TabVer: Tabular Fact Verification with Natural Logic

Rami Aly
University of Cambridge
Department of Computer Science
and Technology, UK
rami.aly@cl.cam.ac.uk

Andreas Vlachos
University of Cambridge
Department of Computer Science
and Technology, UK
andreas.vlachos@cl.cam.ac.uk

Abstract

Fact verification on tabular evidence incentivizes the use of symbolic reasoning models where a logical form is constructed (e.g., a LISP-style program), providing greater verifiability than fully neural approaches. However, these logical forms typically rely on well-formed tables, restricting their use in many scenarios. An emerging symbolic reasoning paradigm for textual evidence focuses on natural logic inference, which constructs proofs by modeling set-theoretic relations between a claim and its evidence in natural language. This approach provides flexibility and transparency but is less compatible with tabular evidence since the relations do not extend to arithmetic functions. We propose a set-theoretic interpretation of numerals and arithmetic functions in the context of natural logic, enabling the integration of arithmetic expressions in deterministic proofs. We leverage large language models to generate arithmetic expressions by generating questions about salient parts of a claim which are answered by executing appropriate functions on tables. In a few-shot setting on FEVEROUS, we achieve an accuracy of 71.4, outperforming both fully neural and symbolic reasoning models by 3.4 points. When evaluated on TabFact without any further training, our method remains competitive with an accuracy lead of 0.5 points.

1 Introduction

Fact verification systems assess the veracity of claims based on evidence and provide an explanation for the prediction. In the case of tabular evidence, verification frequently relies on symbolic reasoning steps, such as the execution of arithmetic functions, to accurately predict whether a claim is supported by evidence (Herzig et al., 2020, *inter alia*). This incentivizes symbolic reasoning systems, where a logical representation of a claim and its tabular evidence (e.g., a LISP-style program) is executed to produce the veracity prediction (Chen et al., 2020; Cheng et al., 2023). Since the execution of these logical forms is deterministic, they serve as faithful explanations of the model's reasoning (Jacovi and Goldberg, 2021). However, these systems typically rely on well-formed tables, constraining their use in many scenarios, such as reasoning over diverse tabular structures as typically found on Wikipedia. Consequently, the majority of recently proposed verification models focus on neural entailment models that latently execute arithmetic functions (Liu et al., 2022b; Gu et al., 2022) or generate a natural language explanation alongside its prediction (Wei et al., 2022, *inter alia*). While systems that produce natural language explanations are more flexible regarding the evidence format, they do not necessarily generate faithful explanations (Atanasova et al., 2023).

An emergent symbolic reasoning paradigm for textual evidence focuses on logical inference by directly comparing claim and textual evidence via natural logic inference (Angeli and Manning, 2014), achieving high prediction accuracy while maintaining faithful explanations (Krishna et al., 2022; Aly et al., 2023). However, current natural logic systems are unable to handle tabular evidence since the semantic relationship captured between aligned claim-evidence spans via natural logic's set-theoretic operators does not extend to arithmetic functions (MacCartney and Manning, 2009). For instance, in Figure 1, no evidence in the table directly corresponds to the part of the claim that states *three municipalities*. Instead, arithmetic computation on the table beyond the expressiveness of natural logic's set-theoretic operators is required (i.e., counting relevant cells).

To this end, we propose TabVer: **Tab**ular Fact **Ver**ification, a natural logic inference system that adds arithmetic reasoning capabilities to reason over tabular evidence directly in natural language. We define a set-theoretic interpretation of

Transactions of the Association for Computational Linguistics, vol. 12, pp. 1648–1671, 2024. https://doi.org/10.1162/tacl_a_00722
Action Editor: Kenji Sagae. Submission batch: 5/2024; Revision batch: 11/2024; Published 12/2024.

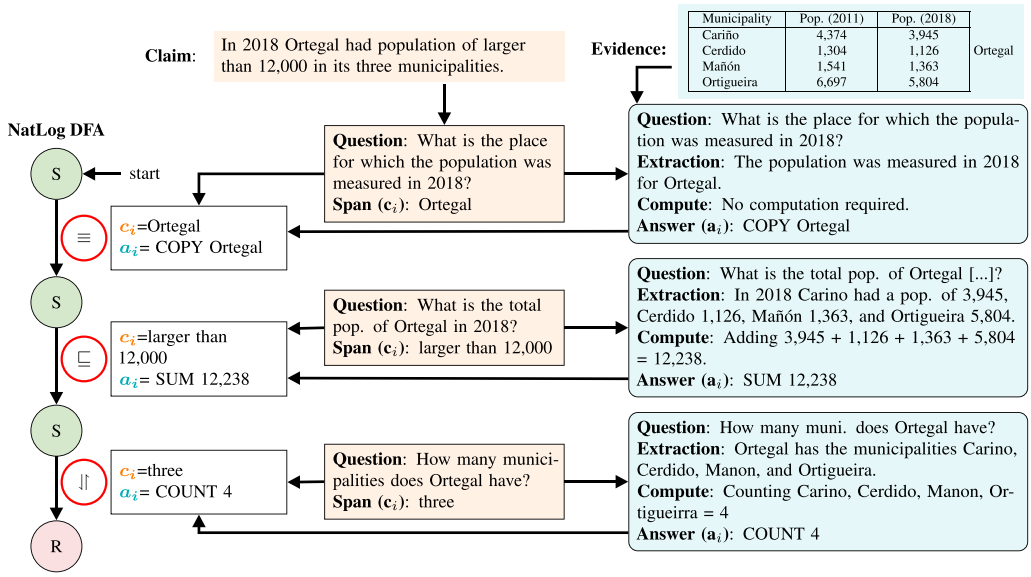

Figure 1: TabVer proposes a set-theoretic view on numerals and arithmetic functions, which is integrated into natural logic proofs as arithmetic comparisons between claim and answers to questions (ArithExps), resulting in deterministic inference (left). To generate ArithExps, TabVer asks questions about salient parts c_i of a claim (middle), which are answered using tabular evidence E following the generation of a rationale and a set-theoretic compatible representation of the computation (right), serving as the answer a_i aligned to c_i.

comparisons between numerals in claim-evidence pairs, and extend that definition to executions of arithmetic functions via arithmetic expressions (ArithExps) to enable their integration into natural logic proofs. The proofs are executed deterministically on a finite state automaton (DFA) as defined in natural logic inference. ArithExps are produced by leveraging large language models (Brown et al., 2020, *inter alia*), generating questions about salient parts of the claim c_i, which are answered via a rationale that produces an answer a_i. As illustrated in Figure 1, TabVer will generate a question such as ''*What is the total population of Ortegal in 2018*'' to verify the part *larger than 12000* in the claim c. Answering this question on the evidence table produces a rationale with the expression *SUM 12,238* as the final answer a_i, indicating the execution of the function $sum(3945, 1126, 1363, 5804) = 12238$ over relevant evidence in E. The aligned pair (*larger than 12000*, *SUM* $12,238$) is then assigned a natural logic operator as part of a natural logic proof, with the predicted operator being consistent with our set-theoretic definitions (cf. Figure 3).

In a few-shot setting with 64 training instances on the tabular subset of the FEVEROUS dataset (Aly et al., 2021), TabVer outperforms previous symbolic reasoning systems, including LPA (Chen et al., 2020), SASP (Ou and Liu, 2022), Binder (Cheng et al., 2023), and a state-of-the-art natural logic system (Aly et al., 2023) by 10.5 accuracy points. Moreover, TabVer outperforms the highest-scoring neural entailment model by 3.4 accuracy points, including baselines such as TAPAS (Herzig et al., 2020), TAPEX (Liu et al., 2022b), PASTA (Gu et al., 2022), and large language models of similar size as TabVer. We confirm the tabular reasoning capabilities of TabVer in a domain transfer setting to Tabfact (Chen et al., 2020) without further training annotations. Our system performs competitively, leading over the strongest baseline by 0.5 accuracy points. Our analysis reveals that TabVer's reading of numerals is more sensitive to numerical inaccuracies and the pragmatic context of a claim (i.e., quantifiers and rounding) than a same-sized LLM baseline, reflecting the annotator guidelines of FEVEROUS more accurately. Finally, the arithmetic functions invoked in TabVer's proofs are more accurate than the ones called in the logical form of our symbolic reasoning baselines.[1]

2 Related Work

Symbolic reasoning systems for fact verification convert text into a logical form or executable

[1]Code at `https://github.com/Raldir/TabVer`.

program (SQL/LISP-style). They typically involve a neural component, either to rank viable candidate programs consisting of hand-crafted functions (Chen et al., 2020) or via neural-symbolic models that generate programs directly (Liang et al., 2017; Ou and Liu, 2022). These programs are faithful explanations since the program's execution is the verdict. With the improved capabilities of large language models to generate code (Chen et al., 2021), Cheng et al. (2023) and Glenn et al. (2024) explore the use of SQL, Python, and FOL to faithfully fact-check tabular claims, however, they only use proprietary models consisting of hundreds of billions of parameters. We show that TabVer outperforms these approaches (when controlled for the language model), which we attribute to the suitability of natural logic to natural language in contrast to query languages like SQL.

The aforementioned symbolic executioners stand in contrast to the more prominent approach of using programs as features to neural systems, typically complemented by the original claim and table. For instance, LISP-style programs are used as a latent signal for a graph neural network (Shi et al., 2020; Zhong et al., 2020; Yang et al., 2020; Gong et al., 2023), and SQL queries and their executions are used as features to an LLM serving as a verdict classifier (Kong et al., 2024; Zhang et al., 2024c; Wu and Feng, 2024). Wang et al. (2024) incrementally update an evidence table with LISP-style operations. Alternatively to symbolic integration into neural systems, Chen (2023) produce natural language explanations using chain-of-thought prompting (Wei et al., 2022). Chen (2023) show that a 175B parameter GPT-3 model competes with fully supervised systems on tabular claims, yet its 6.7B variant performed only slightly above chance. This observation has been further confirmed by Zhang et al. (2024a) with Llama2-Chat-7B. Finally, large-scale instruction-tuning on tabular tasks has been explored (Zhuang et al., 2024; Zhang et al., 2024b; Liu et al., 2023), however they do not produce explanations. Conclusively, previous systems either rely on large proprietary models to achieve competitive performance or they sacrifice prediction explainability.

In contrast to these explicit meaning representations, Angeli and Manning (2014) propose to use NatLog (MacCartney and Manning, 2007, 2009) for textual inference, operating directly on natural language by *comparing* texts in a premise with an associated hypothesis using set-theoretic relations. Thus, as a framework of flexible compositional inference, it circumvents the requirement to convert statements into rigid logical forms, and typically independently from one another. These favorable properties of natural logic inference have subsequently recently been explored for fact verification, resulting in accurate predictions while maintaining transparency with plausible explanations (Krishna et al., 2022; Aly et al., 2023). Aly et al. (2023) exploit natural logic's operations on natural language by casting the operators into a question-answering framework to leverage recent advances of instruction-tuned language models. This paper is the first attempt to extend natural logic inference for fact verification to the tabular domain.

