

How Can Similarity Be Explained? Modeling Multiple Points of View with Inverse simFOL

Nino PIREAUD ^{a,1}, Victor DAVID ^a, Pierre MONNIN ^a, Elena CABRIO ^a,
Serena VILLATA ^a and Anthony HUNTER ^b

^a *Université Côte d'Azur, Inria, CNRS, I3S, Sophia-Antipolis, France*

^b *University College London*

ORCID ID: Victor David <https://orcid.org/0000-0002-4216-0876>, Pierre Monnin
<https://orcid.org/0000-0002-2017-8426>, Elena Cabrio
<https://orcid.org/0000-0001-9374-7872>, Serena Villata
<https://orcid.org/0000-0003-3495-493X>, Anthony Hunter
<https://orcid.org/0000-0001-5602-7446>

Abstract. Similarity is often reduced to a single score, sufficient for ranking but insufficient for explanation, as it does not reveal which aspects drive the comparison. We introduce *inverse simFOL*, an interpretable framework that infers clause and symbol-level importance weights explaining a target similarity score. Building on a fixed and transparent simFOL pipeline over first-order logic (FOL), the problem is formulated as a discrete, constrained optimization problem and solved via genetic search to recover diverse explanations. Experiments on STS-B pairs and an argument-level case study show that similarity admits multiple valid explanations and is strongly context-dependent.

Keywords. Explainable Similarity, Text-to-Logic, Argumentation

1. Introduction

Similarity is commonly treated as a single intrinsic quantity: given two sentences, formulae, or arguments, a similarity function outputs a score used for ranking, retrieval, or clustering. This view underlies benchmarks such as STS-B [1], where similarity is reduced to an averaged human judgment. However, this masks an important fact: similarity is not unique, as annotators disagree and averaging hides a distribution of plausible interpretations [2].

Beyond this variability, similarity is also inherently context-dependent. Recent work models it as a conditional notion, where the same pair can receive different scores depending on external context or perspective [3,4], while representation learning suggests that it emerges from interactions between fine-grained components rather than a single global

¹Corresponding Author: Nino Pireaud, nino.pireaud@inria.fr

comparison [5]. Together, these findings suggest that similarity should be understood not as a fixed relation, but as a contextual and multi-perspective judgment.

This issue is particularly salient in structured contexts where meaning depends on role and perspective. Consider the pair “I saw a black cat” and “I saw a brown cat”. Depending on the context—e.g., superstition versus allergy—the color difference may be either crucial or irrelevant, leading to different similarity judgments for the same pair. More generally, the relevance of information depends on its role within a structured setting, such as a claim, stance, or argumentative objective. These observations motivate a shift from the standard forward question: *What is the similarity between two objects?* to a more explanatory one: **Which interpretable profile explains a target similarity score?**

We address this via inverse simFOL, building on simFOL [6], a hierarchical and fully interpretable pipeline from text to logical comparison. Texts are translated into CNF FOL, and similarity is computed across symbols, literals, clauses, and formulae; at the argument level, support and claim are combined. In this work, we rely on a fixed instance of this pipeline: symbols use SBERT similarity [7], literals combine predicate and term similarity, clauses rely on fuzzy Tversky [8], and clause sets are aligned via best match. This forward model is transparent and satisfies desirable properties (see [6]).

In practice, even with a fixed logical representation, similarity remains underdetermined because it depends on which components are emphasized. This is especially relevant in argumentation, where contributions are often enthymematic [9] and must be interpreted relative to context. For instance, “I saw a black cat” and “I saw a brown cat” may be dissimilar in a superstition context but similar in an allergy-related discussion.

Thus, in argumentative settings, similarity is inherently context-dependent. It is not only a property of texts, but of their role within an argument. This is why integrating similarity into argumentation frameworks is essential: it enables more precise evaluation by accounting for context, guiding aggregation, and reducing redundancy between arguments [10].

Building on this insight, we keep the simFOL hierarchy fixed and treat clause and symbol weights as explanatory variables. Inverse simFOL then searches for weight profiles that best justify a given similarity score. This inversion separates *how similarity is computed* from *which components matter*, providing both interpretability and flexibility. Instead of a single score, the model produces structured explanations, and different target values may correspond to different valid profiles, reflecting alternative interpretations of the same pair.

The goal of this paper is therefore not to introduce another black-box predictor of similarity, but to show that similarity itself can be explained as a structured and context-dependent object, particularly in argumentative settings.

Contributions. (i) We evaluate the text-to-FOL step across three LLMs, showing its impact on similarity. (ii) We introduce inverse simFOL with an efficient genetic search to recover interpretable weight profiles at sentence and argument levels. (iii) We analyze the resulting solution space on three STS-B-based FOL datasets, providing hierarchical structure and interpretability insights, and illustrate the approach on argumentative cases.

2. A Fixed and Explainable simFOL Instance

This section specifies the fixed forward instance of simFOL used in this work [6]. Similarity is computed hierarchically over grounded CNF FOL representations (quantifier-free,

without variables), ensuring tractability [11]. Symbols are first compared and combined at the literal level. Clauses are then aligned via best-match using flat similarity, providing a stable structure independent of context. Given this alignment, weighted similarity emphasizes relevant predicates, terms, and clauses. Scores are finally aggregated at the formula level and, for arguments, combined over supports and claims. This separation between alignment (structure) and weighting (importance) is central: the comparison remains fixed, while interpretative variation is captured through the weights.

