Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Retrieval-Assisted Scalar Grounding for Natural-Language Optimization: A Diagnostic Study

Soroush Vahidi

sv96@njit.edu

New Jersey Institute of Technology

Research Article

Keywords:

Posted Date: November 2nd, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11558034/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Retrieval-Assisted Scalar Grounding for Natural-Language Optimization: A Diagnostic Study

Soroush Vahidi [1*]

[1*]Department of Computer Science, Ying Wu College of Computing, New Jersey Institute of Technology, University Heights, Newark, NJ 07102-1982, USA.

Corresponding author(s). E-mail(s): sv96@njit.edu;

Abstract

Retrieval and structural checks can make natural-language optimization outputs appear usable even when numerical grounding—assigning textual quantities to model parameters—is incomplete. We examine this diagnostic gap through three questions: fixed-catalog schema identification, the validity of coverage/type readiness as a numerical check, and the mechanisms underlying grounding outcomes. On 331 queries from NLP4LP, a benchmark for natural-language linear and mixed-integer linear programming, and 335 schema documents, term frequency–inverse document frequency retrieval and a pretrained dense encoder identify the target schema for 301 and 313 queries. Lexical grounding with multiplicative-word extraction passes coverage/type readiness on 265 queries, but only seven meet the complete stored-reference scalar-tolerance criterion: 2.64% of ready outputs. A mutually exclusive residual partition distinguishes retrieval, coverage, type, and numerical failures. Six grounding procedures under fixed retrieval yield between seven and 46 complete successes, with candidates, scoring, abstention, and decoding changing together. A separate 49-query assignment study with fixed candidates and scores improves exact slot counts without increasing the five whole-instance tolerance successes, limiting attribution to search. The methodological contribution is joint reporting of retrieval, readiness, and complete scalar agreement with explicit assistance, denominators, and evidence boundaries. Reference-derived inventories and benchmark exposure restrict the findings to assisted grounding. Supplementary relationship judgments and author contrasts assisted by artificial intelligence illustrate semantic limits without establishing accuracy. A versioned public release supports saved-record reanalysis; missing historical predictions and restricted originals prevent full experimental reproduction.

Keywords: natural language processing, optimization modeling, knowledge representation, information retrieval, semantic grounding, intelligent information systems

Acknowledgements. The author thanks his mother for emotional support, Professor Ioannis Koutis for academic guidance, NJIT for Wulver computing access, and Anders Borum for complimentary Secure ShellFish access. Cohere Labs Catalyst, Google Cloud Research Credits and io.net supported the author's broader research; their use for this study's experiments is not established. Study-specific educational credits and in-kind support are distinguished in Statements and Declarations.

1 Introduction

Natural-language optimization interfaces must connect a problem's quantities to the appropriate parameters and mathematical roles. Retrieving a suitable template and filling its slots provide complementary parts of this structure, but neither operation alone establishes that the numerical mapping expresses the intended problem. A capacity and a cost may share a value, and reversing a bound can preserve both the number and its type.

Entity extraction, structured representations, and formulation generation address different parts of this translation [2, 13, 20, 21]. Database interfaces distinguish schema elements from their links to textual values [9, 26], and Text2Opt-Bench explicitly separates modeling from data binding [10]. Building on these established concerns, we ask which intermediate measurements reveal numerical binding failures and which favorable measurements can coexist with incomplete numerical results.

Retrieval-assisted scalar grounding provides an inspectable setting for this diagnosis. A retriever selects a document from a fixed catalog, and an existing deterministic procedure assigns numerical mentions (occurrences of quantities in the text) to scalar slots (parameters requiring single numerical values), producing a partial parameter mapping. The benchmark supplies reference-derived inventory information and includes all target schemas (Section 3.2). This setting supports analysis of intermediate checks under supplied structure, rather than evaluation of autonomous model formulation.

Three research questions organize the paper:

1. **RQ1—Schema retrieval:** How well does fixed-catalog retrieval identify the benchmark's target schema?
2. **RQ2—Diagnostic validity:** How often do structural coverage and type-readiness checks conceal failures of complete numerical grounding?
3. **RQ3—Grounding mechanisms:** What do matched grounding comparisons and frozen-score assignment controls reveal about candidate availability, scoring, abstention, and search?

The 331-query analysis of NLP4LP, a natural-language linear and mixed-integer linear programming benchmark, jointly reports schema hits (correct top-ranked templates), coarse readiness (coverage/type checks), and agreement across every required stored-reference scalar, then partitions residual failures. It includes a comparison of six complete grounding procedures on that frame and a separate 49-query assignment control that fixes candidate pools and scores within decoder comparisons. Neither population is an independent confirmatory sample.

