ClaimIt: Finding Convincing Views to Endorse a Claim

Shunit Agmon
Technion, Israel
shunita@campus.technion.ac.il

David Avigdor
Technion, Israel
davidavigdor40@gmail.com

Brit Youngmann
Technion, Israel
brity@technion.ac.il

Amir Gilad
Hebrew University, Israel
amirg@cs.huji.ac.il

Benny Kimelfeld
Technion, Israel & RelationalAI
bennyk@cs.technion.ac.il

ABSTRACT

The demonstration presents ClaimIt — a tool for extracting views that support a user-provided claim. Such views can assist users in finding evidence of phenomena of interest, criticizing given claims by proposing opposing viewpoints, inspecting the robustness of statements with respect to subpopulations, and so on. To be useful, the view should constitute a "natural" characterization of a significant subpopulation. In a recently published work, we focused on claims that compare groups by an aggregate query, and explored the measurement of naturalness as well as the algorithmic challenge of handling the plenitude of possible views. ClaimIt realizes the framework as an interactive system that enables users to phrase their claims in a convenient user interface, extract supporting views, sort them by different measures of naturalness, and control the weights of individual measures in a global ranking function. In the demonstration of ClaimIt, the audience will be able to suggest and analyze different claims on various datasets.

PVLDB Reference Format:
Shunit Agmon, David Avigdor, Brit Youngmann, Amir Gilad, and Benny Kimelfeld. ClaimIt: Finding Convincing Views to Endorse a Claim. PVLDB, 18(12): 5331 - 5334, 2025.
doi:10.14778/3750601.3750664

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/TechnionTDK/claim-endorsement-demo.

1 INTRODUCTION

While data is used to qualify stated claims, data-based arguments may also be misleading. For example, Simpson's Paradox implies that basing a claim on the general population may miss the validity of the claim on important subpopulations. On the other direction, cherrypicking refers to basing the claim on a query that may seem natural but involves minor conditions that are crucial for supporting the claim, rather than its opposite. Within the area of fact checking [5], several studies proposed ways of assessing whether a given claim, commonly phrased as an aggregate query over a database, is cherry-picked [2, 6]. In recent work [1], we studied the problem of *claim endorsement*, taking the reverse direction: given a claim that is false in the database, find *natural* views of the data where the claim holds. Effective claim endorsement can help users relate their statements to data, better understand the mechanism of cherry-picking, find queries that contradict or weaken a stated claim, assess how robust a given claim is, assess how amenable a given dataset is to supporting contradictory claims, and so on.

For illustration, consider the Stack Overflow Developers Survey dataset[1] with information about Hi-Tech workers. There, the average salary of people with a master's degree is *lower* than that of people with only a bachelor's degree. Claim endorsement can point out that the opposite is true for people in the field of Data Science and Machine Learning, and for people in Germany. This is also the case for subpopulations with less compelling characterization, such as people who use Zoom for office communication, and people who do not know their organization's size. This illustrates that an effective view that endorses the claim should capture a subpopulation that is significant and characterized by a *natural* query.

As the notion of a "natural" view is subjective, our study explored various measures of naturalness and conducted a user study that compared the ability of the measures to convince people and capture their intuition [1]. The main technical challenge that we addressed is the high computational cost of claim endorsement: there may be prohibitively many candidate views, and each may require costly computation to determine its validity and score of naturalness. Yet, responsiveness is critical in real-time claim endorsement, like the system we describe in this demonstration. In [1], we devised *anytime algorithms* that target the incremental generation of high-quality views (called *refinements* later on) from the very beginning.