Finally, tabular question answering (Jin et al., 2022) is a common component to decompose a claim and reasoning processes. Yang and Zhu (2021) supplement the evidence with answers to questions generated via decomposition templates while Suadaa et al. (2021) supplement the evidence with information from a table-to-text model. More recently, Ye et al. (2023) use LLMs to decompose tables and questions. However, all three methods feed these modified tables into a pre-trained neural model (Herzig et al., 2020), ultimately producing veracity predictions without explanations. Finally, even for textual evidence, most previous work that generates questions conditioned on the claim does not construct proofs from the answers (Rani et al., 2023; Fan et al., 2020; Jobanputra, 2019).

3 Method

Given a claim c and a set of evidence tables E, the task is to predict a veracity label $\hat{y} \in \{\text{Supports, Refutes, Not Enough Information (NEI)}\}$, and to accompany the prediction with an explanation. Since evidence might require arithmetic reasoning beyond the expressiveness of natural logic, as shown in Figure 1 with *three municipalities*, TabVer's explanation is a proof $P = m_1, \ldots, m_l$, consisting of quintuples $m_i = (c_i, e_i, q_i, a_i, o_i)$, where o_i describes the set-theoretic relation (NatOp) between a claim span c_i and the result a_i of arithmetic computations executed over relevant evidence e_i.

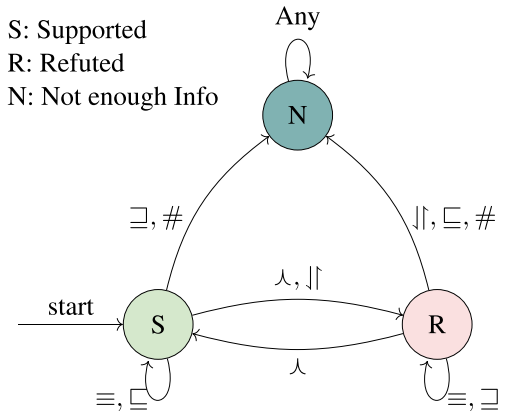

Figure 2: The finite state automaton (DFA), following natural logic inference (Angeli and Manning, 2014). The transitions in the DFA denote NatOps and the states the veracity labels. The final state on which the the proof terminates determines the overall veracity.

TabVer performs arithmetic reasoning steps in a question-answering framework, producing an arithmetic expression (ArithExp) with a_i being the answer to a question q_i for a claim span c_i answered over evidence e_i. The sequence of operators $O = o_1, \ldots, o_l$ is then the input to a finite state automaton that specifies the claim's veracity label $\hat{y} = \text{DFA}(O)$. We follow the DFA for textual entailment described in Angeli and Manning (2014), shown in Figure 2.

To enable the assignment of NatOps o to ArithExps, we need to expand the set-theoretic definition of these operators. To this end, we first discuss the set-theoretic relationship for numerals that occur in claim and evidence without the need for further computation (Section 3.1). We subsequently expand this definition to ArithExps where arithmetic functions are applied to evidence, by mapping function executions on relevant evidence to numerical representations (Section 3.2). TabVer produces its quintuples $(c_i, e_i, q_i, a_i, o_i)$ by first generating a question q_i about a claim span c_i that contains salient information (Section 3.3). This question is answered using the evidence E by producing a rationale, consisting of extracted evidence e_i, the execution of appropriate arithmetic functions on e_i, and the final answer a_i (Section 3.4). Finally, a proof generation model M_P, trained on proofs containing ArithExps and associated NatOps following our set-theoretic definitions, assigns a NatOp o_i to the claim-answer pair. TabVer follows QA-NatVer (Aly et al., 2023) for the proof generation process by selecting over multiple proof candidates.

3.1 A Set-theoretic Perspective on Numerals

We first define a set-theoretic interpretation of the relationship between numerals in claim spans and evidence (or answers calculated on the evidence with ArithExps), within the context of natural logic. Specifically, we consider five set-theoretic relationships (NatOps) $o \in \{\equiv, \sqsubset, \sqsupset, \curlywedge, \curlyvee, \}$.[2] Figure 3 shows examples of numerical expressions as evidence e_i with the associated claim span c_i for each NatOp. For instance, a claim span *about a hundred goals* would generally follow from the evidence *99 goals* since the explicit adverbial modifier *about* widens the scope of the numeral *a hundred* to a larger set, including, e.g., *99* and *101*. However, even bare numerals can carry implicit meaning beyond the utterance itself, referred to as scalar implicature (Grice, 1975, *inter alia*), and are subject to both semantics and pragmatics.

Linguistic approaches to numerals typically consider an upper-bounded (exact) and a lower-bounded (at least) reading, depending on several factors such as whether an environment is upward- or downward-entailing[3] (Panizza et al., 2009). Suitably, the effect of these environments on the entailment relationship between claim and evidence is modelled explicitly in natural logic (MacCartney, 2009), enabling these different readings of numerals into a model of natural logic. Since the majority of claims appear in an upward-entailing environment, we focus here on the set-theoretic reading of numerals in an upper-bounded definition. We discuss a downward-entailing projection of numerals when following an *at least* reading in Appendix A. In an upper-bounded reading, the terminology of natural logic can be extended such that evidence spans like *5 goals* aligned to claim spans with a strictly smaller number like *two goals* are assigned the alternation NatOp ($\curlyvee$) since an upper-bounded reading assumes that 2 goals and 5 goals are mutually exclusive without covering

[2]We do not define a mapping to the independence NatOp (#) since it is applied when none of the other operators are predicted. Similarly to Krishna et al. (2022) for textual relations, we observe that the cover NatOp occurs only very rarely, thus replacing it with the independence NatOp (#).

[3]Downward-entailing environments are, for instance, negative environments, antecedent clauses of conditional constructions, restrictors of universal quantifiers (Spector, 2013). Example for upward (downward) entailment: Messi (has not) scored 50 goals in a season.

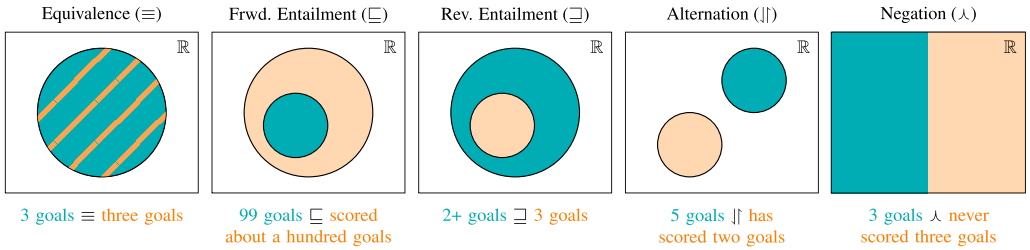

Figure 3: A set-theoretic view on the relationship between claim spans c_i and numerical expressions in the evidence e_i when following an upper-bounded (i.e., exact) interpretation of numerals.

the entire universe, i.e., all natural numbers (cf. Appendix A).

Another component of a numeral's reading to consider is its *pragmatic halo* (Lasersohn, 1999), where a number can represent a range of values due to the intended degree of approximation to the truth in a specific context. As seen earlier, a halo can be indicated explicitly with modifiers (cf. *about*), yet it is also often defined implicitly. For instance, a claim like ''*Messi scored a hundred goals in the 2010 season.*'' might be considered supported by evidence that states he scored 101 goals in the context of an environment with low requirements of numerical precision, e.g., on social media, since the communicated content ($\{100\}$) is weaker than the asserted content ($101 \in \{100\} \cup H_{100}$), with H_{100} being the pragmatic halo of 100 as a set of (integer) numbers.[4] However, the evidence would lead to the claim's refutation in an environment where exactness is required, i.e., when $H_{100} = \emptyset$[5], e.g., statements in scientific articles. The size of the pragmatic halo typically increases with larger numbers, thus it becomes less necessary pragmatically to be precise. Therefore, Vlachos and Riedel (2015) consider a fixed threshold on the absolute percentage error between numbers in a claim and evidence. Yet, in reality, the halo of a number is more dynamic: In decimal number systems, such as English, multiples of ten and five generally have a larger pragmatic halo than others due to the communicative tool of rounding (Woodin et al., 2024). For instance, the claim that Messi scored 100 goals while evidence states he scored 101 is more likely to be accepted than the reverse since 101 is not expected to be a rounded number, hence $|H_{100}| > |H_{101}|$. Conveniently, the pragmatic halo can be expressed by natural logic via a projection to the entailment NatOps (e.g., Frwd. Entailment in Figure 3) and is learned on annotated proof data (cf. Section 4.3).

3.2 Arithmetic Expressions

Since evidence e_i is often stated in terms different than those needed to verify a textual claim, e.g., as seen in Figure 1, we introduce ArithExps, which map tabular evidence to numerals by executing arithmetic functions. ArithExps are function executions that produce an answer a_i for a question q_i to an associated claim span c_i over relevant evidence e_i from the table E. For the computation of a_i we consider functions $\mathbb{F} : \mathbb{R}^n \rightarrow \mathbb{R}$ that take as input evidence e_i and output a single numeral. The answer of the ArithExp is represented as the result of the computation prepended by the function's name: $\text{Name}(\mathbb{F}) \oplus \mathbb{F}(e_i)$, with $e_i \subseteq E$.[6] Figure 1 shows the ArithExp *SUM 12,238* (for the sum of the cells 3,945, 1,126, 1,363, and 5,804) as answer a_i aligned to the claim span c_i *larger than 12,000*. To extend ArithExps to cover more complicated computations, we enable function composition, i.e., a function $\mathbb{G}$ as an input argument to $\mathbb{F}$. The ArithExp for function composition is the final computation, i.e., $\mathbb{F}$ for $\mathbb{F}(\mathbb{G}(E_i), \mathbb{G}(E_j), \cdots)$.

[4]Note that this phenomenon is distinct from truth-conditional vagueness where modifiers are hidden. While a sentence like ''*Messi scored about 100 goals, he scored 102.*'' is semantically valid, ''*Messi scored 100 goals, he scored 102.*'' is not without explicitly correcting the previous statement with a modifier like *actually*; e.g., ''*Messi scored 100 goals, actually he scored 102.*'' (Lauer, 2012).

[5]Lasersohn (1999) argues that the term *exact* also leaves room for pragmatic slack at times, e.g., in a statement such as *Mary arrives exactly at 5 o'clock*, where deviations by milliseconds are permissible in most situations. We ignore this notion for simplicity.

[6]Despite the ArithExp's treatment as a numeral, the function name as part of a_i is important since the semantics of a numeral varies between arithmetic functions (e.g., COUNT 5 versus COMP 5) and thus affect the comparison against claim span c_i.