The contributions are correspondingly focused:

1. A diagnostic protocol for reference-assisted grounding with a supplied parameter inventory that jointly reports retrieval, coverage/type readiness, and complete stored-reference scalar agreement with explicit input assistance and denominators.
2. A saved-record analysis of the 331-query benchmark that quantifies the readiness–grounding discrepancy through a mutually exclusive residual partition while preserving the conditional populations of Exact20, the mean within-tolerance slot fraction on comparable schema hits.
3. A distinction between whole-procedure comparisons and frozen-score assignment controls that clarifies what retained candidate, scoring, and search records can explain, supported by a versioned reanalysis package with explicit reconstruction limits.

The contribution lies in applying these diagnostics to supplied-slot grounding, not in priority for component-level evaluation or a new retrieval or matching algorithm. Semantic fidelity requires further validation: supplementary relationship judgments and contrasts prepared by the author with artificial intelligence (AI) assistance illustrate this boundary without providing independent semantic gold or provider-performance claims. Online Resource 1 preserves the other formulation, extraction, and historical auxiliary results under their own protocols.

Section 2 places this diagnostic approach in the context of grounding and formulation research. Sections 3 and 4 define the endpoints and comparisons; Section 5 addresses RQ1–RQ3, and Section 6 interprets the findings and limitations.

2 Related Work

2.1 Structured Grounding and Numerical Binding

StruG learns column grounding, value grounding, and column–value mapping for text-to-SQL (Structured Query Language) [9]; RAT-SQL encodes schema relations and linking [26]. Quantity interpretation similarly associates values with units and concepts [4, 24]. These studies distinguish element identification from linking to textual evidence. Their learned representations do not implement the supplied-slot protocol evaluated here.

Text2Opt-Bench studies solver-verified optimization problems with increasing numerical data, even when formulation structure remains simple [10]. Its BIND interventions, which externalize numerical data for programmatic binding, distinguish oracle-assisted data externalization from trained binding specialists; the study also documents failures of an untrained two-stage pipeline. We do not reproduce these interventions. Our narrower question concerns complete stored-reference scalar agreement among outputs that pass coverage/type checks and the explanations supported when assignment inputs and scores are fixed. The scaling and intervention results are therefore not directly comparable to these exposed-catalog conditional diagnostics.

2.2 Formulation Systems and Evaluation Endpoints

Optimization systems also differ in their outputs and evaluation endpoints. Natural Language for Optimization (NL4Opt) and Ner4Opt, a named-entity recognition framework for optimization, examine extraction and representations [13, 21]. OptiMUS and PaMOP generate formulations or solver code with feedback [2, 17], whereas the OptMATH data-synthesis framework and the ORLM and DeepOR modeling systems develop data or trained models [12, 16, 27]. Retrieval for code generation, as in OPT2CODE, differs from selecting the parameter inventory itself [3]. Online Resource 1, Table S1 and Section S11, provide the broader system context. End-to-end outcomes from these systems cannot substitute for scalar-binding measurements.

A related diagnostic perspective appears in Refai and Ahmed's component-level evaluation of generated formulations, including decision-variable and constraint precision/recall and constraint/objective root mean squared error [22]. OPT-Engine varies optimization-problem complexity to examine formulation and solving limitations, including constraint formulation [7]. Within our reference-assisted scalar inventory, we condition complete stored-reference agreement on readiness and use ordered gate accounting and fixed-score decoder comparisons to delimit failures and their attribution.

The role of search also depends on the task. OptMATH synthesizes modeling data through back-translation and forward checking [16]; our external extraction check evaluates neither that framework nor full-pipeline transfer. Autoformulation combines large language model (LLM)-guided Monte Carlo tree search with symbolic pruning over formulations [5], including modeling decisions outside our supplied-slot assignment controls. These distinctions describe task scope, not empirical comparisons with either system.

Structural validity and semantic validation require different checks. Constrained semantic-role inference and the incremental parsing method PICARD impose structural restrictions [19, 25], whereas EquivaMap checks mappings between formulations that preserve feasibility and optimality [28]. Our stored-scalar test establishes neither formulation equivalence nor fidelity to the source description; these stronger endpoints delimit what a positive diagnostic result means.

2.3 Methodological Position

For reference-assisted supplied slots, we combine complete stored-reference agreement, ordered gate accounting, explicit conditional populations, and controlled attribution from saved records. Classical maximum-sum assignment [14], existing retrievers, and existing grounders serve as diagnostic instruments. The six-method comparison holds retrieval and queries fixed while changing several grounding components; the separate factorial holds scores fixed within decoder comparisons. This design distinguishes procedure-level gains from search effects and readiness from formulation correctness.

3 Problem Setting and Evaluation Framework

3.1 Problem Formulation

Given a query q, the system returns a top-1 catalog schema and a partial mapping of its scalar parameters [1]. Schema retrieval ranks the catalog $\mathcal{S} = \{s_1, \ldots, s_M\}$ of M schemas and returns

$$\hat{s} = \arg\max_{s \in \mathcal{S}} \text{score}(q, s). \tag{1}$$

Parameter instantiation extracts numerical mentions and assigns them to scalar slots of $\hat{s}$ using deterministic rules. Incorrect retrieval can produce an inappropriate eligible slot set even when the query contains usable numerical evidence.