2.1. Logical forms

We consider a quantifier-free CNF first-order logic (CNF FOL) language $\mathcal{L}$, built from predicate symbols and constants. Formulae are constructed using the logical connectives $\neg$ and $\vee$. We write $\Phi \vdash \alpha$ for logical entailment, and a set of formulae Φ is inconsistent if $\Phi \vdash \perp$. In our setting, each sentence is translated into a finite CNF FOL representation, i.e., a set of clauses $\Phi = \{C_1, \dots, C_n\}$, where each clause is a disjunction of literals, and each literal is a predicate applied to an ordered tuple of constants. At the argument level, we adopt a notion of FOL argument defined within our restricted CNF FOL setting, where representations are not necessarily minimal.

Definition 1. A **CNF FOL argument** is a pair $A = \langle \Phi, \alpha \rangle$, $\Phi \subseteq \mathcal{L}$ is consistent ($\Phi \not\vdash \perp$) and entails $\alpha \in \mathcal{L}$ ($\Phi \vdash \alpha$). We use $S(A) = \Phi$ for the support, $C(A) = \{\alpha\}$ for the claim.

2.2. Predicate and constant similarity

The lowest level compares predicates and constants using `simP`, instantiated as cosine similarity between SBERT embeddings (rescaled to $[0, 1]$). This enables semantic comparison beyond syntax: negation is encoded at the predicate level (e.g., `Not.Odd`), allowing related expressions such as `Not.Odd` and `Even` to be matched, while preserving graded proximity (e.g., `Cat` closer to `Animal` than to `Car`, and `Black` similar to `Brown`).

2.3. Literal similarity

Literal similarity combines predicate similarity and parameter (constants) similarity by averaging both. Parameter similarity mixes an *ordered* comparison, capturing structural relations via position-wise matching (e.g., `Father(Bruce, Bob)` vs. `Father(Bob, Bruce)`), and an *unordered* best-match comparison, capturing global thematic similarity. The parameter λ controls this trade-off; after conducting preliminary tests, we fix $\lambda = 0.8$ in all experiments, favoring structural alignment while retaining some flexibility.

Definition 2. Let $\lambda \in (0, 1)$. For two literals $(P, \vec{a})$ and $(Q, \vec{b})$, the **flat literal similarity** is: $\text{simL}^{\text{flat}, \lambda}((P, \vec{a}), (Q, \vec{b})) = \frac{1}{2} \left(\text{simP}(P, Q) + \text{sim}_{\text{para}}^{\lambda}(\vec{a}, \vec{b}) \right)$,

where $\text{sim}_{\text{para}}^{\lambda}(\vec{a}, \vec{b}) = \lambda \text{sim}_{\text{ord}}(\vec{a}, \vec{b}) + (1 - \lambda) \text{sim}_{\text{unord}}(\vec{a}, \vec{b})$.

The ordered component is $\text{sim}_{\text{ord}}(\vec{a}, \vec{b}) = \frac{\sum_{i=1}^{\min(|\vec{a}|, |\vec{b}|)} \text{simP}(a_i, b_i)}{\min(|\vec{a}|, |\vec{b}|)}$,

and the unordered component is $\text{sim}_{\text{unord}}(\vec{a}, \vec{b}) = \frac{\sum_{a_i \in \vec{a}} \text{simP}(a_i, B_{a_i}^{\vec{b}}) + \sum_{b_j \in \vec{b}} \text{simP}(b_j, B_{b_j}^{\vec{a}})}{|\vec{a}| + |\vec{b}|}$,

where $B_x^{\vec{y}} = \arg \max_{y \in \vec{y}} \text{simP}(x, y)$.

The flat version captures structural similarity without reweighting and serves as a stable backbone for alignment. The weighted version extends it with importance weights on predicates and parameters, modulating their contribution so that more informative elements have greater impact. We use w_p for both, normalized to sum to 1, enabling an interpretable, attention-like aggregation.

Definition 3. Let $\lambda \in (0, 1)$. For two literals $(P, \vec{a})$ and $(Q, \vec{b})$, the **weighted literal similarity** is: $\text{simL}^{\text{weight}, \lambda}((P, \vec{a}), (Q, \vec{b})) =$

$$\frac{w_p(P)w_p(Q)\text{simP}(P, Q) + \text{sim}_{\text{para}}^{\lambda}(\vec{a}, \vec{b})(\lambda w_{\text{ord}} + (1 - \lambda)w_{\text{unord}})}{w_p(P)w_p(Q) + \lambda w_{\text{ord}} + (1 - \lambda)w_{\text{unord}}}, \text{ where}$$

$$w_{\text{ord}} = \sum_{i=1}^{\min(|\vec{a}|, |\vec{b}|)} w_p(a_i)w_p(b_i), \text{ and } w_{\text{unord}} = \sum_{a_i \in \vec{a}} w_p(a_i)w_p(B_{a_i}^{\vec{b}}) + \sum_{b_j \in \vec{b}} w_p(b_j)w_p(B_{b_j}^{\vec{a}})$$

2.4. Clause similarity

Clause similarity (flat and weighted) is defined using fuzzy Tversky [8], a generalization of classic overlap measures such as Dice [12].

