}
</text>
```

Figure 14: Fact Extraction Prompt

```
# instruction
You will be given a topic, and a Wikipedia passage where the topic is mentioned.
    Your task is to write a self-contained paragraph explaining technical or
    domain-specific terms in the topic. Your goal is to provide background
    information on the given topic for people who are unfamiliar with it. If a
    term, event or concept in the topic has multiple interpretations or meanings,
    list all plausible ones.

# input
Topic: Infanta Amalia
Wikipedia article: Infanta Amalia of Spain
Infanta Amalia of Spain (Spanish: Amalia de Borbón y Borbón-Dos Sicilias; 12
    October 1834  27 August 1905) was the youngest daughter of Infante Francisco
    de Paula of Spain. Her eldest brother, Francisco de Asís, married Queen
    Isabella II of Spain, who was Amalia's first cousin.

# output
"Infanta Amalia" refers to a title and name in Spanish and Portuguese contexts.
    "Infanta" is a title used in Spain and Portugal for the daughters of a
    monarch who are not heir apparent, similar to "princess" in English. "Amalia"
    is a given name. Therefore, "Infanta Amalia" would refer to a princess named
    Amalia within a royal family in Spain or Portugal.

# input
Topic: The Great Gatsby
Wikipedia article: The Great Gatsby
It was also performed in the summer of 2012 at the Aspen Music Festival and
    School. It was performed at Seagle Festival in Schroon Lake, NY in the summer
    of 2018.

# output
"The Great Gatsby" here likely to a musical adaptation, play, opera, or other
    performance based on the novel "The Great Gatsby" by F. Scott Fitzgerald. The
    novel is a classic work of American literature published in 1925. The
    performances mentioned in the passage are likely adaptations of the novel for
    the stage or other artistic mediums.

# input
Topic: {{ topic }}
Wikipedia article: {{ claim.context_block.full_title }}
{{ claim.context_block.content }}
```

Figure 15: Generate Background Information Prompt

```
# instruction
You will be given an entity and a Wikipedia paragraph where it is mentioned.
You will also be provided with a list of search results that may contain
    information about the entity, and other similar entities.
Your task is to write a self-contained paragraph explaining the differences
    between entities with similar names in the search results.
Entities with similar names might lead to confusion, and the goal here is to
    disambiguate them. Pay attention to the following:

- People with the same last name, but different first names. Or People with the
    same name but different professions or time periods.
- Events with the same name but different years or locations. For example, "The
    Olympics" could refer to the winter or summer games, or games held in
    different years.
- Organizations with similar names but different purposes or locations.
- etc.

# input
Entity: members of the royal family of Spain named Amalia

[1] Title: Infanta María Amalia of Spain
María Amalia, Infanta of Spain (9 January 1779 in Madrid - 22 July 1798 in
    Madrid), was a Spanish princess. She was a daughter of King Charles IV of
    Spain, in 1795, she married her uncle Infante Antonio Pascual of Spain.

[2] Title: Infanta Amalia of Spain > Childhood
She was born at the royal Palace of Madrid on 12 October 1834 as the eleventh
    child and sixth daughter of Infante Francisco de Paula of Spain, younger
    brother of King Fernando VII of Spain, and his wife, Princess Luisa Carlota
    of Bourbon-Two Sicilies. Infanta Amalia's mother was the niece of her father
    since her maternal grandmother, Infanta Maria Isabella of Spain, was the
    elder sister of Infante Francisco de Paula.

[3] Title: Infanta María Amalia of Spain > Early life
Born at the Royal Palace of El Pardo, Maria Amalia was the second surviving
    daughter of King Carlos IV of Spain (1748-1819) and his wife Maria Luisa of
    Parma (1751-1819), a granddaughter of Louis XV of France.

[4] Title: Infanta Amalia of Spain
Infanta Amalia of Spain (Spanish: Amalia de Borbón y Borbón-Dos Sicilias; 12
    October 1834 - 27 August 1905) was the youngest daughter of Infante Francisco
    de Paula of Spain. Her eldest brother, Francisco de Asís married Queen
    Isabella II of Spain, who was Amalia's first cousin. She was one of only two
    of five sisters who made a royal marriage. In 1865 she married Prince
    Adalbert of Bavaria, a son of King Ludwig I of Bavaria. Upon her marriage she
    moved to Munich, where she spent the rest of her life. However she remained
    attached to her native country and was instrumental in arranging the marriage
    of her eldest son Prince Ludwig Ferdinand of Bavaria with her niece Infanta
    Paz of Spain.