A *schema* comprises the retrieved template and its parameter structure. The core task excludes complete program synthesis and list-, vector- or dictionary-valued recovery. Numerical grounding is rule-based; optional dense retrieval uses a pretrained encoder.

Let $K_G(q)$ contain keys whose gold values are numeric scalars (integers or floats, excluding booleans). Let $P_{\text{all}}(\hat{s})$ contain *all* keys in the predicted schema's parameter-metadata dictionary, with a fallback to all keys in its reference parameter record when the parameter metadata is not a dictionary. This construction does not first filter entries by scalar shape. The benchmark constructs assignments on the eligible set

$$E(q, \hat{s}) = K_G(q) \cap P_{\text{all}}(\hat{s}). \tag{2}$$

Here, gold denotes the stored benchmark reference, not independently verified semantic truth. Gold scalar status therefore filters the slot inventory *before* assignment: gold keys affect both prediction construction and evaluation, although the mention-selection rule does not receive gold numerical magnitudes. Let $y_q : K_G(q) \to \mathbb{R}$ be the gold map and $\hat{y}_q$ the partial assignment on $A(q) \subseteq E(q, \hat{s})$; masked inputs may leave a raw unknown token rather than a numeric value. Retrieval quality measures top-1 schema match, whereas downstream quality measures how well eligible scalars are filled from text.

3.2 Evaluation Inputs and Reference Assistance

The main four-arm and matched-baseline experiments use reference information during prediction, not only for scoring. They are *gold-assisted*: the expected scalar count and slot identities depend on $K_G(q)$ before value selection, so slot-name type heuristics also operate on reference-derived names. The numeric selector does not receive gold magnitudes, objective coefficients, or constraint identities as separate targets. This restriction on the selector's inputs does not make the full pipeline autonomous.

The catalog is also benchmark-derived. Its builder pairs each test description with the same record's parameterized description, falling back to its description and a metadata signature. The retained catalog contains 331 test-record documents plus four fillers; all 331 target IDs are present, and six query–target texts match exactly. This is known-template identification rather than held-out template generation. Fitting an index without supervision does not establish independence from the evaluation source.

The later joint/factorial controls remove the query-specific gold-key filter but retain the selected reference template's slot names, shapes, and descriptions, with types inferred heuristically from names. These *partially assisted* controls also use any available parameter relations from that template and measure assignment conditional on supplied structure. The 20-case execution check is gold-assisted: it additionally retains reference code and unfilled scalar/structured values. Online Resource 1, Section S4.3, details these boundaries and distinguishes prediction reconstruction from aggregation. The saved records identify neither an autonomous formulation score nor an adjustment that would estimate one from these conditional scores.

3.3 Inference Dependencies and Practical Scope

The lexical path uses catalog-fitted term frequency–inverse document frequency (TF–IDF), cosine ranking, numerical parsing, and typed assignment, without generative-model calls or task-specific neural training. Optional BGE-M3 retrieval uses

a pretrained multilingual text encoder to represent queries and schemas as dense vectors; the external generation and relationship studies use model services separately. Cached dependencies and metadata permit local lexical inference, but no comparative cost, privacy, or reliability benefit is measured. Historical replay also requires the run's ordering and environment (Section 6.3).

3.4 Metric Levels and Denominators

Schema R@1 (recall at rank one) is $N^{-1}\sum_{i=1}^{N} H_i$, where $H_i = \mathbf{1}[\hat{s}_i = s_i^*]$ indicates a schema hit, s_i^* is query i's reference schema, and $N = 331$. For downstream metrics, let $E_i = E(q_i, \hat{s}_i)$ and let $A_i \subseteq E_i$ denote the assigned slots. Let $T_i \subseteq A_i$ contain assignments accepted by the implemented coarse-type compatibility predicate. Compatibility is broader than literal type equality: in addition to same-type matches, integer mentions are accepted for float slots, and integer or float mentions for currency slots. Expected types are inferred heuristically from slot names, not from annotated semantic roles. Define

$$c_i = \begin{cases} |A_i|/|E_i|, & |E_i| > 0, \\ 0, & |E_i| = 0, \end{cases} \qquad \textit{Coverage} = \frac{1}{N}\sum_{i=1}^{N} c_i, \tag{3}$$

$$t_i = \begin{cases} |T_i|/|A_i|, & |A_i| > 0, \\ 0, & |A_i| = 0, \end{cases} \qquad \textit{TypeMatch} = \frac{1}{N}\sum_{i=1}^{N} t_i. \tag{4}$$

Coverage and TypeMatch in Eqs. (3) and (4) are *macro-averages over queries* (each query receives equal weight), with zero-denominator queries contributing zero. Separate type-stratified diagnostic files instead pool slots by type. Abstention leaves a slot unfilled and reduces coverage; TypeMatch conditions on assigned slots within each query before averaging across all queries. Missing retrieval yields an empty eligible set and zero coverage and type score. A masked mention can count as assigned without a comparable numerical value, but its implemented unknown type fails the coarse-type predicate.