The key idea is to compare clauses via their literals. A membership function measures how well a literal matches a clause by aggregating its similarity to all literals in the other clause; we use the max operator, matching each literal to its closest counterpart. Fuzzy Tversky then combines shared (A) and non-overlapping parts (B, C), with parameters α, β controlling the trade-off; $\alpha = \beta = \frac{1}{2}$ yields Dice, so that literals from both clauses are of equal importance, also ensuring symmetry.

Definition 4. Let $\mathbb{X}$ be a set of objects, let $x \in \mathbb{X}$, $Y \subseteq \mathbb{X}$, let $\oplus$ be an aggregation operator, and let $\text{sim}_{\mathbb{X}}$ be a similarity measure on $\mathbb{X}$. The **membership function** of x in Y is $\oplus_{\text{sim}_{\mathbb{X}}}^{\oplus}(x, Y) = \oplus_{y \in Y} \text{sim}_{\mathbb{X}}(x, y)$.

Definition 5. Let C_1, C_2 be two clauses, $\alpha, \beta \geq 0$, and let $\oplus_{\text{simL}}^{\oplus}$ be a membership function over literals. Then the **Fuzzy Tversky clause similarity** is:

$$\begin{aligned} \text{Tve}^{\alpha, \beta, \oplus_{\text{simL}}^{\oplus}}(C_1, C_2) &= \frac{A}{A + \alpha B + \beta C}, \text{ where} \\ A &= \frac{1}{2} (\sum_{x \in C_1} \oplus_{\text{simL}}^{\oplus}(x, C_2) + \sum_{y \in C_2} \oplus_{\text{simL}}^{\oplus}(y, C_1)), \\ B &= \sum_{x \in C_1} (1 - \oplus_{\text{simL}}^{\oplus}(x, C_2)), \quad C = \sum_{y \in C_2} (1 - \oplus_{\text{simL}}^{\oplus}(y, C_1)). \end{aligned}$$

In our experiments, flat and weighted clause similarity are defined as: $\text{simC}_{\text{dic}}^{\text{flat}}(C_1, C_2) = \text{Tve}^{\frac{1}{2}, \frac{1}{2}, \oplus_{\text{simL}}^{\text{max}}}(C_1, C_2)$ and $\text{simC}_{\text{dic}}^{\text{weight}}(C_1, C_2) = \text{Tve}^{\frac{1}{2}, \frac{1}{2}, \oplus_{\text{simL}}^{\text{weight}, 0.8}}(C_1, C_2)$.

2.5. Formulae similarity

At the level of clause sets, we use weighted average best-match. Clause weights w_c are normalized to sum to 1 within each clause set (i.e., per formulae). Best-match alignment is determined using flat clause similarity, while the final score uses weighted clause similarity, yielding a stable structure with interpretable variation.

Definition 6. Let Φ, Ψ be two sets of clauses, their **weighted best-match similarity** is:

$$\text{simS}^{\text{bm}}(\Phi, \Psi) = \frac{\sum_{(C_1, C_2) \in \text{bm}(\Phi, \Psi)} w_{\oplus}(C_1, C_2) \text{simC}(C_1, C_2)}{\sum_{(C_1, C_2) \in \text{bm}(\Phi, \Psi)} w_{\oplus}(C_1, C_2)}, \text{ where}$$

bm uses flat clause similarity $\text{simC}^{\text{flat}}$ to define best-matching clause pairs, i.e., $\text{bm}(\Phi, \Psi) = \left\{ (C_1, C_2) \mid (C_1 \in \Phi, C_2 = \arg\max_{C' \in \Psi} \text{simC}^{\text{flat}}(C_1, C')) \right\}$
 $\left\{ \text{or } (C_1 \in \Psi, C_2 = \arg\max_{C' \in \Phi} \text{simC}^{\text{flat}}(C_1, C')) \right\}$

and where $w_{\oplus}$ denotes the *contribution weight* of a clause pair, computed using an aggregation operator $\oplus$: $w_{\oplus}(C_1, C_2) = \oplus(w_c(C_1), w_c(C_2))$. In this paper, $w_{\oplus}$ is instantiated as the average operator, i.e., w_{avg} .

2.6. Argument similarity

A similarity model $\mathbf{M} = \langle \text{simP}, \text{simL}, \text{simC}, \text{simS} \rangle$ defines the composition of the four hierarchical similarity levels. For arguments, similarity is then computed by combining support and claim similarities, with a parameter η controlling their relative importance.

Definition 7. Let $\eta \in (0, 1)$ and let $\mathbf{M} = \langle \text{simP}, \text{simL}, \text{simC}, \text{simS} \rangle$ be a similarity model. For arguments A, B , their **argument similarity** is defined by:

$$\text{simArg}_{\eta}^{\mathbf{M}}(A, B) = \eta \text{simS}(\text{S}(A), \text{S}(B)) + (1 - \eta) \text{simS}(\text{C}(A), \text{C}(B)).$$

3. Inverse simFOL

Inverse simFOL reformulates similarity as an explanatory problem. Instead of computing a score from fixed weights, we are given a pair (X, Y) and a target similarity s^* , and we search for interpretable weight profiles that make this score plausible.

3.1. Inverse formulation

We formalize the inverse problem with two assumptions: (i) a discrete weight space for interpretability and efficiency, and (ii) a tolerance on the target score, since similarity values arise from coarse judgments, making exact matching unnecessarily rigid.