In this demonstration, we introduce ClaimIt—an interactive system that operationalizes the claim-endorsement framework of [1], along with its algorithmic solutions. The frontend offers a user-friendly interface that empowers users to articulate claims effortlessly, extract relevant supporting refinements, and sort suggestions based on various measures of naturalness. Users can adjust the weights of individual measures, enabling customization of the global ranking to suit specific needs or preferences. ClaimIt does not require prior knowledge of databases and SQL. During the demonstration, attendees will have the opportunity to engage with ClaimIt by proposing and exploring a variety of claims across diverse datasets, gaining valuable insights into the framework and the measurement of naturalness, as well as the system's functionality and its ability to provide data-driven support for statements.

Proceedings of the VLDB Endowment, Vol. 18, No. 12 ISSN 2150-8097.
doi:10.14778/3750601.3750664

A companion video for this submission is available at https://youtu.be/k2KxyeHzRq8.

[1] https://survey.stackoverflow.co/2022 (accessed Jul 2025)

2 CLAIM ENDORSEMENT

We first describe the framework [1]. We have a database D that consists of a single relationwith the relation name R and the attribute set Att(D)=$\{A_1, \ldots, A_q\}$. By $D[A_{i_1}, \ldots, A_{i_\ell}]$ we denote the (set-semantics) projection of D to the attributes $A_{i_1}, \ldots, A_{i_\ell}$. By a slight abuse of notation, for a single attribute A_i we may view $D[A_i]$ as the set of *values* rather than single-value *tuples*.

Additionally, we have a group-by-aggregate query Q of the form

$$\text{SELECT } A_{gb}, \alpha(A_{agg}) \text{ FROM D WHERE } \phi \text{ GROUP BY } A_{gb} \quad (1)$$

where $A_{gb} \in \text{Att}(D)$ is the group-by attribute, $A_{agg} \in \text{Att}(D)$ is the aggregate attribute, and α is an aggregate function among Count, Sum, Average, Median, Min and Max. The result of Q on D, denoted $Q(D)$, is a set of tuples of the form (g, v), where $g \in D[A_{gb}]$ and

$$v = \alpha(\{t.A_{agg} \mid t \in D \wedge \phi(t) \wedge t.A_{gb} = g\}) .$$

For example, consider the sample of the Stack Overflow dataset-shown in Table 1. An analyst may be interested in justifying a Master's degree, so she issues the following query Q:

```
SELECT EducationLevel, Average(Salary)
FROM D
GROUP BY EducationLevel;
```

In $Q(D)$, we find that the average income for people with a Master's degree ($78.7K) is lower than that of a Bachelor's degree ($82K).

We consider the case where the analyst restricts attention to the relationship between two groups of interest, g_1 and g_2, in the result $Q(D)$.For these, the analyst may be interested in endorsing a specific claim. We define it formally as follows. Given two tuples (g_1, v_1) and (g_2, v_2) in $Q(D)$, a *claim* is the tuple $\kappa = (g_1, g_2, >)$. A group-aggregate query Q' *endorses* claim κ (on D) if, in the result $Q'(D)$, group g_1 is associated with a higher numeric value than g_2, that is, for two numbers v_1' and v_2' it holds that $(g_1, v_1') \in Q'(D)$ and $(g_2, v_2') \in Q'(D)$ and $v_1' > v_2'$. We consider the situation where Q violates $(g_1, g_2, >)$, and seek a *refinement* Q' that satisfies it. We focus on refinements that add predicates to the WHERE clause (as done previously, e.g., [6]).

Continuing the running example, the analyst wishes to compare the average income for different degree holders, with the initial assumption that a higher degree implies a higher salary. Yet, she finds that the average income for people with a Master's degree ($78.7K) is actually *lower* than that of people with a Bachelor's degree ($82K). In our formalism, the analyst is interested in the relationship between g_1 = Master's and g_2 = Bachelor's and their corresponding values in $Q(D)$, namely $v_1 = 78.7$ and $v_2 = 82$. Hence, the claim (Master's, Bachelor's, >) is violated by Q.