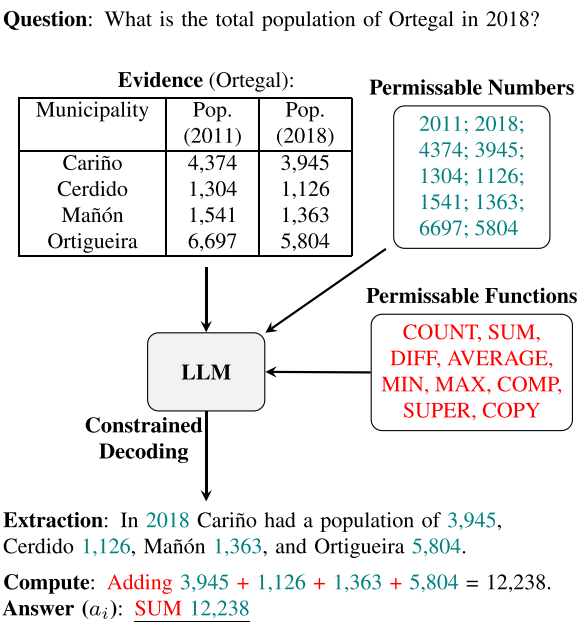

Figure 4: Tabular question answering via a rationale that produces the answer a_i via an ArithExp. An LLM jointly extracts relevant information from table cells and executes appropriate functions. The generation is constrained to permissible functions and to numbers that appear in the evidence to alleviate hallucinations.

The full list of permissible functions $\mathbb{F}$ we consider is shown in Figure 4. In addition to the functions *count, sum, diff, average, min*, and *max*, we consider comparative functions as a separate function class. Comparatives could be modeled by the *diff* function, thus subtracting quantities between relevant arguments. However, we represent them as a unique ArithExp since they serve a different semantic function in relation to a claim span c_i. The comparative ArithExp can be used for both implicit (e.g., *Person X had more votes than Person Y*) as well as explicit comparisons (e.g., *Person X had 5000 votes more than Person Y*) since the difference in quantity is indicative of both polarity and magnitude. Finally, to cover the base case where all relevant information is already contained in e_i (i.e., no computation is required, cf. Section 3.1), we consider a copy function.

3.3 Question Generation

We generate questions that can directly be linked to salient parts of a claim c_i, as seen in Figure 1. For instance, the question *What is the total population of Ortegal in 2018* directly corresponds to the claim span *larger than 12,000*. We use a fine-tuned large language model $M_{QG}(c, T)$, which takes a claim c and a prompt template T as input and autoregressively generates a collection of questions $q_1 \ldots q_l$ along with their corresponding targeted claim spans. The output is formatted as a list of questions and claim spans 1. [q1] [c1] 2. [q2] [c2]$\cdots$. To ensure that the generated claim span occurs verbatim in the claim, we employ constrained decoding to restrict the sampling of c_i to spans of the claim c (including c itself). Thereby we prevent the model from introducing words or phrases that are not present in the claim, a behavior we observed even after fine-tuning. Additionally, we use constrained decoding to enforce the enumeration format defined in the prompt above for generating multiple questions jointly. By conditioning the generation of questions on previously generated ones, we can improve coverage of salient information in the claim and reduce the likelihood of redundant or repetitive questions (Fan et al., 2020).

3.4 Tabular QA with ArithExps

The next step is to answer a generated question q_i using information from the evidence tables E whilst using only permissible functions $\mathbb{F}$ to obtain the answer a_i and the ArithExp. We use a fine-tuned language model M_{QA} that takes as input a question q_i, associated with claim span c_i, and evidence tables E and it generates a rationale J consisting of three parts: table-to-text conversion to extract relevant evidence e_i from table cells in E, the execution of relevant arithmetic functions on e_i, and the answer representation a_i. The components of the rationale J are generated jointly in an autoregressively fashion:

$$M_{QA}(J \mid q_i, E) = \underbrace{\prod_{u=1}^{U} p_\theta(s_u | s_{<u}, q_i, E)}_{p_\theta(e_i | q_i, E)} \cdot \underbrace{\prod_{m=1}^{M} p_\theta(f_m | f_{<m}, q_i, e_i, E)}_{p_\theta(\mathbb{F} | q_i, E, e_i)}$$

with s_u and f_m being decoded tokens over the extracted evidence e_i from cells in E and arithmetic functions $\mathbb{F}$, respectively, and θ being the parameterization of M_{QA}. The components of J are generated using a different decoding scheme, illustrated in Figure 4. To avoid hallucinations in the extracted evidence e_i, the probability $p_\theta(e_i | q_i, E)$ is set to 0 for sequences where numbers in the generated sequence do not occur in any table cells

of E. The generation of $p_\theta(\mathbb{F}|q_i, E, e_i)$ is constrained such that the first generated word needs to be a trigger word for one of the permissible functions $\mathbb{F}$ (such as *Adding* for *sum*), followed by the function itself. Finally, the answer a_i is constructed deterministically from the rationale to align with the representation of ArithExps as described in Section 3.2. As shown in the example Figure 4, the extraction of the population numbers is constrained to numbers occurring in the evidence table, such as *2018; 3,945; 1,126; 1,363; and 5,804* (blue colour), the arithmetic computation is constrained to start with the trigger word *Adding* (red color), followed by the execution of the associated function, with the final answer a_i being *SUM 12,238*.

If no function is considered relevant to further process the evidence e_i, the model M_{QA} outputs *N/A* after the extraction of evidence and subsequently does not return an ArithExp. If the evidence tables do not contain any relevant information to answer q, then the model returns *N/A* as the relevant evidence e_i, which is mapped to an independence NatOp (#), leading to an NEI verdict prediction according to the DFA (cf. Figure 2). Parts of a claim that do not require separate questioning (such as *In 2018* in Figure 1) are assumed to be contained in extracted evidence for answering questions about claim c. QA-NatVer's span alignment algorithm aligns these claim spans to extracted evidence e_i from all the questions $q_1 \ldots q_l$ to the claim c.

4 Evaluation

4.1 Data

FEVEROUS We train and evaluate models on the tabular subset of FEVEROUS (Aly et al., 2021), i.e., the claims where all evidence elements across all evidence sets are cells from tables. FEVEROUS consists of complex claims and tables with irregular structures. To focus on the natural logic-based tabular fact verification component of fact-checking, we use gold evidence tables (i.e., not ones selected via a retrieval system from a knowledge source) throughout our experiments. The resulting dataset consists of 2,011 claims, with 35%, 61.7%, and 3.2% being supported, refuted, and NEI claims, respectively (cf. Appendix Table 8). Out of the 2,011 claims, 521 are labelled as requiring numerical reasoning.

Models are trained on 64 FEVEROUS instances, selected uniformly from its training data. The veracity labels in the resulting training data are thus similarly imbalanced as the FEVEROUS development data. To train TabVer we additionally manually annotated these training instances with rationales and natural logic proofs. These proofs contain ArithExps as defined in Section 3.1. The training distribution of arithmetic functions is also imbalanced. For details see Appendix B.

TabFact We further evaluate models trained on FEVEROUS in a domain transfer scenario on TabFact (Chen et al., 2020), without further training on the latter. Contrary to FEVEROUS, TabFact only contains two veracity labels: Supported and Not Supported, the latter covering both refutations and NEI instances. TabFact contains only well-structured tables; the first row is always the table header. TabFact is designed to be evaluated on gold evidence tables E. We evaluate methods on its development set, consisting of 12,851 claims with evenly distributed labels, out of which 4,424 are simple (R1) and 8,427 complex claims (R2).

4.2 Baselines

We compare TabVer against strong baselines that can be categorized into two classes: (i) classifiers that predict a veracity label without symbolic mechanisms or explanation production (ii) symbolic reasoning models that produce faithful explanations.

Classification models. **DeBERTa+NLI** is a DeBERTaV3 model (He et al., 2023) fine-tuned on multiple NLI tasks. **PASTA** (Gu et al., 2022) is a DeBERTaV3 model further pre-trained on different tabular operations. **TAPAS** (Herzig et al., 2020) is a transformer pre-trained on tabular data with additional table-aware positional embeddings. **TAPEX** (Liu et al., 2022b) is based on BART (Lewis et al., 2020), pre-trained as an SQL executor and fine-tuned on tabular data via table linearization. We follow typical encoder-only fine-tuning, where a linear transformation from embeddings to veracity labels is jointly optimized with the pre-trained model itself. Furthermore, we evaluate several LLMs, including **Llama2-Chat-7B** (Touvron et al., 2023)

and **MistralOrca-7B** (Jiang et al., 2023). We fine-tuned the LLMs via LoRA (Hu et al., 2022).

Symbolic Reasoning Models. We compare against **LPA** (Chen et al., 2020), a LISP-style program synthesis algorithm with hand-crafted functions and trigger words to prune the search space. It incorporates a fine-tuned transformer to rank candidate programs. **SASP** (Ou and Liu, 2022) is built on top of neural symbolic machines (Liang et al., 2018) and considers both lexical and structure features to constrain program candidates and further uses TaBERT (Yin et al., 2020) and an LSTM (Hochreiter and Schmidhuber, 1997) for program generation. We also consider **Binder** (Cheng et al., 2023), an approach that uses LLMs to map tabular claims to SQL queries and to execute specific API calls embedded in the queries on tables. To maintain comparability with TabVer, Binder uses MistralOrca-7B as the LLM. If no viable program can be found for a given claim, LPA, SASP, and Binder fall back to an NEI/Not Supported prediction. **QA-NatVer** (Aly et al., 2023) constructs natural logic inference proofs by casting natural logic operators into a question-answering framework. We linearize the evidence table and use the Flan-T5 3B backbone (Chung et al., 2024).

4.3 Implementation Details

Claim Decomposition. A FEVEROUS claim typically contains multiple factual statements which might not all be covered by the annotated evidence tables, since its annotations are only required to be sufficient (but not necessarily complete) to reach a verdict. Subsequently, the annotation for a refuted claim might lack evidence for some other parts of the claim, resulting in erroneous NEI predictions. Thus, a FEVEROUS claim c is decomposed into a list of sub-claims C, such that each sub-claim is an atomic statement contained in the claim. We use a language model M_D, fine-tuned on manually annotated decompositions of the same FEVEROUS training instances described above. During inference, the sub-claims are enumerated following the question generation decoding scheme. The decomposition prompt is shown in Appendix C. The predictions for each subclaim are aggregated into a final verdict $\hat{y}$ via deterministic rules:

$$\begin{aligned}
\hat{y} = \text{Supp} \quad & \text{iff } \forall c \in C.\text{DFA}(O_c) = \text{Supp} \\
\hat{y} = \text{Ref} \quad & \text{iff } \exists c \in C.\text{DFA}(O_c) = \text{Ref} \\
\hat{y} = \text{NEI} \quad & \text{iff } \nexists c \in C.\text{DFA}(O_c) = \text{Ref} \\
& \wedge \nexists c \in C.\text{DFA}(O_c) = \text{Supp},
\end{aligned}$$

thus a claim c is supported, iff all subclaims are supported by evidence, refuted iff there exists a sub-claim that is refuted, and not enough information is predicted in all other cases. Since these rules are deterministic, the final prediction remains faithful. See Figure 5 for an example.