Definition 8. Let two set of formulae $X, Y \subseteq \mathcal{L}$ and $\delta \in (0, 1]$. A **discrete weight profile** $w \in \mathcal{W}^{\delta}$ consists of normalized weight vectors (blocks) associated with X and Y , where each block takes values in $\{0, \delta, 2\delta, \dots, 1\}$ and sums to 1. Blocks correspond to clause weights and symbol weights (predicates and constants). When X and Y are arguments, w additionally includes $\eta \in (0, 1)$ controlling the support–claim balance.

Example 1. Consider the sentences X : “I saw a black cat” and Y : “I saw a brown cat”. A possible CNF FOL representation is:

$$X = \{ \text{See}(I, \text{Cat}), \text{Black}(\text{Cat}) \}, \quad Y = \{ \text{See}(I, \text{Cat}), \text{Brown}(\text{Cat}) \}.$$

We use $\delta = 0.05$ as in our experiments. A valid discrete weight profile $w \in \mathcal{W}^{\delta}$ is:

- clause weights for X : $w_c^X = (\text{See}(I, \text{Cat}) : 0.7, \text{Black}(\text{Cat}) : 0.3)$,
- clause weights for Y : $w_c^Y = (\text{See}(I, \text{Cat}) : 0.7, \text{Brown}(\text{Cat}) : 0.3)$,
- symbol weights for X : $w_p^X = (\text{See} : 0.2, \text{Black} : 0.4, I : 0.1, \text{Cat} : 0.3)$,
- symbol weights for Y : $w_p^Y = (\text{See} : 0.2, \text{Brown} : 0.5, I : 0.1, \text{Cat} : 0.2)$.

All weights belong to $\{0, 0.05, 0.10, \dots, 1\}$ and each block sums to 1, hence $w \in \mathcal{W}^{\delta}$.

Definition 9. Let two set of formulae $X, Y \subseteq \mathcal{L}$, $s^* \in [0, 1]$ a target similarity score, δ a discretization step, and $\varepsilon \geq 0$ a tolerance threshold. Let $\text{Sim}_w(X, Y)$ denote the forward **simFOL** similarity under a profile w . The **inverse simFOL problem** is to find weight profiles satisfying: $\text{InvSim}(X, Y, s^*, \varepsilon) = \{w \in \mathcal{W}^\delta \mid |\text{Sim}_w(X, Y) - s^*| \leq \varepsilon\}$.

The objective is to find structured weight redistributions that match the target score within tolerance; δ controls granularity and ε precision.

3.2. Genetic search for inverse simFOL

We formulate inverse simFOL as a discrete optimization problem over structured weight profiles. Each candidate solution is a vector w composed of *normalized blocks*, i.e., sub-vectors whose entries sum to 1. For sentence pairs (X, Y) , w contains four blocks: clause weights and symbol weights for each sentence. For argument pairs, the same structure applies to supports and claims, and is complemented by a scalar $\eta \in (0, 1)$ controlling their relative contribution (Definition 7).

Objective. Given a target similarity s^* , we define the score function $\text{Sim}_w(X, Y)$ as:

$$\begin{aligned} \text{Sim}_w(X, Y) &= \text{simS}_w^{\text{bm}}(X, Y) && \text{for sentences, and} \\ \text{Sim}_w(X, Y) &= \eta \text{simS}_w^{\text{bm}}(S(X), S(Y)) + (1 - \eta) \text{simS}_w^{\text{bm}}(C(X), C(Y)) && \text{for arguments.} \end{aligned}$$

The objective is to minimize the **absolute error**: $\text{Err}(w) = |\text{Sim}_w(X, Y) - s^*|$.

This problem is non-convex and typically admits multiple solutions, each corresponding to a distinct explanatory perspective.

Overview. We solve this problem using a genetic algorithm (Algorithm 1). Each individual in the population encodes a discrete weight profile w . The algorithm iteratively minimizes $\text{Err}(w)$ while maintaining a diverse set of solutions.

Initialization (Algorithm 2). The **initial population** $\mathcal{P}$ is generated using structured sampling rather than purely random weights. For each block, we sample a *top-3 ranking*, i.e., an ordering of the three most important elements, while the remaining elements share equal importance. We then assign decreasing weights according to this ranking and normalize the block. For argument pairs, η is sampled from a discretized uniform grid over $[0, 1]$. This initialization strategy reduces the search space while preserving meaningful explanatory patterns.

Evaluation and solution extraction. At each iteration, all individuals $w \in \mathcal{P}$ are evaluated using $\text{Err}(w)$. Individuals such that $\text{Err}(w) \leq \varepsilon$ are added to a solution set $\mathcal{S}$ and immediately replaced in the population by newly initialized individuals, preventing the search from collapsing onto already discovered explanations.

Selection. Parent selection is performed using *tournament selection*. A tournament size $k = \lfloor \frac{1}{\text{TR}} \rfloor$ is derived from the **turnover rate** TR. For each parent, k individuals are sampled uniformly at random and the one with lowest Err is selected. This balances selection pressure and diversity.