In a recent work [1], we studied the problem of searching for query refinements. To that end, we assume a space $\mathcal{P}$ of predicates that can be used to refine the query Q. We consider equality predicates A=v as *atoms* or atomic predicates, and we focus on conjunctions of up to m such atoms, where m is a parameter. We are given a set of attributes that does not include A_{gb} and A_{agg}, to be used in the atomic predicates. Let Q be a group-aggregate query as in Equation (1), and let $p \in \mathcal{P}$ be a predicate. The *refinement of Q by p* is the query Q_p where ϕ is replaced by $\phi' = \phi \wedge p$.

As aforesaid, in our running example the query Q violates the claim (Master's, Bachelor's, >). Consider the predicate p_1 given by the expression OpSys = Linux (people who use Linux-based operating systems at work). In contrast to Q, the refinement Q_{p_1} satisfies the claim: the average income for Master's degree holders is $87K, yet only $82K for Bachelor's. Another possible refinement is defined by the predicate p_2=YearsCode="0-5" (developers with little coding experience), where v_1=87 and v_2=64.

Table 1: Sample of the Stack Overflow dataset.

ID	YearsCode	OpSys	EducationLevel	Salary (K)
1	10-15	Linux	Bachelor's degree	100
2	5-10	Windows	Master's degree	49
3	0-5	Linux	Bachelor's degree	64
4	0-5	Linux	Master's degree	87
5	10-15	Windows	Master's degree	100

Naturalness measures. Supporting the analyst claim can be performed by finding a certain refinement where the claim holds. However, this refinement should be *natural* in the sense that it should not be overly specific and restricted. For example, "Developers from Thailand who use Cisco Webex Teams for office communication" is, arguably, overly specific and can hardly serve as significant support for the claim. To this end, a *naturalness measure* (for a query Q and a claim κ) is a function ν that maps pairs (Q_p, D), where Q_p is a refinement and D is a database, to a numerical score $\nu(Q_p, D)$. A higher score for Q_p than for Q_p' means that Q_p is considered more natural than Q_p' for the database D. Intuitively, ν aims to quantify (or be well correlated with) the likelihood of a critical listener accepting the claim if it is presented with this refinement.

We focused on specific measures of naturalness [1], designed to capture diverse intuitive aspects of naturalness. Suppose that p is defined using the attributes $A_1, \ldots, A_\ell$. **Coverage:** The coverage of a refinement Q_p is the fraction of database tuples covered by $\phi \wedge p$. **Embedding similarity:** Word embeddings well capture the semantics of text [3]. The cosine similarity between the embedding of the predicate (treated as text) and that of the target attribute measures their relatedness. **Statistical significance:** *Hypothesis testing* determines whether the difference between group values is significant and indicative of an actual phenomenon. This measure is defined when α is Average (two-sided independent T-test) or α = Median (median test [8] with Yates correction [10]). The score is the complement of the p-value. **Mutual information (MI):** Arguably, relevant predicates involve attributes with some dependence on the target attribute. MI quantifies this correlation. We use MI between the attribute list $(A_1, \ldots, A_\ell)$ that defines p and the target attribute A_{agg}. **ANOVA:** Analysis of Variance (ANOVA) also quantifies the dependence between $(A_1, \ldots, A_\ell)$ and A_{agg}. The attribute combination induces a partition of A_{agg} values into groups, each associated with a value combination. ANOVA measures the uniformity of the means of A_{agg} among the groups.

Fix a predicate space $\mathcal{P}$ and a collection of naturalness measures. Claim Endorsement aims at finding the most natural refinements of Q than endorse the claim κ over the database D. Often, we do not wish to select one naturalness measure but rather *combine* several measures. For example, we might wish to retrieve the top-k refinements according to each naturalness measure and then examine all of them. Our user study [1] indicates that statistical significance and coverage are most aligned with participants intuitions.