We use the same decomposition for TabVer as well as all symbolic reasoning baselines we consider to maintain comparability. Classification models use the original claim as input instead, since the impact of evidence incompleteness is expected to be minimal and decomposition can lead to error propagation. With the exception of Wang and Shu (2023), who represent a claim as a conjunction over subclaims, the aggregations over verdicts of parts of a claim are executed via neural mechanisms and thus do not guarantee faithfulness (Chen et al., 2022; Zhao et al., 2024).

Experimental Setup. We do not consider a validation set for hyperparameter-tuning, following the real-world few-shot learning setting of Alex et al. (2021). TabVer fine-tunes the question generation model M_{QG}, the question answering model M_{QA}, and the proof generation model M_P on annotated handcrafted rationales and proofs described in Section 4.1. M_{QG}, M_{QA}, and and the claim decomposition model M_D are MistralOrca-7B models, fine-tuned using LoRA (Hu et al., 2022). We use the proof generation model M_P of Aly et al. (2023). Specifically, we fully fine-tune a FlanT5-3B parameter model and a smaller BART0 model (406M parameters) (Lin et al., 2022) as M_P to measure the accuracy of TabVer across model sizes. While it would be of interest to simplify TabVer by using MistralOrca-7B (or another powerful LLM) for all components, the implementation in Aly et al. (2023) currently only supports the training of encoder-decoder models, following Liu et al. (2022a). Furthermore, while M_{QG}, M_{QA}, and M_D require language generation, the proof generation model M_P of Aly et al. (2023) solves a discriminative task (answering binary/ternary questions), for which encoder-decoders have shown to be competitive to decoder-only models on smaller

		Full		Numerical		Execution Found
		Accuracy	Macro F_1	Accuracy	Macro F_1	(%)
	Majority Baseline	61.7	20.5	64.8	21.6	–
Classific.	DeBERTav3	$53.9_{0.7}$	$36.8_{0.4}$	$55.6_{0.6}$	$36.0_{1.3}$	–
	PASTA	$54.6_{2.8}$	$34.1_{0.4}$	$55.3_{4.3}$	$32.6_{1.4}$	–
	TAPAS	$53.6_{7.6}$	$35.9_{4.1}$	$52.9_{7.3}$	$33.8_{3.4}$	–
	TAPEX	$53.6_{1.5}$	$34.0_{0.9}$	$52.8_{3.4}$	$32.9_{2.1}$	–
	Llama2-Chat-7B	$56.0_{4.0}$	$30.9_{1.6}$	$55.0_{6.1}$	$30.9_{2.5}$	–
	MistralOrca-7B	$68.0_{1.1}$	$45.4_{4.4}$	$64.5_{3.2}$	$43.6_{3.0}$	–
Symbolic	LPA (w/o decomp)	$31.6_{0.4}$	$27.5_{0.5}$	$37.3_{0.7}$	$28.1_{0.9}$	54%
	LPA	$21.8_{0.1}$	$21.4_{0.2}$	$22.3_{0.4}$	$21.3_{0.4}$	41%
	SASP (w/o decomp.)	$52.9_{2.6}$	$29.8_{1.8}$	$55.1_{3.4}$	$29.3_{1.9}$	98%
	SASP	$58.8_{0.8}$	$29.6_{0.8}$	$61.5_{1.2}$	$29.4_{0.8}$	95.2%
	Binder (w/o decomp.)	$60.9_{1.2}$	$38.0_{1.3}$	$61.0_{1.6}$	$40.1_{2.2}$	95.7%
	Binder	$62.7_{1.4}$	$37.3_{1.3}$	$63.7_{1.8}$	$39.3_{1.6}$	95.4%
	QA-NatVer	$54.0_{1.1}$	$34.8_{0.2}$	$52.6_{1.6}$	$28.9_{0.3}$	100%
TabVer	BART0	$69.9_{0.3}$	$49.4_{0.9}$	$66.7_{0.3}$	$42.4_{0.8}$	100%
	FlanT5-xl	$\mathbf{71.4}_{0.5}$	$\mathbf{51.0}_{0.5}$	$\mathbf{70.1}_{1.3}$	$\mathbf{45.8}_{0.3}$	100%

Table 1: Verdict accuracy and macro-averaged F_1 on FEVEROUS. *Numerical* reports scores exclusively on the subset of claims involving numbers. *Execution found* is the proportion for which a program or proof was found.

scale (i.e., $\leqslant$ 11B parameters) (Chia et al., 2024). We leave the exploration of alternative model architectures and backbones for TabVer to future work. Implementation details and the prompts for all models are in Appendix C and A, respectively. Results are averaged over five runs with standard deviation indicated. In all other cases, results are reported using default seed 42.

5 Results

FEVEROUS Results on FEVEROUS are shown in Table 1, reporting both accuracy and macro average F_1 due to the dataset's label imbalance. TabVer outperforms all baselines both on the full dataset as well as the numerical reasoning subset by 3.4 and 5.6 accuracy points, respectively. We see similar differences in terms of F_1 with a lead of 5.6 points. Except for the LLM MistralOrca-7B baseline, all classification models perform poorly in a few-shot scenario on FEVEROUS. Llama2-Chat-7B model's surprisingly poor performance confirms previous observations on few-shot tabular fact-verification (Chen, 2023; Zhang et al., 2024a,b). In addition to the classification baselines being outperformed by TabVer, they lack transparency and faithful explanations. To highlight TabVer's data efficiency, we compare it against a fully supervised TAPAS classification model trained on 18,836 tabular FEVEROUS claims, where it achieves an accuracy score of 73.0, performing only 1.6 accuracy points better than TabVer.

While symbolic reasoning baselines provide faithful explanations, their performance is substantially worse than TabVer. Symbolic reasoning systems that construct semantic representations are unable to handle diverse and complex tabular structures (e.g., nested table headers) as present in FEVEROUS. For instance, the rule-based LPA approach finds a suitable program only for 41% of claims. The accuracy for claims where LPA finds a program is 55.8 points, improving by 25.6 points on its overall performance but still being outperformed substantially by TabVer. While the rate of executable programs is much higher for SASP and Binder due to the generation of programs being neural-guided, the overall performance is worse than TabVer, with a difference of 8.7 accuracy points for the best performing symbolic baseline, Binder. Finally, QA-NatVer has a 100% execution rate due to its flexibility by operating on natural language similarly to TabVer, however, the difficulty of aligning linearized evidence to claims and the lack of arithmetic reasoning capabilities result in low scores. Interestingly, the symbolic baselines perform better or comparably on the numerical subset than on the full dataset, while we observe the opposite for the majority of classification models and natural logic-based

		Full		R1		R2	
		Accuracy	Macro F_1	Accuracy	Macro F_1	Accuracy	Macro F_1
Full Supervision	LPA	65.2	64.2	77.6	77.5	57.4	55.6
	SASP	74.7	74.7	86.1	86.1	68.9	68.9
	TAPAS	82.1	82.0	92.8	92.8	76.5	76.4
Classific.	DeBERTav3	$50.7_{0.4}$	$49.7_{1.3}$	$50.8_{0.7}$	$49.5_{1.7}$	$50.6_{0.2}$	$49.8_{1.1}$
	PASTA	$50.4_{0.6}$	$46.1_{5.6}$	$50.6_{1.1}$	$46.4_{6.1}$	$50.4_{0.5}$	$45.9_{5.4}$
	TAPAS	$53.9_{5.9}$	$53.0_{6.8}$	$58.8_{10.6}$	$58.5_{11.3}$	$51.3_{3.6}$	$49.9_{4.7}$
	TAPEX	$49.7_{4.3}$	$44.3_{5.1}$	$49.5_{3.4}$	$47.6_{3.8}$	$49.8_{2.9}$	$43.3_{2.9}$
	Llama2-Chat-7B	$51.2_{1.6}$	$47.3_{4.2}$	$51.5_{2.5}$	$47.8_{4.3}$	$51.1_{1.2}$	$47.0_{4.2}$
	MistralOrca-7B	$60.6_{3.1}$	$58.1_{6.0}$	$67.2_{4.2}$	$65.9_{5.8}$	$57.2_{2.6}$	$53.3_{6.4}$
Symbolic	LPA	$59.4_{1.4}$	$57.9_{1.4}$	$70.4_{2.5}$	$70.3_{2.5}$	$53.8_{0.9}$	$50.2_{1.0}$
	SASP	$48.7_{2.8}$	$45.1_{2.9}$	$50.7_{3.0}$	$47.5_{2.0}$	$47.7_{3.0}$	$43.8_{3.7}$
	Binder	$65.1_{1.0}$	$\mathbf{65.1}_{1.0}$	$\mathbf{76.9}_{0.6}$	$\mathbf{76.9}_{0.6}$	$59.1_{1.3}$	$59.1_{1.3}$
	QA-NatVer	$50.9_{0.1}$	$43.6_{0.3}$	$52.7_{0.2}$	$49.8_{0.1}$	$49.9_{0.1}$	$49.1_{0.2}$
TabVer	BART0	$62.8_{0.8}$	$62.3_{0.9}$	$71.1_{1.0}$	$71.1_{1.1}$	$58.6_{0.6}$	$57.5_{0.9}$
	Flan-T5-xl	$\mathbf{65.6}_{0.3}$	$64.8_{0.6}$	$72.6_{0.5}$	$72.2_{0.6}$	$\mathbf{62.1}_{0.4}$	$\mathbf{60.8}_{0.9}$

Table 2: Verdict accuracy and macro-averaged F_1 in a transfer scenario on TabFact, when trained on FEVEROUS. R1: Tabfact's subset of simple claims. R2: TabFact's subset of complex claims.

approaches, confirming the difficulty for these meaning representations to model complex textual claims correctly. Qualitative examples and representation limitations are discussed in Appendix Figures 6 and 7.

TabFact. Results in a domain-transfer scenario without TabFact training data are shown in Table 2. TabVer still remains competitive with our baselines with an accuracy lead of 0.5 accuracy points and an F_1 of 0.3 points worse than the best baseline (Binder). The performance against the symbolic reasoning systems is particularly noteworthy since LPA and SASP have been designed specifically for TabFact, and Binder's SQL parsing excels at well-structured tables. Subsequently, LPA, SASP, and Binder find viable programs more frequently than on FEVEROUS, with 78%, 99.8%, and 100%, respectively. Binder performs the best out of all baselines, outperforming TabVer particularly on simple claims (R1) that do not require complex reasoning to predict correctly. Yet, on complex claims (R2) TabVer performs better than Binder. Binder's performance discrepancy between FEVEROUS and TabFact is noteworthy, highlighting a fundamental limitation to previous approaches when applied to diverse tables (cf. Listing 4), which TabVer successfully addresses.

Training classification baselines, such as TAPAS, on Tabfact's 92,283 training samples, using the same experimental setup, results in scores substantially outperforming all considered models (82.1 accuracy points). In contrast, TAPAS achieves a score barely above random in our transfer setting (53.9 accuracy points) since the small training size is insufficient for fine-tuning the model to the task and learning the linear transformation described in Section 4.3. This problem is exemplified with TAPEX as it is pre-trained only on SQL queries, necessitating substantial data during fine-tuning to learn a mapping to natural language. Compared to fully-supervised symbolic systems, TabVer remains competitive to LPA with an accuracy lead of 0.4 points, but falls behind SASP substantially with an accuracy difference of 9.1 accuracy points.