Crossover and offspring generation. A new set of **offspring** $\mathcal{C}$ is generated until reaching size $\lfloor N \cdot \text{TR} \rfloor$. For each child: (i) with probability P_c , two parents p_1, p_2 are selected and combined using $\text{CROSSOVER}(p_1, p_2)$; (ii) otherwise, a parent is copied.

Crossover operates at the block level: each block is independently inherited from either parent with probability 0.5, preserving coherent substructures and normalization.

Algorithm 1. Genetic inverse simFOL

Require: pair (X, Y) , target s^* , population size N , iterations T , tolerance ε , crossover probability P_c , mutation probability P_m , turnover rate TR

Ensure: solution set $\mathcal{S}$

```

1:  $\mathcal{P} \leftarrow \text{INITPOPULATION}(X, Y, N)$ 
2:  $\mathcal{S} \leftarrow \emptyset$ 
3: for  $t = 1$  to  $T$  do
4:   Compute  $\text{Err}(w)$  for all  $w \in \mathcal{P}$ 
5:   Add solutions with  $\text{Err}(w) \leq \varepsilon$  to  $\mathcal{S}$ 
6:   Replace  $n$  individuals with  $\text{Err}(w) \leq \varepsilon$ 
   by  $\text{INITPOPULATION}(X, Y, n)$ 
7:    $k \leftarrow \lfloor 1/TR \rfloor$ 
8:    $\mathcal{P}_{\text{parents}} \leftarrow \text{TournamentSelect}(\mathcal{P}, k)$ 
9:    $\mathcal{C} \leftarrow \emptyset$ 
10:  while  $|\mathcal{C}| < \lfloor N \cdot TR \rfloor$  do
11:    if  $\text{rand} < P_c$  then
12:      Sample  $p_1, p_2 \in \mathcal{P}_{\text{parents}}$ 
13:       $c \leftarrow \text{CROSSOVER}(p_1, p_2)$ 
14:    else
15:       $c \leftarrow$  copy of random parent
16:    if  $\text{rand} < P_m$  then
17:       $c \leftarrow \text{MUTATE}(c)$ 
18:     $\mathcal{C} \leftarrow \mathcal{C} \cup \{c\}$ 
19:   $\mathcal{P} \leftarrow$  best  $N$  individuals from  $\mathcal{P} \cup \mathcal{C}$ 
20: if  $\mathcal{S} \neq \emptyset$  then
21:   return  $\mathcal{S}$ 
22: else
23:   return  $\arg\min_{w \in \mathcal{P}} \text{Err}(w)$ 

```

Algorithm 2. InitPopulation (Top-3 ranking)

Require: objects (X, Y) , population size N

Ensure: population $\mathcal{P}$

```

1: Identify blocks  $\mathcal{B}$  induced by  $(X, Y)$ 
2:  $\mathcal{P} \leftarrow \emptyset$ 
3: while  $|\mathcal{P}| < N$  do
4:    $w \leftarrow \emptyset$ 
5:   for each block  $B \in \mathcal{B}$  do
6:     Sample a top-3 ranking  $\pi_B$ 
7:     Assign decreasing weights (step  $\delta$ )
8:     Distribute remaining mass uniformly
9:     Normalize to obtain  $w_B$ 
10:    Append  $w_B$  to  $w$ 
11:   if argument mode then
12:     Sample discretized  $\eta \in (0, 1)$ 
13:     Append  $\eta$  to  $w$ 
14:    $\mathcal{P} \leftarrow \mathcal{P} \cup \{w\}$ 
15: return  $\mathcal{P}$ 

```

Figure 1. Genetic search and ranking-based initialization for inverse simFOL.

Mutation. With probability P_m , a child undergoes mutation. Mutation operates independently on each block by transferring a fixed mass δ from one component (with non-zero weight) to another (with non-maximal weight), preserving normalization. For argument pairs, η is mutated independently by adding or subtracting δ and clipping to $[0, 1]$.

Replacement. The new population selects the best N individuals from $\mathcal{P} \cup \mathcal{C}$ (elitist replacement), ensuring non-decreasing solution quality.

Hyperparameters. The algorithm uses: a population size $N = 100$, a number of iterations $T = 100$, a crossover probability $P_c = 0.7$, a mutation probability $P_m = 0.3$, a turnover rate $TR = 0.25$, a mutation step $\delta = 0.05$, and a tolerance $\varepsilon = 0.05$. These values balance exploration and exploitation, with emphasis on recovering diverse solutions. These parameters were selected through preliminary iterative experiments to balance solution quantity and diversity. Mutation and the population size help maintain diversity during the search, while tournament selection and crossover exploit promising solutions.

3.3. Exploration and classification of explanatory solutions

The inverse problem often yields many valid solutions, due to small weight variations and near-equivalent similarity combinations, making the space redundant and hard to interpret. We address this by structuring it hierarchically, grouping solutions into interpretable families based on their dominant source of importance and their set of salient elements.

Dominance filtering. We first classify solutions depending on whether similarity is mainly driven by clauses or symbols. For each block B , we define *important elements* as those whose weight exceeds a threshold based on the average weight ($1/|B|$) scaled by a factor (1.5 in our studies). We then compare the sum of important clauses and symbols:

- **Clause-dominant:** more important clauses than symbols,
- **Symbol-dominant:** more important symbols than clauses,
- **Neutral:** balanced importance.