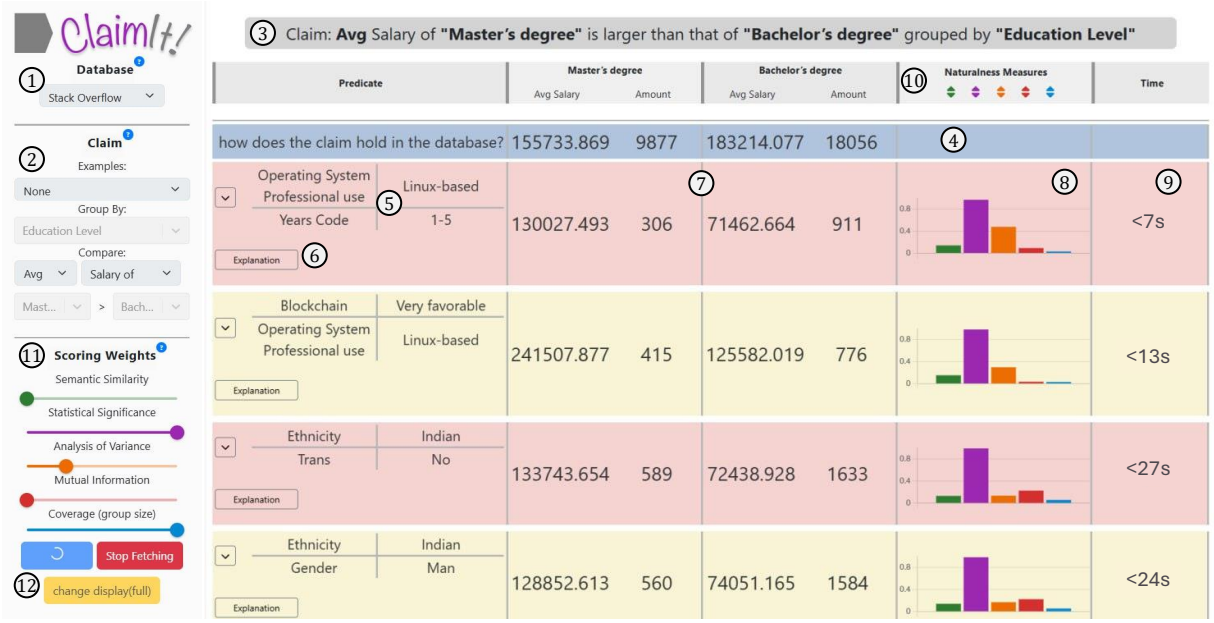

Figure 1: User interface of ClaimIt.

3 SYSTEM OVERVIEW

We provide an overview of the implementation of ClaimIt. Full details and algorithms can be found in [1]. We implemented ClaimIt using Python and Node.js for the backend and React for the UI. We used Google Gemini [9] to generate English explanations.

3.1 Endorsement Search

The main technical challenge is the high computational cost: refinements can be made out of many attributes, attribute combinations, and value assignments; moreover, each candidate refinement may require costly computation to verify its correctness and measure its naturalness. Efficiency is critical in data exploration, where users interactively react to refinements by formulating new ones.

Instead of materializing all possible candidates, we devise an *anytime algorithm* that targets the incremental generation of high-quality refinements from the very beginning. We instantiate the framework on the aforementioned measures of naturalness. More technically, the algorithms enumerate refinements in a ranked fashion, where the ranking function is, intuitively, well correlated with the naturalness measure, yet efficient to handle.

Our framework deploys several main components, depicted in Figure 2. The *Prioritizer* produces a ranked list of attribute combinations according to an easy-to-compute scoring function for each naturalness measure. The *Merger* merges the ranked lists in an interleaving fashion, without repeating any attribute combinations. Finally, the *Refinement Fetcher* iterates over the attribute combinations in ranked order. For each one, it computes all value assignments (corresponding to supporting refinements) using a single SQL query. The supporting refinements are output as they are found, instead of waiting until all of them are retrieved. As the search progresses, the sum of the top-k naturalness scores increases. We empirically found that the sum increases faster at the beginning of the search, due to the prioritization [1]. Based on this finding, the user can stop the search when they observe that the top-k sum of naturalness scores did not increase for some time. While we have no provable guarantees, we found that it takes up to a minute for the top-25 refinements to stabilize on most of our datasets.