Reading of Numerals. We further analyze TabVer's reading of numerals by isolating its ability to consider the context of numbers mentioned in a claim.[7] We automatically construct a diverse probing dataset that considers variations of numbers in supported claims by adding numerical inaccuracies, rounding numbers, adding modifiers (i.e., approximately, about, around), and adding cardinal determiners (i.e., at most/least). We measure the proportion between veracity predictions

[7]The ability of models to make pragmatic inferences has been explored in Jeretic et al. (2020), however, their dataset was constrained to a minimal scenario with four numbers (2, 3, 10, 100) and two quantifiers (some, all). Importantly, while their dataset focuses on correctness, our goal is instead to probe a model's reading of numerals.

	Class.	Binder	TabVer
Inaccuracy Δ +1	63.4%	57.7%	36.3%
Inaccuracy Δ 2%	42.7%	38.4%	29.1%
Inaccuracy Δ 10%	37.8%	38.4%	26.3%
Inaccuracy Δ 25%	31.7%	51.9%	23.6%
Rounding	33.3%	47.4%	30.3%
Modifiers (e.g., about)	40.6%	56.4%	57.0%
Cardinal (incorrect)	32.9%	53.8%	31.8%
Cardinal (correct)	37.8%	78.8%	49.1%

Table 3: Probing how modifications to numbers impact veracity predictions. We report the proportion of claims for which a veracity prediction correctly labelled as supported does not flip after an edit to a claim's numeral. We consider absolute and relative numerical inaccuracies, approximations, explicit modifiers, and cardinals.

correctly labelled as supported and veracity predictions that remain supported after an inserted numeric variation. The probing dataset consists of 1638 claims. For a detailed description of the constructed variations see Appendix D.

Table 3 shows the results of the probe for TabVer, Binder, and the MistralOrca-7B classification baseline. TabVer is substantially more sensitive to small numeric inaccuracies. Only for 36.3% is the claim's prediction maintained when adding 1 to the original number, compared to 63.4% and 57.7% for Binder and the classifier, respectively. This trend is also observed for relative numerical inaccuracies and rounded numbers. We argue TabVer's behavior is more representative of its training data, since FEVEROUS instances are annotated to be refuted if numbers mentioned without modifier do not match exactly due to the guidelines given to annotators (Aly et al., 2021). In contrast, when adding explicit modifiers we observe that TabVer maintains its prediction more frequently than our baseline, with 57% versus 40.6% and 56.4% for the classifier and Binder, respectively. Finally, TabVer's more nuanced reading of numerals is also seen for cardinals: While the classifier cannot differentiate between incorrect cardinal determiners (e.g., 12 being modified to *at most 10* and changing the veracity label, and *at most 15* while preserving it), both TabVer and Binder differentiate between the two. Yet, Binder overall favours the prediction of supported, due to the answer-biased voting strategy deployed by Cheng et al. (2023).

	LPA	Binder	TabVer
Overall	43.8	76.5	76.0
Filter/Copy	41.7	85.6	90.4
Comparisons	33.3	0.0	25.0
Count	75.0	85.7	46.4
Sum	0.0	100.0	100.0
Diff	0.0	0.0	16.6
Min/Max	0.0	0.0	0.0

Table 4: Evaluation of arithmetic functions incorporated in TabVer's proofs, compared to LPA and Binder.

ArithExp	Constr.	Rationale	Decomp	Full	Num.
✓	✓	✓	✓	72.0	71.4
✓	✗	✓	✓	69.2	66.2
✗	✗	✓	✓	66.1	61.0
✗	✗	✗	✓	60.9	59.9
✓	✓	✓	✗	66.3	63.7
✗	✗	✗	✗	44.6	43.0

Table 5: Ablation study on the components of TabVer.

Correctness of ArithExps. To assess the quality of natural logic proofs with invoked ArithExps, we randomly select 160 FEVEROUS samples and annotate the arithmetic functions required to reach the correct verdict. We compare these annotations with functions identified by TabVer in the final proof used to assess the claim's veracity. LPA's programs are used as a comparison baseline. As seen in Table 4, the overall accuracy of TabVer's arithmetic function calls outperforms the LPA baseline with 76.0 versus 43.8 accuracy points, and is comparable with Binder's score of 76.5. The largest performance lead for Binder is observed for the count function whereas TabVer is more accurate at comparisons.

TabVer Ablation. Table 5 shows an ablation study of TabVer's components. We see a substantial performance decline when removing the generation of ArithExps, dropping accuracy on the numerical subset by 10.0 accuracy points. When additionally removing the extracted evidence e_i from the rationale and instead falling back to table linearization, we observe performance comparable to QA-NatVer, as expected. In line with our expectations, a major accuracy drop is observed on the full FEVEROUS data since the extraction and formatting of evidence is particularly useful

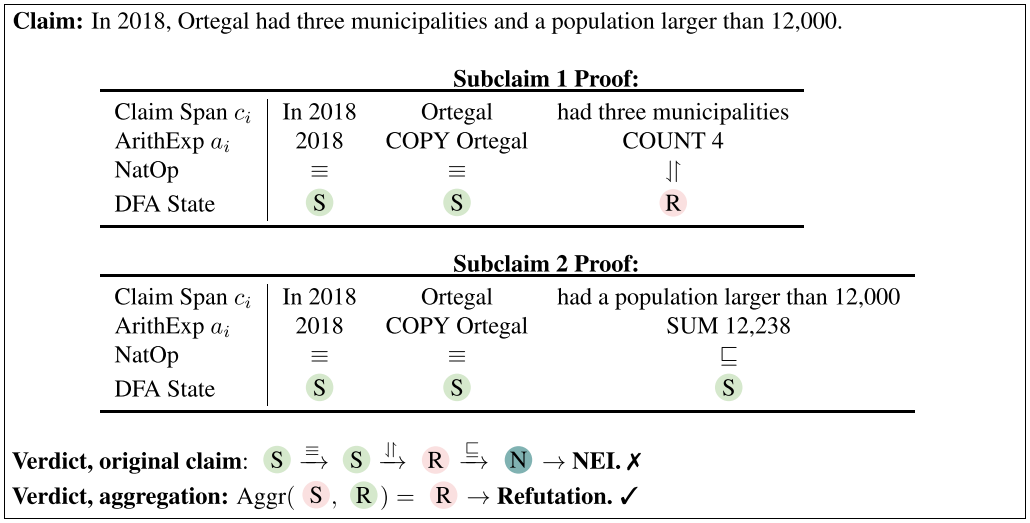

Figure 5: Illustrating claim decomposition and verdict aggregation. Further, claim decomposition partially addresses the issue of producing less informative predictions by constraining natural logic to left-to-right execution.

for non-arithmetic claims. Finally, the removal of claim decomposition results in accuracy scores worse than the majority baseline. We observe that the removal of claim decomposition results in substantially more NEI predictions for longer claims, further discussed in Section 6.

6 Limitations

While the addition of arithmetic reasoning capabilities addresses a vital limitation of natural logic-based systems, TabVer is not attempting to modify natural logic's model of compositional entailment itself (i.e., the DFA in Figure 2). NatLog fails some inferences, such as De Morgan's laws for quantifiers, generally having less deductive power than first-order logic (MacCartney and Manning, 2014; Karttunen, 2015). In contrast, TabVer incorporates relevant reasoning processes in the generated proof either explicitly, e.g., ArithExps and claim decomposition, or latently, e.g., the assignment of NatOps between an aligned claim and evidence span. Moreover, inference rules that cannot be produced by NatLog affect the granularity of the proof: Consider a natural-language instantiation of De Morgan's law from MacCartney (2009) where the claim ''*Some birds do not fly*'' is entailed by the evidence ''*Not all birds fly*''. Due to NatLog's limitations, the most fine-grained correct proof would be to align (*Not all*, *Some do not*), (*birds*, *birds*) and (fly, fly) to produce the proof S $\xrightarrow{\equiv}$ S $\xrightarrow{\equiv}$ S $\xrightarrow{\equiv}$ S ; thus the reasoning between the negations and quantifiers in the aligned pair required to arrive at the set-theoretic relation is omitted from the proof itself. Therefore, the proofs of TabVer are not necessarily fully comprehensive explanations, as they do not fully explain the production of the proof.

Moreover, proofs of TabVer do not allow assigning NatOp sequences to individual claim spans, such as *cat* → *dog* → *poodle*, which can be a limitation for multi-hop claims where multiple pieces of evidence from one or more tables have to be combined for a single span beyond arithmetic functions. Furthermore, proofs are produced and executed from left to right. However, NatLog does not impose such constraints and is instead non-deterministic by design. This can lead to inconsistencies, as the rearrangement of a NatOp sequence O can lead to differently informative veracity predictions (MacCartney and Manning, 2009; Angeli et al., 2016). For instance, consider a variation of the running example shown in Figure 5: ''*In 2018, Ortegal had three municipalities and a population larger than 12,000.*''. Assuming the same NatOp relations are assigned, an NEI verdict would be produced: S $\xrightarrow{\equiv}$ S $\xrightarrow{⇃↾}$ R $\xrightarrow{\sqsubseteq}$ N. TabVer mitigates this issue via two mechanisms: (i) using claim decomposition to avoid long proofs where such phenomena occur, and (ii) considering multiple proof candidates at different granularity levels,

following Aly et al. (2023) (e.g., *three municipalities and a population larger than 12,000* could be considered a single span with the ⇵ NatOp). As shown in Figure 5, by breaking the original claims into atomic units of information, the individual subclaim verdicts (via DFA transitions S $\xrightarrow{\equiv}$ S $\xrightarrow{\equiv}$ S $\xrightarrow{⇵}$ R and S $\xrightarrow{\equiv}$ S $\xrightarrow{\equiv}$ S $\xrightarrow{\sqsubseteq}$ S for subclaim 1 and 2, respectively) aggregate into the correct overall verdict. Both mechanisms also help in dealing with complex, multi-clause claims, where multiple erroneous and independent facts can lead to NEI predictions (double ↓↑)—another weak point of natural logic's nondeterministic composition of NatOps.

7 Conclusion

This paper presented TabVer, a natural logic inference system that adds arithmetic reasoning capabilities for few-shot fact verification on tables. We presented a set-theoretic definition between numerals in a claim and answers calculated on evidence via ArithExps. We proposed a method for leveraging LLMs to generate ArithExps via claim-aware question generation and rationale-guided question answering with constrained decoding. TabVer outperforms all baseline systems on FEVEROUS and in a domain-transfer setting on Tabfact, highlighting our model's generalizability. We show that TabVer has learned a nuanced understanding of numerals, more sensitive to the context of a claim than other baselines. Future work investigates natural logic for scalar implicature on diverse datasets with different requirements for numerical precision.