Hierarchical clustering. Within each group, solutions are clustered according to the identity of their important elements. For clause-dominant solutions, clustering is first performed over clauses (lvl-1), then refined over symbols (lvl-2); for symbol-dominant solutions, the order is reversed (lvl-1 on symbols and lvl-2 on clauses). In the neutral case, clustering starts with clauses (lvl-1), which offer stronger discrimination due to their smaller number, and is then refined with symbols (lvl-2). This two-level hierarchy progressively partitions the solution space into explanatory families.

Coherence filtering. Although the hierarchy reduces the explanatory space, some cases still yield many clusters. We therefore introduce a *coherence constraint*: a cluster with high-importance symbols is kept only if at least one clause containing these symbols is also classified as important. This aligns symbol- and clause-level importance, yielding fewer, more interpretable clusters, while favoring structured explanations over more distributed or compensatory patterns.

The three preceding tools reduce and highlight interpretable explanatory patterns by separating dominant similarity sources and aligning symbol- and clause-level evidence. However, it is a methodological choice that can be adapted depending on the criteria used to categorize solutions.

Argument-level extension. For arguments, we may further refine the analysis using the parameter η , which controls the balance between support and claim. Solutions are thus additionally classified as *support-dominant*, *claim-dominant*, or *neutral*, providing a complementary dimension of interpretation.

4. Experimental Study

The experiments assess whether inverse simFOL produces meaningful and compact explanations, and evaluate the effectiveness of the proposed search procedure.

4.1. Logical data

We consider sentence pairs translated into quantifier-free CNF FOL using three local LLMs (Gemma3:27B, Qwen2.5-32B-Instruct, Llama3.1:70B), yielding three logical

Model	Quantity	Negation	Lexical	Semantic	Logic	Avg
GEMMA-FS	0.58	1.00	0.60	0.87	0.50	0.71
QWEN-ZS	0.75	1.00	0.33	0.53	0.80	0.68
LLAMA-ZS	0.75	1.00	0.50	0.60	1.00	0.77

Table 1. Human evaluation of LLM-generated outputs based on 5 rules; bold values are the best scores.

variants of the same dataset. These representations are not evaluated as standalone text-to-logic systems, but used to assess the robustness of inverse simFOL explanations across different logical realizations. The exact guidelines, code, and datasets are available at: <https://github.com/N1N0-P/Inverse-simFOL.git>.

Evaluation protocol. We evaluate CNF FOL quality using a lightweight rule-based annotation scheme with four labels: *no error* (fully satisfied rule), *minor error* (small deviations not affecting meaning), *major error* (meaning-altering violation), and *not applicable*. The rules cover five aspects of translation quality:

- **Quantities:** correct encoding of exact or vague amounts (e.g. “three cats” $\rightarrow$ `Quantity(3, cat)`, “most cats” $\rightarrow$ `Most(cat)`).
- **Negation:** faithful representation of negation, including lexical variants (e.g. “not black” $\rightarrow$ `Not.Black(cat)`, “unhappy” $\rightarrow$ `Unhappy(cat)`).
- **Lexical alignment:** predicates and constants are grounded in the input text (e.g. “a man runs” $\rightarrow$ `Man, Run`).
- **Semantic alignment:** the meaning is preserved under paraphrases or lexical variation (e.g. “a boy eats” $\sim$ “a child consumes food”).
- **Logical connectives:** correct encoding of logical structure, especially conditionals (e.g. “if A then B” $\rightarrow$ $\neg A \vee B$).

Prompting setting. We conducted preliminary experiments comparing zero-shot and few-shot prompting, and selected the setting that provided the most accurate logical representations. In the following, QWEN-ZS and LLAMA-ZS denote zero-shot prompting, while GEMMA-FS refers to a few-shot setting where the model is guided by the evaluation guidelines and examples of text-to-FOL translations. All reported results use this fixed configuration with temperature set to 0 to maximize reproducibility.

Human annotation. We randomly sampled 15 sentence pairs from STS-B and generated FOL translations with the three LLMs, resulting in 45 examples. To ensure coverage, each rule was applicable in at least four cases. The dataset was annotated by 4 experts in logic, followed by a reconciliation phase to produce a ground truth. For quantitative evaluation, we map labels to scores as follows: *no error* = 1, *minor error* = 0.5, and *major error* = 0. *Not applicable* cases are excluded from the average.

Table 1 reports the average scores over the 45 examples, computed from ground-truth labels obtained after expert reconciliation. All models achieve perfect performance on negation. Encoding of quantities and logical connectives is generally strong for QWEN-ZS and LLAMA-ZS, while GEMMA-FS performs slightly lower on these aspects. In contrast, lexical and semantic alignment are more challenging for QWEN-ZS and LLAMA-ZS, especially for QWEN-ZS, whereas GEMMA-FS remains more consistent.

Rule applicability. The impact of each rule depends on both semantic importance and frequency. Missing a negation can strongly alter meaning, whereas missing a quantity may be negligible. Lexical and semantic alignment apply to all predicates and terms (tens of thousands of occurrences) and thus dominate similarity through accumulation, while quantities and negation occur in only 1–3% of cases and logical connectives in less than 1%. Overall, frequent rules drive global similarity, whereas rarer but critical ones, such as negation, have strong local effects.