3.2 UI Overview

ClaimIt UI (shown in Fig. 1) is composed of two key elements: a control sidebar on the left, and a result table in the center. On the left sidebar (①), the user selects the database on which they want to impose the claim, out of the databases available in the backend. Each database is associated with an aggregate attribute. The user then formulates the claim (②) by selecting the group-by attribute, the two groups to compare, and the aggregation function (currently supported: Average, Median, and Count). The claim is then previewed at the top center of the screen (③).

After the user inputs their claim, ClaimIt calculates the supporting refinements for the claim, each representing a subpopulation where the claim holds. The claim is first evaluated over the full database and the result is presented in the first row of the table (④). Next, the table is populated with supporting subpopulations. Each subpopulation is defined by a predicate (⑤), composed of up to two attribute-value pairs.[2] A natural-language explanation can be generated by pressing the explanation button (⑥). The number of tuples in each subpopulation and the aggregate values of the groups are also displayed (⑦), along with a bar chart (⑧) showing the naturalness score according to each measure. The rightmost column contains the time it took to find the subpopulation (⑨).

[2]The system supports any number of atoms, but for interactive running times we limit the demonstration to two atoms.

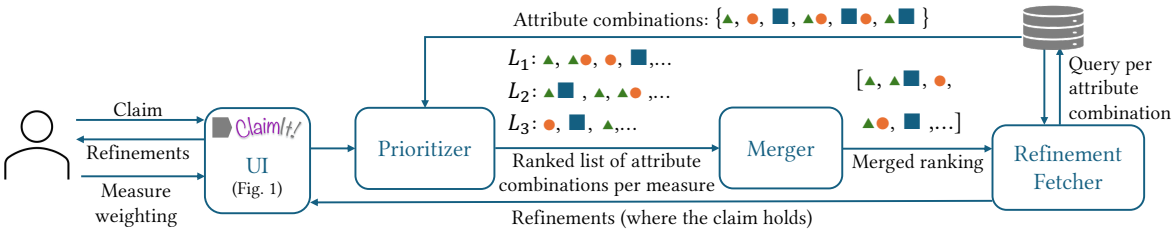

Figure 2: Implementation overview.

Supporting subpopulations are ranked by their average naturalness score. As a result of the merged prioritization described in Section 3.1, the subpopulations with the highest average naturalness are usually fetched at the beginning of the search. The user can then sort the supporting subpopulations by each of the naturalness measures using the arrows above the bar chart column ((10)), or by a weighted combination of them using the sliders on the left sidebar ((11)). Additionally, the user may choose to group the resulting subpopulations by their defining attributes, using a button on the sidebar ((12)). This is useful for diversifying the top subpopulations, in cases where few attribute combinations yield a large number of supporting subpopulations.

4 DEMONSTRATION

We demonstrate ClaimIt over several datasets, including H&M, stack overflow and the American census dataset, which we mention in this section. The participants will act as data analysts, aiming to explore a claim by looking for subpopulations endorsing it.

Navigating ClaimIt. We will first demonstrate basic analysis of claims on several datasets. We will use the system to analyze violated intuitive claims, and search for subpopulations that support the claims. We will show how to formulate the claims using the UI and examine it through the resulting supporting subpopulations. For example, for H&M[3] [7] (15.2M tuples, 19 attributes), we will inspect the difference in number of shopping transactions performed by 25 versus 40 year-olds (which are the highest and lowest points of shopping counts in this dataset, respectively). For this query, we found that while overall, 25-year-olds shop more, the situation is reversed in many categories of children's clothes.