For a schema-hit query, let $C_i \subseteq A_i$ contain slots with both a numeric prediction and numeric gold value. For each comparable slot,

$$e_{i,s} = \frac{|\hat{y}_{i,s} - y_{i,s}|}{\max(1.0, |y_{i,s}|)}. \tag{5}$$

A slot passes the inclusive tolerance rule if $e_{i,s} \leq 0.20$: absolute error at most 0.20 for $|y| < 1$, otherwise relative error at most 20%. No additional type gate is imposed on this numerical comparison. The conditional denominator counts only queries with a schema hit and at least one comparable slot. With $J = \{i : H_i = 1,\ |C_i| > 0\}$,

$$x_i = \frac{1}{|C_i|}\sum_{s \in C_i} \mathbf{1}[e_{i,s} \leq 0.20], \qquad \textit{Exact20 (on hits)} = \frac{1}{|J|}\sum_{i \in J} x_i. \tag{6}$$

The comma-separated values (CSV) field `exact20` stores x_i, a slot fraction rather than a whole-query indicator. It is blank on schema misses or when C_i is empty. These queries are excluded from the reported Exact20 mean, not assigned zero; an empty J is reported as unavailable. The main-table denominators are 291, 291, 303, and 320 *queries* for TF–IDF baseline, TF–IDF ratio-aware, BGE-M3, and Oracle. Because missing numerical predictions are not counted as errors in this conditional metric, abstention can improve Exact20 while lowering coverage. The method-dependent subsets therefore do not support an unconditional accuracy ranking.

InstantiationReady retains its historical metric identifier for reproducibility and is described here as *coarse readiness*:

$$\textit{InstantiationReady}_i = \mathbf{1}[c_i \geq 0.8 \ \wedge \ t_i \geq 0.8]. \tag{7}$$

The score in Eq. (7) is averaged over all N queries. It checks neither schema correctness nor numerical values and can tolerate unfilled or incompatible slots. StrictInstantiationReady (strict readiness) additionally requires $H_i = 1$, but still does not check values.

$$\textit{StrictInstantiationReady}_i = \mathbf{1}[H_i = 1 \ \wedge \ c_i \geq 0.8 \ \wedge \ t_i \geq 0.8]. \tag{8}$$

Empty eligible or assigned sets fail both readiness criteria. Thus an 80% coarse-readiness rate is not an 80% correct-instance rate.

To state the complete endpoint explicitly, let F_i^G indicate a schema hit with every required scalar assigned, comparable, type-compatible and within tolerance:

$$F_i^G = \mathbf{1}[\, H_i = 1,\ K_G(q_i) \neq \varnothing, \\ A_i = C_i = T_i = K_G(q_i), \\ \max_{s \in K_G(q_i)} e_{i,s} \leq 0.20\,]. \tag{9}$$

An empty required set fails this criterion. Unlike conditional Exact20, F_i^G cannot reward missing keys or incomparable values. Write I_i and S_i for ordinary and strict readiness, and F^G, S and I for the corresponding query sets. Since $F^G \subseteq S \subseteq I$, we also report

$$\rho_I = \frac{\sum_i F_i^G}{\sum_i I_i}, \qquad \rho_S = \frac{\sum_i F_i^G}{\sum_i S_i}. \tag{10}$$

These fractions describe agreement among a method's ready outputs; a zero denominator would be reported as unavailable. They are not semantic precision estimates: the denominators are method-dependent subsets of the same exposed benchmark. These descriptive summaries receive no independence-based uncertainty interval or new hypothesis test.

4 Methods and Experimental Setup

4.1 RQ1: Retrieval Protocol

We use the 331 historically designated NLP4LP test queries [1, 2], each with one target schema in the fixed 335-document catalog (Section 3.2). The variants are *orig* (full text), *noisy* (controlled lexical corruption and numerical masking), and *short* (first sentence). The test designation does not imply an untouched evaluation set because the benchmark was used repeatedly during development.

We compare BM25 probabilistic lexical ranking, TF–IDF cosine retrieval, latent semantic analysis (LSA), and the pretrained BGE-M3 encoder [6, 8, 18, 23]; Table S2 summarizes their representations. Classical indexes fit the catalog without query-label supervision, and only the top-1 schema enters grounding. Uniform random retrieval gives the theoretical reference $1/335 \approx 0.0030$. Oracle selection substitutes the target schema as a control; it is not a guaranteed upper bound on other grounders. BGE-M3 reporting metadata specifies float32 precision and unit-Euclidean-length (L2-normalized) 1024-dimensional embeddings and cosine similarity, but does not authenticate the executed device or complete environment. Dataset tooling uses Hugging Face `datasets` [15].