[5] Title: Infanta Amalia of Spain > Later life and death
Although Infanta Amalia lived for the rest of her life in Munich, she remained
    attached to her native country. She visited Spain often and her eldest son
    Prince Ludwig Ferdinand of Bavaria was born at the royal palace of Madrid.
    She spent the winters at the residence of Munich and the summers at
    Nymphenburg Palace. Her husband died in 1875; Amalia outlived him by thirty
    years. Amalia maintained her affiliation with Spain in the next generation.
    All of her five children spoke Spanish fluently and she encouraged her son
    Ludwig Ferdinand to marry her niece and goddaughter Infanta Maria de la Paz
    of Spain. The couple married in 1883.
```

Figure 16: Generate Entity Report Prompt

```
# output
There are two entities with similar names.
    1. Infanta Amalia of Spain: Infanta Amalia of Spain (Spanish: Amalia de
       Borbón y Borbón-Dos Sicilias; 12 October 1834 - 27 August 1905) was the
       youngest daughter of Infante Francisco de Paula of Spain.
    2. Infanta María Amalia of Spain: María Amalia, Infanta of Spain (9 January
       1779 in Madrid - 22 July 1798 in Madrid), was a Spanish princess. She was
       a daughter of King Charles IV of Spain, in 1795, she married her uncle
       Infante Antonio Pascual of Spain.


These two individuals seem to be separate entities, but may be relatives.

# input
Entity: Antoine Émile Henry Labeyrie

[1] Title: Antoine Émile Henry Labeyrie
Antoine Émile Henry Labeyrie (born 12 May 1943) is a French astronomer, who held
    the Observational astrophysics chair at the Collège de France between 1991
    and 2014, where he is currently professor emeritus. He is working with the
    Hypertelescope Lise association, which aims to develop an extremely large
    astronomical interferometer with spherical geometry that might theoretically
    show features on Earth-like worlds around other suns, as its president. He is
    a member of the French Academy of Sciences in the Sciences of the Universe
    (sciences de l'univers) section. Between 1995 and 1999 he was director of the
    Haute-Provence Observatory.

[2] Title: Galluis > Notable residents
Antoine-Germain Labarraque (1777 - 1850) was a French chemist and pharmacist,
    notable for formulating and finding important uses for "Eau de Labarraque" or
    "Labarraque\'s solution", a solution of sodium hypochlorite widely used as a
    disinfectant and deodoriser. He died in Gallius on 9 December 1850.

[3] Title: Antoine Lavoisier
Antoine-Laurent de Lavoisier (/lvwzie/ l-VWAH-zee-ay; French: [twan l d
    lavwazje]; 26 August 1743\xa0- 8 May 1794), also Antoine Lavoisier after the
    French Revolution, was a French nobleman and chemist who was central to the
    18th-century chemical revolution and who had a large influence on both the
    history of chemistry and the history of biology.

[4] Title: Antoine Germain Labarraque
Antoine Germain Labarraque (28 March 1777 - 9 December 1850) was a French chemist
    and pharmacist, notable for formulating and finding important uses for "Eau
    de Labarraque" or "Labarraque\'s solution", a solution of sodium hypochlorite
    widely used as a disinfectant and deodoriser.

[5] Title: Antoine Germain Labarraque
| Antoine Germain Labarraque | |
| --- | --- |
| Portrait of Labarraque | |
| Born | (1777-03-28)28 March 1777Oloron-Sainte-Marie, Pyrénées-Atlantiques,
    France |
| Died | 9 December 1850(1850-12-09) (aged\xa073)near Paris, France |
| Nationality | French |
| Education | College of Pharmacy, Paris |
| Occupation(s) | chemist and pharmacist |
| Known\xa0for | using sodium hypochlorite as a disinfectant and deodoriser |
| Parents | * François Labarraque (father) * Christine Sousbielle (mother) |
```

Figure 17: Generate Entity Report Prompt (Continued)

```
# output
There are multiple notable French scientists with similar names beginning with
    Antoine.
  1. Antoine Émile Henry Labeyrie (born 1943) is a French astronomer and
      professor emeritus who held the Observational astrophysics chair at the
      Collège de France.
  2. Antoine-Laurent de Lavoisier (1743-1794) was a French nobleman and chemist
      central to the 18th
  3. Antoine Germain Labarraque (1777-1850) was a French chemist and pharmacist
      known for developing "Labarraque's solution," a sodium hypochlorite
      disinfectant.
While these individuals share similar first names and French nationality, they
    worked in different fields and time periods.