Focusing on the naturalness measures. Next, we will show the role of the naturalness measures in the exploration of the supporting subpopulations. Given a large set of returned subpopulations, we will rank them by the various naturalness measures, and see how each highlights different subsets, enabling the analysts to discover interesting and diverse subpopulations.

We then focus on a specific claim and show how different weighting schemes of naturalness affect the ranking. On the Stack Overflow dataset (38K tuples, 47 attributes), we will inspect the difference in average salary between bachelor's and master's degree holders. We will demonstrate what aspects of naturalness each measure captures. As a reference, we will inspect the degree that the audience agrees with the results of the user study conducted in the full paper [1] regarding the importance of different measures. For example, in Figure 1, when weighting the naturalness measures following the results of the user study, the top predicate describes people with little coding experience, meaning that a master's degree can sometimes replace the role of experience.

Diversifying the supporting subpopulations. Finally, we will inspect claims where there are many similar supporting subpopulations: for example, the gender pay gap as reflected in the American census dataset [4] (1.4M tuples, 120 attributes). A single attribute combination can yield many different subpopulations supporting the claim (e.g., 124 combinations of occupation and weekly work hours); we will show how to use the grouping feature in the system to group subpopulations by their defining attributes, where the group is represented by the predicate with the highest average naturalness score. This enables the analysts to see a more diverse set of interesting supporting subpopulations at a glance and allows them to focus on different aspects of the claim.

[3]https://relbench.stanford.edu/datasets/rel-hm (accessed Jan 2025)

ACKNOWLEDGMENTS

The work of Shunit Agmon and Benny Kimelfeld was funded by the Israel Science Foundation (ISF) under grant 768/19 and the German Research Foundation (DFG) under grant KI 2348/1-1. The work of Amir Gilad was funded by the Israel Science Foundation (ISF) under grant 1702/24 and the Alon Scholarship.

REFERENCES

[1] Shunit Agmon, Amir Gilad, Brit Youngmann, Shahar Zoarets, and Benny Kimelfeld. 2024. Finding Convincing Views to Endorse a Claim. *PVLDB* 18, 2 (2024), 439–452. https://doi.org/10.14778/3705829.3705857
[2] Abolfazl Asudeh, Hosagrahar Visvesvaraya Jagadish, You Wu, and Cong Yu. 2020. On detecting cherry-picked trendlines. *VLDB* 13, 6 (2020), 939–952.
[3] Aylin Caliskan, Joanna J Bryson, and Arvind Narayanan. 2017. Semantics derived automatically from language corpora contain human-like biases. *Science* 356, 6334 (2017), 183–186.
[4] Frances Ding, Moritz Hardt, John Miller, and Ludwig Schmidt. 2021. Retiring Adult: New Datasets for Fair Machine Learning. *Advances in Neural Information Processing Systems* 34 (2021).
[5] Zhijiang Guo, Michael Schlichtkrull, and Andreas Vlachos. 2022. A survey on automated fact-checking. *TACL* 10 (2022), 178–206.
[6] Yin Lin, Brit Youngmann, Yuval Moskovitch, HV Jagadish, and Tova Milo. 2021. On detecting cherry-picked generalizations. *VLDB* 15, 1 (2021), 59–71.
[7] Joshua Robinson, Rishabh Ranjan, Weihua Hu, Kexin Huang, Jiaqi Han, Alejandro Dobles, Matthias Fey, Jan Eric Lenssen, Yiwen Yuan, Zecheng Zhang, et al. 2024. Relbench: A benchmark for deep learning on relational databases. *Advances in Neural Information Processing Systems* 37 (2024), 21330–21341.
[8] Peter Sprent and Nigel C Smeeton. 2007. *Applied nonparametric statistical methods.* CRC press, Chapter 5, 86–111.
[9] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, et al. 2023. Gemini: a family of highly capable multimodal models. (2023).
[10] Frank Yates. 1934. Contingency tables involving small numbers and the χ^2 test. *Supplement to the Journal of the Royal Statistical Society* 1, 2 (1934), 217–235.