Acknowledgments

This work was supported by the Engineering and Physical Sciences Research Council Doctoral Training Partnership (EPSRC). Andreas Vlachos is supported by the ERC grant AVeriTeC (GA 865958). The authors would like to thank Sana Kidwai for helpful conversations on linguistic concepts and Chenxi Whitehouse for useful discussions and feedback on the paper. We further thank the anonymous reviewers and the action editor Kenji Sagae for their valuable and detailed feedback.

References

Neel Alex, Eli Lifland, Lewis Tunstall, Abhishek Thakur, Pegah Maham, C. Jess Riedel, Emmie Hine, Carolyn Ashurst, Paul Sedille, Alexis Carlier, and others. 2021. RAFT: A real-world few-shot text classification benchmark. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Rami Aly, Zhijiang Guo, Michael Sejr Schlichtkrull, James Thorne, Andreas Vlachos, Christos Christodoulopoulos, Oana Cocarascu, and Arpit Mittal. 2021. 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)*. https://doi.org/10.18653/v1/2021.fever-1.1

Rami Aly, Marek Strong, and Andreas Vlachos. 2023. QA-NatVer: Question Answering for Natural Logic-based Fact Verification. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8376–8391, Singapore. Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.emnlp-main.521

Gabor Angeli, Neha Nayak Kennard, and Christopher D. Manning. 2016. Combining natural logic and shallow reasoning for question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 442–452, Berlin, Germany. Association for Computational Linguistics. https://doi.org/10.18653/v1/P16-1042

Gabor Angeli and Christopher D. Manning. 2014. NaturalLI: Natural Logic Inference for common sense reasoning. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 534–545, Doha, Qatar. https://doi.org/10.3115/v1/D14-1059

Pepa Atanasova, Oana-Maria Camburu, Christina Lioma, Thomas Lukasiewicz, Jakob Grue Simonsen, and Isabelle Augenstein. 2023. Faithfulness tests for natural language explanations. In *Proceedings of the 61st*

Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 283–294, Toronto, Canada. Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.acl-short.25

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*, virtual.

Jiangjie Chen, Qiaoben Bao, Changzhi Sun, Xinbo Zhang, Jiaze Chen, Hao Zhou, Yanghua Xiao, and Lei Li. 2022. LOREN: Logic-Regularized Reasoning for Interpretable Fact Verification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10): 10482–10491. Number: 10. https://doi.org/10.1609/aaai.v36i10.21291

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021. Evaluating large language models trained on code. ArXiv:2107.03374v2 [cs].

Wenhu Chen. 2023. Large language models are few(1)-shot table reasoners. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1120–1130. Dubrovnik, Croatia. Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.findings-eacl.83

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*, Online.

Zhoujun Cheng, Tianbao Xie, Peng Shi, Chengzu Li, Rahul Nadkarni, Yushi Hu, Caiming Xiong, Dragomir Radev, Mari Ostendorf, Luke Zettlemoyer, Noah A. Smith, and Tao Yu. 2023. Binding language models in symbolic languages. In *The Eleventh International Conference on Learning Representations (2023)*.

Yew Ken Chia, Pengfei Hong, Lidong Bing, and Soujanya Poria. 2024. InstructEval: Towards holistic evaluation of instruction-tuned large language models. In *Proceedings of the First edition of the Workshop on the Scaling Behavior of Large Language Models (SCALE-LLM 2024)*, pages 35–64, St. Julian's, Malta. Association for Computational Linguistics.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

Angela Fan, Aleksandra Piktus, Fabio Petroni, Guillaume Wenzek, Marzieh Saeidi, Andreas Vlachos, Antoine Bordes, and Sebastian Riedel. 2020. Generating fact checking briefs. In

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 7147–7161, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.emnlp-main.580

Parker Glenn, Parag Pravin Dakle, Liang Wang, and Preethi Raghavan. 2024. BlendSQL: A scalable dialect for unifying hybrid question answering in relational algebra. ArXiv: 2107.03374v2 [cs]. https://doi.org/10.18653/v1/2024.findings-acl.25

Hongfang Gong, Can Wang, and Xiaofei Huang. 2023. Double graph attention network reasoning method based on filtering and program-like evidence for table-based fact verification. *IEEE Access*, 11:86859–86871. https://doi.org/10.1109/ACCESS.2023.3304915

Herbert P. Grice. 1975. Logic and conversation. In *Speech Acts*, pages 41–58. Brill. https://doi.org/10.1163/9789004368811_003

Zihui Gu, Ju Fan, Nan Tang, Preslav Nakov, Xiaoman Zhao, and Xiaoyong Du. 2022. PASTA: Table-operations aware fact verification via sentence-table cloze pre-training. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4971–4983, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.emnlp-main.331

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing. In *The Eleventh International Conference on Learning Representations (2023)*.

Jonathan Herzig, Pawel Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Eisenschlos. 2020. TaPas: Weakly supervised table parsing via pre-training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4320–4333, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.398

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735, PubMed: 9377276

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations (2022)*.

Alon Jacovi and Yoav Goldberg. 2021. Aligning faithful interpretations with their social attribution. *Transactions of the Association for Computational Linguistics*, 9:294–310. https://doi.org/10.1162/tacl_a_00367

C. J. M. Jansen and M. M. W. Pollmann. 2001. On round numbers: Pragmatic aspects of numerical expressions. *Journal of Quantitative Linguistics*, 8(3):187–201. https://doi.org/10.1076/jqul.8.3.187.4095

Paloma Jeretic, Alex Warstadt, Suvrat Bhooshan, and Adina Williams. 2020. Are natural language inference models IMPPRESsive? Learning IMPlicature and PRESupposition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8690–8705, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.768

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7B. ArXiv:2310.06825v1 [cs].

Nengzheng Jin, Joanna Siebert, Dongfang Li, and Qingcai Chen. 2022. A survey on table question answering: Recent advances. In *Knowledge Graph and Semantic Computing: Knowledge Graph Empowers the Digital Economy*, pages 174–186, Singapore. Springer Nature. https://doi.org/10.1007/978-981-19-7596-7_14

Mayank Jobanputra. 2019. Unsupervised question answering for fact-checking. In *Proceedings of the Second Workshop on Fact Extraction and VERification (FEVER)*, pages 52–56,

Hong Kong, China. Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-6609

Lauri Karttunen. 2015. From natural logic to natural reasoning. In *Computational Linguistics and Intelligent Text Processing*, pages 295–309, Cham. Springer International Publishing. https://doi.org/10.1007/978-3-319-18111-0_23

Edward L. Keenan. 2017. The quantifier questionnaire, *Handbook of Quantifiers in Natural Language: Volume II*, pages 1–20, Springer. https://doi.org/10.1007/978-3-319-44330-0_1

Kezhi Kong, Jiani Zhang, Zhengyuan Shen, Balasubramaniam Srinivasan, Chuan Lei, Christos Faloutsos, Huzefa Rangwala, and George Karypis. 2024. OpenTab: Advancing large language models as open-domain table reasoners. In *The twelfth International Conference on Learning Representations (ICLR 2024)*.

Amrith Krishna, Sebastian Riedel, and Andreas Vlachos. 2022. ProoFVer: Natural logic theorem proving for fact verification. *Transactions of the Association for Computational Linguistics*, 10:1013–1030. https://doi.org/10.1162/tacl_a_00503

Peter Lasersohn. 1999. Pragmatic halos. *Language*, 75(3):522–551. Publisher: Linguistic Society of America. https://doi.org/10.2307/417059

Sven Lauer. 2012. On the pragmatics of pragmatic slack. In *Proceedings of SINn und BEDEUtung*, volume 16, pages 389–402.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.703

Chen Liang, Jonathan Berant, Quoc Le, Kenneth D. Forbus, and Ni Lao. 2017. Neural symbolic machines: Learning semantic parsers on freebase with weak supervision. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 23–33, Vancouver, Canada. Association for Computational Linguistics. https://doi.org/10.18653/v1/P17-1003

Chen Liang, Mohammad Norouzi, Jonathan Berant, Quoc V. Le, and Ni Lao. 2018. Memory augmented policy optimization for program synthesis and semantic parsing. In *Advances in Neural Information Processing Systems 31 (2018)*.

Bill Yuchen Lin, Kangmin Tan, Chris Scott Miller, Beiwen Tian, and Xiang Ren. 2022. Unsupervised cross-task generalization via retrieval augmentation. In *Advances in Neural Information Processing Systems 35 (2022)*.

Haokun Liu, Derek Tam, Muqeeth Mohammed, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin Raffel. 2022a. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. In *Advances in Neural Information Processing Systems*.

Qian Liu, Bei Chen, Jiaqi Guo, Morteza Ziyadi, Zeqi Lin, Weizhu Chen, and Jian-Guang Lou. 2022b. TAPEX: Table Pre-training via Learning a Neural SQL Executor. In *International Conference on Learning Representations (2022)*.

Qian Liu, Fan Zhou, Zhengbao Jiang, Longxu Dou, and Min Lin. 2023. From zero to hero: Examining the power of symbolic tasks in instruction tuning. ArXiv:2304.07995v1 [cs].

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations (2019)*.

Bill MacCartney. 2009. *Natural language inference*. Ph.D. thesis, Stanford University.

Bill MacCartney and Christopher D. Manning. 2007. Natural logic for textual inference. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 193–200, Prague. Association for Computational Linguistics. https://doi.org/10.3115/1654536.1654575

Bill MacCartney and Christopher D. Manning. 2009. An extended model of natural logic. In *Proceedings of the Eight International Conference on Computational Semantics*, pages 140–156, Tilburg, The Netherlands. Association for Computational Linguistics. https://doi.org/10.3115/1693756.1693772

Bill MacCartney and Christopher D. Manning. 2014. Natural logic and natural language inference. In *Computing Meaning: Volume 4*, pages 129–147, Springer. https://doi.org/10.1007/978-94-007-7284-7_8

Suixin Ou and Yongmei Liu. 2022. Learning to generate programs for table fact verification via structure-aware semantic parsing. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7624–7638, Dublin, Ireland. Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.acl-long.525

Daniele Panizza, Gennaro Chierchia, and Charles Clifton. 2009. On the role of entailment patterns and scalar implicatures in the processing of numerals. *Journal of Memory and Language*, 61(4):503–518. https://doi.org/10.1016/j.jml.2009.07.005, PubMed: 20161494

Anku Rani, S. M. Towhidul Islam Tonmoy, Dwip Dalal, Shreya Gautam, Megha Chakraborty, Aman Chadha, Amit Sheth, and Amitava Das. 2023. FACTIFY-5WQA: 5W Aspect-based Fact Verification through Question Answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10421–10440, Toronto, Canada. Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.acl-long.581

Qi Shi, Yu Zhang, Qingyu Yin, and Ting Liu. 2020. Learn to combine linguistic and symbolic information for table-based fact verification. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5335–5346, Barcelona, Spain (Online). International Committee on Computational Linguistics. https://doi.org/10.18653/v1/2020.coling-main.466

Benjamin Spector. 2013. Bare numerals and scalar implicatures. *Language and Linguistics Compass*, 7(5):273–294. https://doi.org/10.1111/lnc3.12018

Lya Hulliyyatus Suadaa, Hidetaka Kamigaito, Kotaro Funakoshi, Manabu Okumura, and Hiroya Takamura. 2021. Towards table-to-text generation with numerical reasoning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1451–1465, Online. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. Arxiv:2307.09288v2 [cs].