# input
Entity: {{ entity_name }}
Original article: {{ claim.context_block.full_title }}
{{ claim.context_block.content }}

{{ search_results.to_string() }}
```

Figure 18: Generate Entity Report Prompt (Continued)

```
# instruction
Determine if a claim extracted from a Wikipedia paragraph is inconsistent with
    any of the provided documents. A claim is deemed inconsistent when at least
    one document contains information that directly contradicts it. If no such
    contradiction existseven when the documents do not explicitly support the
    claimthe claim is considered consistent.

Step-by-Step Instructions:

1. Identify the Claim
Definition: A brief statement directly extracted from a Wikipedia paragraph.
Note: The full meaning of the claim might require context provided by the
    original paragraph.

2. Review the Documents
Definition: Passages, tables, or pieces of text retrieved from Wikipedia.
Task: Ignore documents that are clearly irrelevant to the claim.
Focus on finding any document that might contain information in clear conflict
    with the claim.

3. Consider Clarifications

Definition: Additional background information provided to clarify ambiguous terms
    or entities.
Task: Use clarifications to distinguish between similar or similarly named
    entities.
Important: Do not use clarifications to support or contradict the claim
    directlythey serve only to clear up ambiguities.

4. Assess for Inconsistencies

Definition of Inconsistency:
The claim is inconsistent if at least one document provides information that
    contradicts it.
Conversely, if no document provides conflicting information, the claim is
    considered consistent.
Measurement: Assign an inconsistency score between 0 (fully consistent) and 1
    (completely inconsistent).
Intermediate scores indicate varying degrees of uncertainty or partial conflict.

5. Common Scenarios & Examples

Example 1: Clear Inconsistency

Claim: "The capital of Thailand is Bangkok."
Document: "The capital of Thailand is Phuket."
Reasoning: A country typically has one capital. The document contradicts the
    claim by listing a different city, yielding a high inconsistency score (e.g.,
    0.8-0.9).

Example 2: Apparent Inconsistency Resolved by Entity Equivalence (Minor
    Inconsistency)

Claim: "The capital of Thailand is Bangkok."
Document: "The capital of Thailand is Krung Thep Maha Nakhon."
Additional Background: It is widely accepted that Bangkok and Krung Thep Maha
    Nakhon refer to the same city.
Reasoning: Although the names differ, they reference the same location; thus, the
    claim is largely consistent (e.g., inconsistency score around 0.2-0.4).
Note: If an explicit clarification were provided stating the equivalence, the
    score would be 0.

Example 3: Misplaced Terms Causing Inconsistency

Claim: "The capital of Thailand is Bangkok."
Document: "The capital of Bangkok is Thailand."
```

Figure 19: Verifier Prompt

```
Reasoning: The document seems to mix up entities by stating that Bangkok is a
    country. With no supporting evidence that this is a mere typo or
    misinterpretation, the conflict earns a high inconsistency score (e.g.,
    around 0.9).

Example 4: Inconsistent Translational Variants

Claim: "The 'Song is Universal' won the Best Modern Rock Song award at the 2010
    Korean Music Awards."
Document: "The Best Modern Rock Song award at the 2010 Korean Music Awards was
    given for 'Universal Song.'"
Additional Clarification: "Bangkok only refers to a city in Thailand, not
    elsewhere." (Not directly applicable here but shows how clarifications work.)
Reasoning: Although the song likely is the same, the differing English
    translations ("Song is Universal" vs. "Universal Song") introduce an
    inconsistency, resulting in a moderately high inconsistency score (e.g.,
    around 0.8).

Example 5: No Conflict (Consistency)

Claim: "Stress is harmful to health, as mentioned in the medical literature."
Document: "Stress is necessary for growth and development, pushing limits,
    enhancing learning, and building resilience."
Reasoning: The document discusses the beneficial aspects of acute or eustress
    compared to chronic stress, which is what the claim addresses. Since these
    are two different perspectives on stress, there is no contradictionthe claim
    is consistent (inconsistency score 0).