4.2 RQ2: Grounding and Endpoint Diagnosis

The rule-based extractor recognizes integers, decimals, written numbers, fractions, percentages, currency-like forms, and comma-separated magnitudes. Ratio-aware extraction adds supported multiplicative words such as "twice" and "triple"; masked tokens remain numerically unknown. Surface cues assign percent, integer, currency, or float types to mentions, while parameter-name heuristics determine expected slot types. Eligible slots follow Eq. (2), and compatibility follows Section 3.4.

Typed greedy visits slots p_i in the run's order. With $\mathcal{T}$ the extracted mentions and $\mathcal{U}_{i-1}$ the mentions already used, it assigns an unused indexed mention:

$$t_i^* = \arg \max_{t \in \mathcal{T} \setminus \mathcal{U}_{i-1}} \mathrm{compat}(p_i, t). \tag{11}$$

Here i indexes slots, rather than queries, and compat ranks candidates by coarse type preference, absolute magnitude, and raw surface text, retaining source order for remaining ties. Mention indices cannot be reused, but equal values at distinct occurrences remain distinct candidates. Slot iteration follows Python set order, so replay requires the historical interpreter/hash configuration. The headline configurations omit optional lexical role cues. Algorithm 1 and Figure 1 describe this protocol. In the algorithm, A is the slot–value mapping, $\mathcal{P}$ the eligible inventory, and $\mathcal{C}_i$ the unused-mention candidates; these differ from the evaluation sets A_i and C_i.

The four headline arms are TF–IDF with baseline extraction, TF–IDF with ratio-aware extraction, BGE-M3 with ratio-aware grounding, and Oracle typed greedy. Their saved query metrics support the endpoints and ordered residual

Algorithm 1 Reference-assisted deterministic parameter instantiation without mention-index reuse.

Require: Query text q, predicted schema $\hat{s}$, gold scalar keys $K_G(q)$
Ensure: Slot–value mapping A

Extract numeric mentions $\mathcal{T}$ from q
Infer a coarse type for each mention in $\mathcal{T}$
Obtain parameter keys from schema $\hat{s}$
Set $\mathcal{P} = P_{\text{all}}(\hat{s}) \cap K_G(q)$; use the run's set iteration order
Infer an expected type for each slot in $\mathcal{P}$
Initialize assignments $A \leftarrow \emptyset$ and used-mention set $\mathcal{U} \leftarrow \emptyset$
for each slot $p_i \in \mathcal{P}$ **do**
    Define candidate set $\mathcal{C}_i \leftarrow \mathcal{T} \setminus \mathcal{U}$
    if $\mathcal{C}_i = \emptyset$ **then**
        continue
    end if
    Score mentions in $\mathcal{C}_i$ using a deterministic slot–mention compatibility rule
    Select the highest-ranked mention t_i^*
    Assign $A[p_i] \leftarrow \text{value}(t_i^*)$ if numeric; otherwise retain its raw unknown token
    Update $\mathcal{U} \leftarrow \mathcal{U} \cup \{t_i^*\}$
end for
return A

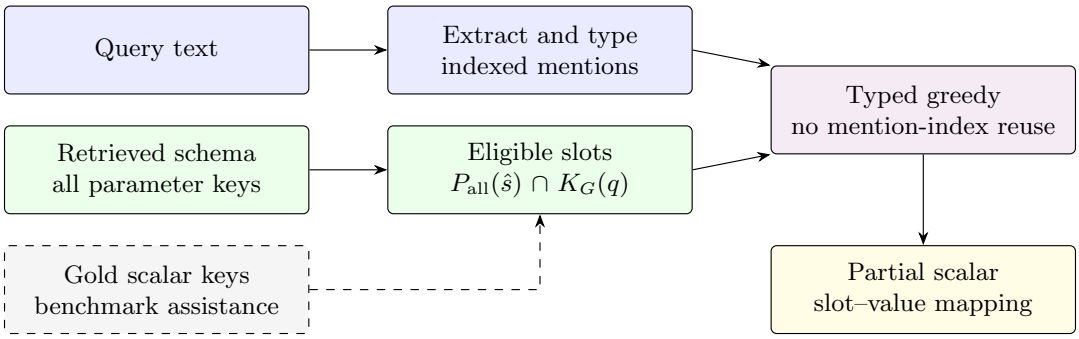


Fig. 1 Historical typed-greedy instantiation with explicit reference assistance. Typed query mentions fill slots formed by intersecting predicted-schema metadata keys (with reference-key fallback) and gold scalar keys. Assignment consumes mention indices, not unique values. The later metadata-only factorial protocol uses a different inventory

partition. Although the two ratio-aware retrieval arms use the same grounding implementation, their frozen assignments differ even on shared hits, so their numerical differences cannot all be attributed to retrieval. Core endpoints require no solver execution. Independent row-indicator and decimal set-intersection calculations verify the readiness-conditioned fractions from authenticated records.