Translation errors. The logical datasets are generated in a single pass (one-shot), without post-hoc correction, and some outputs do not comply with the CNF FOL format required by simFOL. We distinguish three error types: (i) *invalid clauses* (e.g. empty clauses or malformed syntax), (ii) *higher-order constructs*, and (iii) *CNF violations*, which may co-occur within the same instance. LLAMA-ZS produces 2430 invalid clauses, 552 higher-order expressions, and 53 CNF violations; GEMMA-FS 1560, 752, and 1; and QWEN-ZS 1780, 492, and 6. From 8626 sentence pairs, fully valid rates are 86% (LLAMA-ZS), 88% (GEMMA-FS), and 86% (QWEN-ZS). The intersection of valid instances yields 6288 pairs (73%), used for consistent comparison across LLMs.

4.2. Sentence-level explanatory similarity

We analyze inverse simFOL at the sentence level using genetic search on the intersection of the three logical datasets. For each sentence pair and target similarity, the algorithm returns a set of weight profiles within a tolerance, or the closest solutions otherwise. A non-negligible proportion of instances are unsolved under this tolerance (3.3% for GEMMA-FS, 9.3% for LLAMA-ZS, and 16.5% for QWEN-ZS). These failures indicate that the target similarity cannot be reached given the initialization, iteration budget, and the quality of the underlying logical representations. They correlate with lexical and semantic alignment quality (Table 1), suggesting that text-to-logic errors directly limit explainability. This behavior is expected, as inverse simFOL is designed to explain similarities between logical representations and is therefore sensitive to the quality of the generated FOL.

We first analyse the solution space, which is large and confirms that similarity admits multiple valid explanations. Using the hierarchical classification of Section 3.3, Table 2 reports the average number of solutions and clusters per dominance family, with and without the coherence constraint, as well as for pairs without coherent solutions. Without coherence, symbol-dominant solutions strongly dominate (601.40 on average), with high cluster multiplicity (135.91 at level 1 and 99.88 at level 2), reflecting substantial combinatorial variability. Clause-dominant solutions are also numerous (63.51), confirming the overall complexity of the space. Applying coherence markedly reduces this variability, decreasing both solution counts (symbol: 24.16; clause: 28.82) and cluster numbers across these families. In contrast, neutral solutions remain largely unaffected, due to their construction within a narrow range of importance profiles, which makes incoherent variants and cluster diversity unlikely. Overall, coherence compresses the solution space into fewer, more stable families, improving interpretability and tractability by reducing cluster fragmentation.

4.3. Argument-level contextual case study

The sentence-level experiments (section 4.2) showed that inverse simFOL recovers interpretable explanatory profiles for a target similarity score. We now investigate whether

Family	Coherence	#solutions	lvl-1 clusters	lvl-2 clusters
Clause	without	338.19 / 265.95 / 231.76	24.19 / 18.27 / 15.85	75.14 / 53.99 / 46.31
	with	239.35 / 191.40 / 174.49	20.98 / 15.13 / 13.43	37.43 / 27.25 / 24.57
	no-coherent solutions	2.23 / 2.19 / 1.16	2.77 / 2.40 / 2.23	4.72 / 4.36 / 3.64
Symbol	without	397.46 / 360.33 / 382.43	57.23 / 44.61 / 38.56	108.25 / 87.58 / 76.24
	with	41.70 / 38.26 / 36.98	7.15 / 6.88 / 6.68	9.69 / 9.27 / 9.00
	no-coherent solutions	102.92 / 90.07 / 94.04	22.55 / 15.20 / 10.87	39.21 / 26.46 / 18.75
Neutral	without	54.93 / 32.02 / 35.29	1.15 / 1.20 / 1.20	1.18 / 1.25 / 1.23
	with	54.87 / 31.93 / 35.18	1.05 / 1.08 / 1.10	1.05 / 1.10 / 1.10
	no-coherent solutions	0.01 / 0.01 / 0.01	1.50 / 1.00 / 1.00	1.50 / 1.00 / 1.40

Table 2. Average metrics per dominance family across three LLM datasets (Gemma-FS / Llama-ZS / Qwen-ZS), including without coherence filtering, with coherence filtering, and without coherent solutions.

these profiles adapt to the argumentative context. Consider the following CNF FOL arguments:

$$A_1 = \langle \{ \text{See}(I, \text{Cat}), \text{Black}(\text{Cat}) \}, \{ \text{BadLuck}(I) \} \rangle,$$

$$A_2 = \langle \{ \text{See}(I, \text{Cat}), \text{Brown}(\text{Cat}) \}, \{ \text{BadLuck}(I) \} \rangle,$$

$$A_3 = \langle \{ \text{See}(I, \text{Cat}), \text{Black}(\text{Cat}) \}, \{ \text{Allergic}(I), \text{Not_Allergic}(I) \vee \text{Sneeze}(I) \} \rangle,$$

$$A_4 = \langle \{ \text{See}(I, \text{Cat}), \text{Brown}(\text{Cat}) \}, \{ \text{Allergic}(I), \text{Not_Allergic}(I) \vee \text{Sneeze}(I) \} \rangle.$$

Both pairs differ by exactly the same syntactic modification in their support: $\text{Black}(\text{Cat})$ is replaced by $\text{Brown}(\text{Cat})$. The only difference is the claim. In the first pair, the claim concerns superstition, where the cat’s color is naturally expected to play a central role. In the second pair, the claim concerns an allergy, where the cat’s color should have little influence on the perceived similarity. Consequently, we consider two target similarity regimes, with illustrative values $\text{sim}(A_1, A_2) = 0.4$ and $\text{sim}(A_3, A_4) = 0.9$.