Final Decision

After review, provide the inconsistency score for the claim:
0: Fully consistent; no document contradicts the claim.
Between 0 and 1: Partial or potential inconsistencies.
1: Fully inconsistent; at least one document directly contradicts the claim.

# input
<claim>
Title: {{ claim.context_block.full_title }}
{{ claim.context_block.content }}

You should only focus on the aspect of this paragraph related to: "{{
    claim.claim_text }}"
</claim>

Read the following clarifications about the claim:
<clarifications>
{% for clarification in clarifications %}
[{{ loop.index }}] {{ clarification }}

{% endfor %}
</clarifications>

Now, read through the documents below and look for any information that conflicts
    with the claim:
<documents>
{% for document in documents %}
[{{ loop.index }}] {{ document }}

{% endfor %}
</documents>
```

Figure 20: Verifier Prompt (Continued)

```
Now, you need to analyze the documents and clarifications to determine an
    inconsistency score that represents your confidence that the claim is
    inconsistent with the documents.

First, rephrase the claim to be more specific by:
1. Incorporating context from the Wikipedia article title and content
2. Preserving the original meaning, but making corrections if the claim appears
    to be misrepresented or incorrectly paraphrased from the claim's context.

<claim_with_context>
[Provide the claim that incorporates the context from the Wikipedia article title
    and content here.]
</claim_with_context>

Based on the claim with context, present your full analysis and arguments:

<analysis>
[Provide a detailed analysis by:
1. Carefully examining the claim and documents for any contradictions or
    inconsistencies, look through examples above if there is concept similar to
    your case
2. Highlighting specific documents where information directly conflicts with the
    claim
3. Making sure that these documents are relevant to the claim. Some documents may
    contain the same entities as the claim, but they are not relevant to the
    claim, given the context
4. Exploring multiple interpretations of the claim's meaning and implications
5. Considering edge cases and ambiguities that could affect the analysis
6. Referencing relevant examples from above (translations, time-related issues,
    ordering) to strengthen your reasoning
7. Explaining your confidence level in identifying any inconsistencies found]

</analysis>

Based on your analysis, provide an inconsistency score from 0 to 1, where:
- 0 indicates the claim is completely consistent with all of the documents
- 1 indicates the claim is completely inconsistent with at least one of the
    documents
- Values between 0 and 1 represent varying degrees of uncertainty

<inconsistency_score>
[A single float from 0 to 1]
</inconsistency_score>
```

Figure 21: Verifier Prompt (Continued)

```
# instruction
You will be given a "claim" statement extracted from a Wikipedia paragraph.
Your task is to conduct a thorough investigation on the entire Wikipedia (except
    the article where the claim comes from) to find any factual inconsistencies
    with this claim.
As you conduct your investigation, you may come across articles that support the
    claim. However, you should continue searching for inconsistencies that might
    exist in other places. Inconsistencies might appear in subtle or indirect
    ways.

You will conduct your investigation in multiple steps. At each step, you should
    think about the information you have gathered so far, and choose one of the
    following actions based on it:

- `explain(topic: str) -> str`: Use this action to understand the basics of a
    specific term or concept you encounter, for example a technical term or the
    rules of a sport.

- `clarify_entity(entity_name_and_description: str) -> str`: Use this action to
    get a report on an entity (person, organization, event etc.) to clarify other
    entities with similar names. This will help you properly differentiate
    similar-sounding entities when researching inconsistencies. For example,
    clarify_entity("WW III wrestling event") will explain all potential events
    with similar names, or the same event in different years.

- `search_wikipedia_outside_claim_article(question: str) -> list`: Use this
    action to explore Wikipedia.

- `report_inconsistency(evidence: str)`: If at any point you are certain that you
    have found an inconsistency, use this action to report it. Evidence should be
    a short sentence that describes the inconsistency. Once you report an
    inconsistency, a human will review it and provide feedback.

# input
Here is the claim to find inconsistencies with:
{{ claim.claim_text }}

Here is more context about the claim for your reference:
Title: {{ claim.context_block.full_title }}
{{ claim.context_block.content }}

{% if action_history %}
Actions you have taken so far:

{{ action_history}}
{% endif %}
```

Figure 22: Controller Prompt