4.3 RQ3: Whole-Procedure and Frozen-Score Comparisons

Six existing procedures share all 331 original queries and TF–IDF retrieval: typed greedy, constrained assignment, maximum-weight matching, optimization-role repair, search-structured grounding, and semantic intermediate-representation (IR) repair. Candidates, context, scoring, abstention, and decoding can differ. Typed greedy and constrained assignment both prevent mention reuse and share ratio-aware patterns, but use different token/mention constructors; the other procedures retain their intrinsic enriched extractors. Methods were selected before this matched comparison, without additional tuning against its labels, although earlier benchmark exposure remains. The comparison therefore evaluates complete procedures rather than isolating the decoder.

The separate factorial holds candidates and edge-score matrices fixed within each decoder

pair. It crosses two candidate constructors, three scorers, two decoders, and two mention-reuse policies: 24 cells on 49 primary queries and 277 scalar slots. Each pair shares canonical lexical slot order and a zero-score null option (leaving a slot unfilled). The ordinal type-preference control assigns numerical scores to ordered preferences; it does not reconstruct every historical greedy decision. The remaining scorers use contextual scores with or without percent/currency marker bonuses. With unrestricted reuse and additive scores, independent per-slot maximization already solves the stated objective. One-to-one matching, which prohibits reuse of a mention index, is thus a diagnostic restriction, not a general requirement of formulation.

The original collection contains 64 cases: 49 primary, 12 development, one overlap-reconciliation, and two reference-only. Query 291 is excluded from the primary slice because its delexicalized group crosses an earlier training/development boundary; this exclusion does not establish independence from prior exposure. The earlier 50-query/280-slot joint pilot and its overlap-free 49-query slice remain distinct. Their saved beam, exact-objective, and value/type-distinct enumeration records support diagnosis of availability, retention, and selection without new decoding or solving. Online Resource 1 (Sections S4.1–S4.3) preserves all populations and controls.

4.4 Retrospective Statistics and Evidence Integrity

All statistical inference is retrospective. The primary 331-query comparisons use 327 frozen delexicalized text groups (formed after masking numerical expressions), marginal 95% group-percentile bootstrap intervals ($B = 10{,}000$ bootstrap resamples, NumPy seed 42), exact groupwise method-swap tests, and Holm adjustment over 16 retrieval/readiness contrasts [11]. Supplementary Table S6 preserves five displayed contrasts; the complete family also includes BGE–TF–IDF retrieval and ten readiness comparisons with grounding alternatives. The release retains exact query-level McNemar checks alongside, rather than in place of, the group tests.

The factorial has a separate 240-test family covering one-factor contrasts and four endpoints. Its historical exploratory bootstrap uses Python's seeded generator, $B = 2000$, seed 42, and the retained order-statistic interval procedure, rather than the primary NumPy procedure. Grouping and multiplicity adjustment neither remove benchmark exposure nor make the marginal intervals simultaneous.

The immutable evidence release authenticates the inputs used here. Headline CSVs preserve query metrics but not original scalar mappings; later cached assignment controls preserve choices. Reaggregation uses each source's own observations, without substituting newer predictions for missing historical ones. Section 6.4 and Online Resource 1, Section S4.3, distinguish public reanalysis from unavailable historical reproduction.

5 Results Organized by Research Question

5.1 RQ1: Fixed-Catalog Schema Retrieval

BGE-M3 identifies the target on 313/331 original queries (94.56%), compared with 301/331 (90.94%) for TF–IDF. The paired discordances are 15 BGE-only and three TF–IDF-only hits; the retrospective exact test gives $p = 0.0075$ and Holm-adjusted $p = 0.0302$ within the 16-test family. Noisy-query rates remain close to their original counterparts, whereas first-sentence truncation reduces every method's retrieval rate (Table 1). The table also reports means across variants, computed before rounding. These results show strong target identification within the fixed catalog, without testing new-schema discovery.

5.2 RQ2: What Readiness Conceals

Ratio-aware TF–IDF passes ordinary readiness on 265 queries (80.06% of 331) and strict readiness on 255; the ordinary-ready set includes ten wrong-schema outputs. Only seven meet the complete scalar-tolerance criterion (2.11% of 331). BGE-M3 and Oracle similarly have 273/331 and 281/331 ordinary-ready cases, but only 3/331 and 8/331 complete successes. Table 2 separates the metric levels. Because Exact20 conditions on comparable schema hits, the larger TF–IDF value does not establish unconditional superiority.