For each argument pair, inverse simFOL generates a population of explanatory solutions. Each solution assigns importance weights to clauses and symbols, producing an explanatory profile of the target similarity. To compare these profiles across contexts, we compute, for every logical label, the proportion of solutions in which the label receives a high importance score. Fisher’s exact test with Benjamini–Hochberg correction [13] is then applied to compare these activation frequencies between the two populations of solutions. The resulting profiles reveal two distinct explanatory strategies. For the superstition pair (A_1, A_2) , inverse simFOL consistently concentrates importance on the color difference. The clause $\text{Black}(\text{Cat})$ is selected in all solutions, whereas it appears in only 9.3% of the solutions for the allergy pair (A_3, A_4) ($p_{\text{FDR}} = 5.9 \times 10^{-11}$). Likewise, the predicate labels Black and Brown are significantly more frequently activated in the superstition context ($p_{\text{FDR}} = 4.5 \times 10^{-2}$ and 5.9×10^{-3} , respectively). These results indicate that the lower similarity of (A_1, A_2) is primarily explained through the semantic distinction between black and brown cats.

Conversely, for the allergy pair (A_3, A_4) , color-related labels are no longer systematically selected. Instead, explanatory importance is redistributed toward information shared by both arguments. This is reflected by the absence of significant differences for structural labels such as $\text{See}(I, \text{Cat})$, See , and Cat , whose activation frequencies remain comparable across both contexts after multiple-testing correction.

Overall, although the logical perturbation between the two argument pairs is identical, inverse simFOL produces substantially different explanatory profiles. In the superstition context, the semantic distinction between black and brown cats becomes the dominant explanation for the similarity score. In the allergy context, this distinction becomes largely incidental, and the explanation shifts toward the logical information shared by

both arguments. These results demonstrate that inverse simFOL does not simply explain similarity scores: it adapts its explanatory profiles to the semantic context in which similarity is evaluated.

5. Conclusion

We introduced inverse simFOL, an interpretable framework that explains similarity through importance profiles over logical representations. Given a target similarity score, inverse simFOL recovers multiple valid weight configurations, each highlighting the logical clauses and symbols that best explain the observed similarity. Our experiments show that similarity admits multiple coherent explanations and that these explanations depend on the semantic context. By organizing the resulting solution space through dominance patterns and coherence filtering, inverse simFOL extracts a compact set of interpretable explanatory profiles rather than a single opaque similarity score. More generally, this work shifts the focus from measuring similarity to explaining similarity, providing a richer understanding of which logical components drive agreement and disagreement between structured representations. Future work includes improving text-to-logic translation and learning explanatory weight profiles directly from annotated data.

Acknowledgements

This work was supported by the French National Research Agency (ANR) grant AG-GREEY ANR-22-CE23-0005, and by the French government, through the 3IA Cote d’Azur Investments in the project managed by ANR-23-IACL-0001.

References

- [1] Cer D, Diab M, Agirre E, Lopez-Gazpio I, Specia L. Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. *arXiv preprint arXiv:170800055*. 2017.
- [2] Benito-Santos A, Ghajari A. Beyond Averages: Learning with Annotator Disagreement in STS. In: *Proceedings of EMNLP*; 2025. p. 35539-45.
- [3] Deshpande A, Jimenez C, Chen H, Murahari V, Graf V, Rajpurohit T, et al. C-STs: Conditional semantic textual similarity. In: *Proceedings of EMNLP*; 2023. p. 5669-90.
- [4] Zhang G, Zhou Y, Bollegala D. Case-condition-aware sentence embeddings for conditional semantic textual similarity measurement. In: *Proceedings of EACL*; 2026. p. 4954-68.
- [5] Chen S, Zhang H, Chen T, Zhou B, Yu W, Yu D, et al. Sub-sentence encoder: Contrastive learning of propositional semantic representations. In: *Proceedings of NAACL*; 2024. p. 1596-609.
- [6] David V, Delobelle J, Mailly JG. Technical Report – A Context-Sensitive Multi-Level Similarity Framework for First-Order Logic Arguments: An Axiomatic Study; 2026.
- [7] Reimers N. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *arXiv preprint arXiv:190810084*. 2019.
- [8] Coletti G, Bouchon-Meunier B. Fuzzy similarity measures and measurement theory. In: *FUZZ-IEEE*. IEEE; 2019. p. 1-7.
- [9] Walton D. Enthymemes, common knowledge and plausible inference. *Philosophy and Rhetoric*. 2001;34(2):93-112.
- [10] Amgoud L, David V. A general setting for gradual semantics dealing with similarity. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. vol. 35; 2021. p. 6185-92.
- [11] David V, Delobelle J, Mailly JG. Similarity Measures for First-Order Logical Arguments. In: *NMR 2025-23rd International Workshop on Nonmonotonic Reasoning*; 2025. .
- [12] Dice L. Measures of the amount of ecologic association between species. *Ecology*. 1945;26(3):297-302.
- [13] Benjamini Y, Hochberg Y. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal statistical society: series B (Methodological)*. 1995;57(1):289-300.