Andreas Vlachos and Sebastian Riedel. 2015. Identification and verification of simple claims about statistical properties. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2596–2601, Lisbon, Portugal. https://doi.org/10.18653/v1/D15-1312

Haoran Wang and Kai Shu. 2023. Explainable claim verification via knowledge-grounded reasoning with large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6288–6304, Singapore. Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.findings-emnlp.416

Zilong Wang, Hao Zhang, Chun-Liang Li, Julian Martin Eisenschlos, Vincent Perot, Zifeng Wang, Lesly Miculicich, Yasuhisa Fujii, Jingbo Shang, Chen-Yu Lee, and Tomas Pfister. 2024. Chain-of-table: Evolving tables in the reasoning chain for table understanding. ArXiv:2401.04398v2 [cs].

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems 36*.

Greg Woodin, Bodo Winter, Jeannette Littlemore, Marcus Perlman, and Jack Grieve. 2024. Large-scale patterns of number use in spoken and written English. *Corpus Linguistics and Linguistic Theory*, 20(1):123–152. Publisher: De Gruyter Mouton. https://doi.org/10.1515/cllt-2022-0082, PubMed: 38344039

Zirui Wu and Yansong Feng. 2024. ProTrix: Building models for planning and reasoning over tables with sentence context. ArXiv:2403.02177v2 [cs].

Xiaoyu Yang, Feng Nie, Yufei Feng, Quan Liu, Zhigang Chen, and Xiaodan Zhu. 2020. Program enhanced fact verification with verbalization and graph attention network. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7810–7825, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.emnlp-main.628

Xiaoyu Yang and Xiaodan Zhu. 2021. Exploring decomposition for table-based fact verification. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1045–1052, Punta Cana, Dominican Republic. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.findings-emnlp.90

Yunhu Ye, Binyuan Hui, Min Yang, Binhua Li, Fei Huang, and Yongbin Li. 2023. Large language models are versatile decomposers: Decomposing evidence and questions for table-based reasoning. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '23, pages 174–184, New York, NY, USA. Association for Computing Machinery. https://doi.org/10.1145/3539618.3591708

Pengcheng Yin, Graham Neubig, Wen-tau Yih, and Sebastian Riedel. 2020. TaBERT: Pretraining for joint understanding of textual and tabular data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8413–8426, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.745

Hangwen Zhang, Qingyi Si, Peng Fu, Zheng Lin, and Weiping Wang. 2024a. Are large language models table-based fact-checkers? ArXiv:2402.02549v1 [cs].

Tianshu Zhang, Xiang Yue, Yifei Li, and Huan Sun. 2024b. TableLlama: Towards open large generalist models for tables. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6024–6044, Mexico City, Mexico. Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.naacl-long.335

Yunjia Zhang, Jordan Henkel, Avrilia Floratou, Joyce Cahoon, Shaleen Deep, and Jignesh M. Patel. 2024c. ReAcTable: Enhancing ReAct for table question answering. *Proceedings of the VLDB*, 14(1). https://doi.org/10.14778/3659437.3659452

Xiaoyan Zhao, Lingzhi Wang, Zhanghao Wang, Hong Cheng, Rui Zhang, and Kam-Fai Wong. 2024. PACAR: Automated fact-checking with planning and customized action reasoning using large language models. In *Proceedings*

of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), pages 12564–12573, Torino, Italia. ELRA and ICCL.

Wanjun Zhong, Duyu Tang, Zhangyin Feng, Nan Duan, Ming Zhou, Ming Gong, Linjun Shou, Daxin Jiang, Jiahai Wang, and Jian Yin. 2020. LogicalFactChecker: Leveraging logical operations for fact checking with graph module network. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6053–6065, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.539

Alex Zhuang, Ge Zhang, Tianyu Zheng, Xinrun Du, Junjie Wang, Weiming Ren, Wenhao Huang, Jie Fu, Xiang Yue, and Wenhu Chen. 2024. StructLM: Towards building generalist models for structured knowledge grounding. In *First Conference on Language Modeling*. Philadelphia, USA.

A Method Details

Table 6 shows the set-theoretic definitions of NatOps $o \in \mathbb{O}$. The effect of environments on the entailment relations is modelled in natural logic via projection functions $\rho : \mathbb{O} \rightarrow \mathbb{O}$ (MacCartney and Manning, 2009). The upward-entailing environment is the default environment with the projection function being the identity. Table 7 shows the projection function $\rho_{\downarrow}$ for downward-entailing environments. This projection function can now be further modified in the context of a numeral's reading in such an environment, following Panizza et al. (2009). Consider the following example: In an upward-entailing environment, the relationship $3 \Updownarrow 4$ holds, i.e., the numbers 3 and 4 are assigned the alternation NatOp following the upper-bounded reading in Section 3.1. However, in a downward-entailing environment, we see that *Everybody who scored 3 goals received a bonus* $\sqsubseteq$ *Everybody who scored 4 goals received a bonus* holds instead, since the numbers have an *at least* interpretation without further specification, following Panizza et al. (2009). The projection function $\rho_{\text{num}\downarrow}$ from an upward-entailing to a downward-entailing environment that results from such an *at least* reading of numerals is shown in Table 7. The prompt templates for M_{QG} and M_{QA} are shown in Listing 1 and 2, respectively.

NatOp	Set-theoretic
Equivalence ($\equiv$)	$x = y$
Frw. Entailment ($\sqsubseteq$)	$x \subset y$
Rev. Entailment ($\sqsupseteq$)	$x \supset y$
Negation ($\curlywedge$)	$x \cap y = \emptyset \wedge x \cup y = U$
Alternation ($\Updownarrow$)	$x \cap y = \emptyset \wedge x \cup y \neq U$
Independence (#)	All other cases

Table 6: Natural logic operators (NatOps) and their set-theoretic definitions.

o	$\equiv$	$\sqsubseteq$	$\sqsupseteq$	$\Updownarrow$	$\curlywedge$	#
$\rho_{\downarrow}(o)$	$\equiv$	$\sqsupseteq$	$\sqsubseteq$	$\smile$	$\curlywedge$	#
$\rho_{\text{num}\downarrow}(o)$	$\sqsubseteq$	$\sqsupseteq$	$\sqsubseteq$	$\sqsubseteq, \sqsupseteq$	$\smile$	#

Table 7: A projection $\rho_{\text{num}\downarrow}$ for NatOps concerning numerals for downward-entailing environments. The cover ($\smile$) NatOp is shown here for completeness only.

Training & Hyperparameters We fine-tune the question generation M_{QG} and question answering M_{QA} model using default hyperparameters. Specifically, we use a learning rate of 2^{-4} and train for a total of 10 epochs across all models and experiments. The maximum generation length for M_{QG} is set to 100 tokens for the generation of question, and the constraint answer selection is set to any-length span in c. For M_{QA} the maximum length of E_{rel} is 100 tokens between every generated number. We use adamw (Loshchilov and Hutter, 2019) as the optimizer. We use a batch size of 1 during training with gradient accumulation, resulting in an effective batch size of 8. For LoRA, we use a rank $r = 16$ and apply it to the query and value vectors of the attention mechanism. For fine-tuning, we exclude tokens of the prompts from the loss computation that are not part of the gold answer, so we are not fine-tuning the instructions, only the answers that follow after the instruction. For our proof generation model M_P we use the default hyperparameters of QA-NatVer (Aly et al., 2023).

```
"system_description": "Always ensure that each generated question is a single sentence. Ensure to enumerate each question and corresponding answer with the same number.",
"task_description": "For a given claim, generate questions with answers that are found in the claim itself. Ensure that the answer is located exactly as a substring in the claim itself. Generate the least number of questions as needed to verify the key information of the claim.",
"task_description_arithmetic":"For a given claim, generate questions with answers that are found in the claim itself. Only generate questions that require arithmetic reasoning, such as counting, addition, or averaging. Ensure that the answer is located exactly as a substring in the claim itself. Generate the least number of questions as needed to verify the key information of the claim.",
"demo_sep": "\n\n\n",
"context_format": "{INST}\n\nClaim: {}\nQuestions: {}",
"response_template": "\nQuestions:",
```

Listing 1: The prompt template for the question generation model Q_{QG}.

```
"system_description": "Ensure that the answer to each question starts on a new line, separated by a single newline character.",
"task_description": "Read the tables below taken from Wikipedia to answer a question regarding the table. Use only the following functions for arithmetic reasoning and state them explicitly when used: {FUNCTIONS}. If none of these functions is needed for reasoning say N/A.",
"demo_sep": "\n\n\n",
"context_format": "{INST}\nQuestion: {}\nTable: {}\nAnswer: {}",
"answer_format": "Extracted evidence from table: {EVIDENCE} Computation: {COMPUTATION} Result: {ARITHEXP}",
"response_template": "\nAnswer:",
"negative_answer": "N/A",
"negative_evidence": "Not sufficient information found in the table.",
```

Listing 2: The prompt template for the question answering model Q_{QA}.

Property	All	Numerical Subset
Number of claims	2011	521
Claims with more than 1 table	129	36
Supported claims	704 (35%)	178 (34.1%)
Refuted claims	1242 (61.7%)	338 (64.8%)
NEI claims	65 (3.2%)	5 (1%)
Avg. number of rows	14.3	15.1
Avg. number of col	4.82	5.9
Avg. num highlighted cells	4.85	7.3
Function annotations on 160 samples		
Num. COPY	143	–
Num. COMPARATIVES	12	–
Num. COUNT	28	–
Num. SUM	1	–
Num. DIFF	6	–
Num. MIN/MAX	3	–

Table 8: Quantitative characteristics of the tabular FEVEROUS subset.

B Dataset Details

Quantitative characteristics of the tabular subset of FEVEROUS are shown in Table 8. The table further shows the statistics for the function annotations of 160 claims. The claims were sampled randomly and annotated by the authors of the paper as function annotations are made irrespectively of any model, limiting potential biases. Note that annotations are in a multi-label format since multiple functions can be required to verify a single claim.

C Implementation Details

The prompt template for the decomposition model M_D is shown in Listing 3. We use the Huggingface checkpoints for LLama2-7B,[8] MistralOrca-7B,[9] TAPAS,[10] and TAPEX.[11] The PASTA checkpoint is taken from the associated repository.[12] For constrained decoding, we used the library guidance-ai.[13] The Mistral models are licensed under Apache2.0 and Llama2 is licensed under the llama license.[14] Our research is consistent with the licenses' intended use. The models are

[8] https://huggingface.co/meta-llama/Llama-2-7b-chat-hf.

[9] https://huggingface.co/Open-Orca/Mistral-7B-OpenOrca.

[10] https://huggingface.co/google/tapas-large.

[11] https://huggingface.co/microsoft/tapex-large.

[12] https://github.com/ruc-datalab/PASTA.