Table 1 Schema retrieval accuracy (recall at rank one, *Schema R@1*) from saved top-1 IDs for 331 queries per variant: original (Orig), noisy, and short; Avg. averages the variants. LSA is latent semantic analysis; TF–IDF is term frequency–inverse document frequency; BM25 is a lexical ranker and BGE-M3 a dense text encoder. Random is the theoretical uniform-selection rate over 335 schemas, $1/335 \approx 0.0030$. Bold marks each column's largest descriptive value, not statistical significance.

Baseline	Orig	Noisy	Short	Avg.
Random	0.0030	0.0030	0.0030	0.0030
LSA	0.8459	0.8882	0.7644	0.8328
BM25	0.8822	0.8912	0.7674	0.8469
TF–IDF	0.9094	0.9033	0.7795	0.8640
BGE-M3	**0.9456**	**0.9426**	**0.8157**	**0.9013**

Table 3 expresses this discrepancy among ready outputs. Of the 265 ordinary-ready ratio-aware outputs, $7/265 = 2.64\%$ satisfy F^G, while $258/265 = 97.36\%$ do not. Adding the schema gate raises the fraction only to $7/255 = 2.75\%$, leaving 248 strict-ready outputs without complete agreement. These fractions are intersections of saved query-level gates, independently checked against their subset relations. They are reanalyses, not new experiments; their method-dependent populations share the exposed benchmark and support neither independent-sample comparisons nor interpretation as semantic precision.

5.2.1 Mutually Exclusive Residual Failures

Table 4 partitions all 331 cases sequentially into schema misses, incomplete eligible coverage, incompatible types, stored-value disagreement, and scalar-tolerance success. Full coverage and type checks require $c_i = t_i = 1$, exceeding readiness's 0.8 thresholds, and missing comparable numerical evidence cannot pass. Every successful case in these four arms also covers all required reference scalar keys; Section 5.3 identifies the eligible-only exception in another method.

For ratio-aware TF–IDF, the partition is $30 + 31 + 49 + 214 + 7 = 331$. Value disagreement accounts for $214/331 = 64.65\%$ of all queries and $214/221 = 96.83\%$ of those passing the schema/full-coverage/full-type gates. Conditional failure rates are $228/231 = 98.70\%$ for BGE-M3 and $239/247 = 96.76\%$ for Oracle, each on a different gate-passing population. Numerical disagreement is therefore the dominant residual failure in these records; the partition does not identify a linguistic cause for each mismatch.

BGE-M3 and ratio-aware TF–IDF share 298 schema hits but differ in Exact20 on 162 of them. Only TF–IDF records its hash seed; missing BGE-M3 runtime details and original scalar maps prevent attribution to retrieval alone. The ratio-word ablation permits a more specific comparison: strict readiness improves by eight cases with no losses ($247 \rightarrow 255$), while complete scalar success rises from six to seven. The exact paired/group $p = 0.0078125$ adjusts to 0.0302. Online Resource 1, Section S2, retains the supporting ablation and schema-gate tables.

5.3 RQ3: Grounding Mechanisms and Their Identifiable Effects

Table 5 compares six grounding procedures with queries and TF–IDF retrieval fixed across all 331 cases. Search-structured grounding has the most complete all-key successes, $46/331 = 13.90\%$, although most queries still fail. Constrained assignment gains 36 complete successes and loses three relative to typed greedy. Its historical eligible-only count is 41; query 102 fills three of seven required keys and is excluded from the all-key count of 40.

Typed greedy has the highest ordinary and strict readiness rates, 0.8006 and 0.7704; all five alternatives are lower, with Holm-adjusted $p \leq 0.0059$ for the ten readiness contrasts. Conditional Exact20 can improve as coverage falls because slots with incomparable values or abstentions are excluded from its denominator. The all-key endpoint cannot count these omissions as complete successes. Its descriptive ordering is not a multiplicity-adjusted superiority claim, and the simultaneous changes in candidates, scoring, abstention, and decoding support a procedure-level trade-off rather than an isolated benefit of global search.

5.3.1 Frozen-Score Assignment on 49 Queries

Table 6 reports the separate 49-query/277-slot population defined in Section 4.3. With contextual

Table 2 Main downstream diagnostics from frozen records on 331 original queries. Bold marks the largest descriptive non-oracle value in each column, not statistical significance. Conditional Exact20 does not establish unconditional superiority.