[13] https://github.com/guidance-ai.

[14] https://github.com/facebookresearch/llama/blob/main/LICENSE.

```
    "system_description": "Ensure to enumerate each subclaim and that each subclaim starts on a new line.",
    "task_description": "Break a claim down into its subclaims. Ensure that each subclaim can stand by itself and does not require the original claim to be understood. Always ensure that each generated subclaim is a single sentence without conjunctive clauses.",
    "demo_sep": "\n\n\n",
    "context_format": "{INST}\n\nClaim: {}\nSubclaims: {}",
    "response_template": "\nSubclaims:",
```

Listing 3: The prompt template for the claim decomposition model M_D.

```
CREATE TABLE jason chambers(row_id int, res text, record text, opponent text, method text, event text, round text)
3 example rows:
SELECT * FROM w LIMIT 3;
row_id  res record  opponent  method  event round
0 win 18 – 5 – 2  dan new submission (rear naked choke) tfc – power fights  1
1 win 17 – 5 – 2  rene gonzalez decision (split)  mainstream mma – cold war n / a
2 loss  16 – 5 – 2  tristan yunker  submission ( armbar ) tfc 7 – total fight challenge 7 1
*/
Q: in mac – midwest absolute challenge , the player be defeat by dan spychalski in 1 round
```

Listing 4: Example input for Binder. A Wikipedia table is cast to a single SQL table by treating the first table row as the header, rendering the use of Binder for complex tables with nested headers or complex structures suboptimal.

intended for use in English. All experiments are run on a single Quadro 8000 with 48GB memory. To fine-tune M_P with a Flan-T5-3B backbone we use a single A100 80GB.

Baselines We use the available implementations for LPA,[15] SASP,[16] and Binder.[17] Identically to Tabfact, all three models consider the the first table row of a FEVEROUS table as the header row. LPA was trained for 20 epochs since the default number of training epochs (10) was not sufficient to reach convergence. Binder uses the default hyperparameters as specified for Tabfact, but we use 4 instead of 18 in-context examples as MistralOrca, with a full number of in-context examples, produced empty answers very frequently. We hypothesise that MistralOrca generates the end of text token too early since the OpenOrca dataset, on which MistralOrca7B has been instruction-tuned, consists of gold answers which are in 93% of the cases shorter than 2.5K tokens. We train DeBERTa, PASTA, TAPEX, and TAPAS using the HuggingFace Trainer for 10 (100) epochs with full (few-shot) data and a learning rate of 1×10^{-5}.[18] To sanity check our training pipeline, we trained TAPAS in a full supervision setting on TabFact's 92,283 training instances, achieving a score of 82.1 accuracy points versus 81.59 points via the official model checkpoint.[19] To fine-tune the LLM baselines with LoRA, we use Huggingface's SFTTrainer.[20]

D Reading of Numerals Probe - Details

We construct the probing dataset by first filtering instances from the FEVEROUS evaluation data labeled as Supported that contain numbers, excluding dates (e.g., 1939) but including percentages and floating point numbers. Afer a further manual inspection a total 91 claims remain. For each claim, we generate 17 variations of a numeral x:

(1) $x + 1$ (Adding one)
(2) $x + x * 0.02$ (Adding 2%)
(3) $x - x * 0.02$ (Subtracting 2%)
(4) $x + x * 0.1$ (Adding 10%)
(5) $x - x * 0.1$ (Subtracting 10%)
(6) $x + x * 0.25$ (Adding 25%)
(7) $x - x * 0.25$ (Subtracting 25%)
(8) rounding via closest number to x that satisfies: $\frac{x'}{1*10^y} \leqslant 9$ (10-ness)
(9) rounding via closest number to x that satisfies: $\frac{x'}{5*10^y} \leqslant 9$ (5-ness)

[15] https://github.com/wenhuchen/Table-Fact-Checking.
[16] https://github.com/ousuixin/SASP.
[17] https://github.com/xlang-ai/Binder.
[18] https://huggingface.co/docs/transformers/en/main_classes/trainer.
[19] https://huggingface.co/google/tapas-large-finetuned-tabfact.
[20] https://huggingface.co/docs/trl/en/sft_trainer.

(10) rounding via closest number to x that satisfies: $\frac{x'}{2.5 * 10^y} \leqslant 9$ (2.5-ness)

(11) (About|Around|Approximately) + 10-ness (Modifier 10-ness)

(12) (About|Around|Approximately) + 5-ness (Modifier 5-ness)

(13) (About|Around|Approximately) + 2.5-ness (Modifier 2.5-ness)

(14) 'At most' + (Subtracting 10%) (Cardinal at most, incorrect)

(15) 'At least' + (Adding 10%) (Cardinal at least, incorrect)

(16) 'At most' + (Adding 10%) (Cardinal at most, correct)

(17) 'At least' + (Subtracting 10%) (Cardinal at least, correct)

Rounding to numbers that satisfy the 10-ness, 5-ness, and 2.5-ness property follows the empirical observation by Jansen and Pollmann (2001) that round numbers satisfying this arithmetic property occur more frequently than round numbers that do not. For instance, the number 1010 does not satisfy either 10-ness, 5-ness, or 2.5-ness and would generally be considered an atypical way of rounding (1000 would most likely be more natural). We follow the terminology of Keenan (2017) to describe the modifiers *at most* and *at least* as cardinal determiners.

We categorize these variations into the numerical classes shown in Table 3 as follows: Inaccuracy Δ +1: Variation 1.

Inaccuracy Δ 2%: Average of Variation 2 + 3.

Inaccuracy Δ 10%: Average of Variation 4 + 5.

Inaccuracy Δ 25%: Average of Variation 6 + 7.

Rounding: Average of Variation 8 + 9 + 10.

Modifiers: Average of Variation 11 + 12 + 13.

Cardinal (incorrect): Average of Variation 14 + 15.

Cardinal (correct): Average of Variation 16 + 17.

Claim: John McCain won 16 delegates during the 2008 Washington Republican presidential primary, which is eight more than Mike Huckabee won.
Evidence: 2008 United States presidential election in Washington (state). Caucuses Republican caucuses and primary

Candidate	State delegate	Percentage	Delegates	Counties carried
John McCain	3,228	25.9%	16	11
Mike Huckabee	2,959	23.52%	8	11
...	...	..	..	...

Generated Subclaim 1: John McCain won 16 delegates during the 2008 Washington Republication presidential primary.
Generated Subclaim 2: Mike Huckabee won eight fewer delegates than John McCain.

Binder (SQL):
Subclaim 1: SELECT ((SELECT SUM(delegates) FROM w WHERE candidate = "john mccain") = 16
Predicted Verdict: SUPPORTS
Subclaim 2: SELECT ((SELECT SUM(delegates) FROM w WHERE candidate = "mike huckabee") - (SELECT SUM(delegates) FROM w WHERE candidate = "john mccain")) = -8
Predicted Verdict: SUPPORTS **Overall Verdict**: SUPPORTS

TabVer: Subclaim 1:

Generated Questions: 1. How many delegates did John McCain win during the 2008 Washington Republican presidential primary? 1. Associated claim Span (c_i): 16
Question Answering: John McCain won 16 delegates in the 2008 Washington Republican presidential primary. Computation: No computation is required. ArithExp (a_i): COPY 16
Generated Proof:

Claim Span	Evidence Span	NatOP	DFA State
John McCain	John McCain	=	S
won	won	=	S
16 delegates	COPY 16	=	S
during the 2008 Washington	in the 2008 Washington		
Republican presidential primary	Republican presidential primary	=	S

Predicted verdict: SUPPORTS

Subclaim 2:
Generated Questions: 1. How many fewer delegates did Mike Huckabee win than John McCain at the 2008 Washington Republican presidential primary? 1. Associated claim span (c_i): eight
Question Answering: Mike Huckabee won 8 delegates and John McCain won 16 delegates. Computation: 8 - 16 = -8. ArithExp(a_i): COMP -8

Claim Span	Evidence Span	NatOP	DFA
Mike Huckabee won			
eight fewer delegates than John McCain	COMP -8	=	S
at the 2008 Washington	in 2008 Washington		
Republican presidential primary	Republican presidential primary	=	S

Predicted verdict: SUPPORTS **Overall verdict:** SUPPORTS

Figure 6: Output produced by TabVer. For comparison, we show Binder's output. While more compact, Binder's SQL query is less readable and omits relevant context from its query (2008 Washington primary).

Claim: Asiacom Philippine, Inc. stands as the major shareholder of Globe Telecom, with 50.85% of total shares.
Evidence: Globe Telecom. Ownership.

Major Shareholder	% of Total	Common Shares	Preferred Shares
Ayala Corporation	13.20%	41,157,276	—
SingTel Group	20.10%	62,646,487	—
Asiacom Philippines, Inc.	50.85%	—	158,515,016

Generated Subclaim 1: Asiacom Philippine, Inc. stands as the major shareholder of Global Telecom.
Generated Subclaim 2: Asiacom Philippine, Inc. has 50.85% of total shares.

Binder (SQL):
Subclaim 1: SELECT (SELECT major shareholder FROM w WHERE row_id = 2) = "Asiacom Philippine, Inc.")
Predicted Verdict: REFUTES
Subclaim 2: SELECT (SELECT major shareholder, % of total FROM w WHERE row_id = 2) = ("Asiacom Philippine, Inc.", "50.85%")
Predicted Verdict: REFUTES **Overall Verdict**: REFUTES

TabVer:

Subclaim 1:
Generated Questions: 1. Who is the major shareholder of Globe Telecom? 1. Associated claim Span (c_i): Asiacom Philippine, Inc
Question Answering: Ayala Corporation, SingTel Group, and Asiacom Philippines, Inc. are major shareholders of Globe Telecom. Computation: No computation is required. ArithExp (a_i): FILTER Ayala Corporation, SingTel Group, and Asiacom Philippines, Inc.
Generated Proof:

Claim Span	Evidence Span	NatOP	DFA
Asiacom Philippine, Inc	FILTER Ayala Corporation, SingTel Group		
stands as the major shareholder	, and Asiacom Philippines, Inc.	$<$	S
of Globe Telecom	of Globe Telecom	=	S

Predicted verdict: SUPPORTS

Subclaim 2:
Generated Questions: 1. What percentage of total shares does Asiacom Philippine, Inc. have? 1. Associated claim span (c_i): 50.85%
Question Answering: Asiacom Philippines, Inc. has 50.85% of total shares. No computation is required. ArithExp(a_i): FILTER 50.85%

Claim Span	Evidence Span	NatOP	DFA state
Asiacom Philippine, Inc.	Asiacom Philippines, Inc.	=	S
has 50.85%	FILTER 50.85%	=	S
of total shares	of total shares	=	S

Predicted verdict: SUPPORTS

Overall verdict: SUPPORTS

Figure 7: Output produced by TabVer. For comparison, we show Binder's output. In contrast to TabVer in its natural logic formulation, Binder's query is unable to handle probabilistic enrichment of natural language, for instance, due to typing errors (e.g., Asiacom Philippines, Inc.).