Method	Coverage	TypeMatch	Exact20	InstReady	Strict
TF–IDF-TG (baseline extr.)	0.8794	0.8515	0.2449	0.7764	0.7462
TF–IDF-TG (ratio-aware extr.)	0.8886	0.8665	**0.2614**	0.8006	0.7704
BGE-M3 + ratio-aware	**0.9154**	**0.8946**	0.2436	**0.8248**	**0.8006**
Oracle-TG	0.9416	0.9230	0.2505	0.8489	0.8489

TG denotes typed-greedy grounding; extr. denotes extraction. TF–IDF-TG uses TF–IDF retrieval; Oracle-TG uses the reference schema. InstReady and Strict abbreviate InstantiationReady and StrictInstantiationReady: coarse readiness without numerical-value checks; Strict also requires the correct schema. Coverage is the assigned fraction of eligible slots, and TypeMatch the compatible fraction of assigned slots, averaged over queries. Coverage, TypeMatch, InstReady, and Strict use all 331 queries. Exact20 macro-averages the per-query slot fractions in Eq. (6) over 291, 291, 303, and 320 comparable schema-hit queries, respectively; noncomparable cases are excluded. Cells aggregate saved query metrics, not retained scalar mappings.

Table 3 Complete stored-reference scalar agreement among ordinary-ready (I) and strict-ready (S) outputs (Eq. (10)). Fractions give complete successes (F^G) over ready outputs. Percentages are descriptive; denominators are method-dependent subsets of the same 331-query frame.

	Ordinary-ready		Strict-ready	
Method	$\lvert F^G\rvert/\lvert I\rvert$	(%)	$\lvert F^G\rvert/\lvert S\rvert$	(%)
TF–IDF base	6/257	2.33	6/247	2.43
TF–IDF ratio	7/265	2.64	7/255	2.75
BGE-M3	3/273	1.10	3/265	1.13
Oracle	8/281	2.85	8/281	2.85

scores and one-to-one assignment, replacing greedy with exact decoding raises exact slot counts from 106 to 136 for typed candidates and from 100 to 133 for rich candidates. Full-tolerance success remains 5/49, with one gained and one lost query. The corresponding historical exact-slot bootstrap intervals are $[3.14, 17.95]$ and $[4.07, 18.84]$ percentage points; the full-tolerance difference interval is $[-6.12, 6.12]$. Neither contextual decoder contrast survives adjustment within the separate 240-test Holm family (adjusted $p = 1$).

All additive-score decoder pairs with unrestricted mention reuse agree, as the separable objective predicts. Some contextual-versus-ordinal slot contrasts survive multiplicity adjustment, but no whole-instance endpoint does. The controls isolate conditional assignment differences without establishing whole-instance equivalence or explaining all 33 additional historical constrained successes on the 331-query frame. Recorded sub-millisecond cell times include instrumentation and do not support clean speed comparisons.

5.3.2 Candidate Availability, Scoring and Search Retention

In the earlier 50-query joint pilot, 32 common pools admit a criterion-correct mapping; the beam retains such a mapping in eight top-ten lists and selects one in three cases. Among those 32 pools, exact-objective diagnosis identifies 23 strictly preferred incorrect optima, six correct/incorrect ties, and three selected correct optima. Candidate availability therefore guarantees neither retention nor favorable ranking. The explicit-constraint variant activated no supported parameter-only relations, so its unchanged result is a no-op rather than evidence against useful constraints.

On the overlap-free 49-query slice, both the strongest local selection and joint selection retain three full-tolerance successes. Value/type-distinct exact enumeration leaves list retention at 9/49 and selected success at 3/49, with zero gained or lost cases. These negative findings limit what additional search achieves for the frozen objectives: better candidate coverage or slot assignment need not improve whole-instance agreement, and exhaustive search cannot correct an objective that strictly prefers an incorrect mapping. Online Resource 1, Section S4.1, retains the full pruning-stage counts, oracle restrictions, and all 24 cells.

Table 4 Mutually exclusive residual partition from saved metrics on 331 original queries per configuration. Categories apply in the listed order. TF–IDF columns differ only in extraction; differences involving BGE-M3 are descriptive.

Category	TF–IDF base	TF–IDF ratio	BGE-M3	Oracle
Wrong schema	30	30	18	0
Incomplete coverage	48	31	33	35
Type mismatch	61	49	49	49
Stored-value disagreement	186	214	228	239
Full scalar agreement	6	7	3	8
Total	331	331	331	331

Full coverage and type compatibility require $c_i = t_i = 1$. Full scalar agreement requires all eligible scalar values within the inclusive tolerance in Eq. (5); it does not certify complete model correctness.

Table 5 Matched grounding baselines from saved metrics on 331 original queries, with TF–IDF retrieval and the gold-assisted inventory rule fixed. Bold marks the largest descriptive value in each column, not statistical significance. Procedures change multiple components; conditional Exact20 denominators can differ.

Grounding method	Coverage	TypeMatch	Exact20	InstReady	Strict	F^G
Typed greedy (reference)	**0.8886**	0.8665	0.2614	**0.8006**	**0.7704**	7/331
Constrained assignment	0.8531	0.8745	0.5772	0.7613	0.7311	40/331
Max-weight matching	0.8436	0.8740	0.5225	0.7432	0.7130	37/331
Optimization-role repair	0.8436	0.8724	0.5210	0.7372	0.7069	35/331
Search-structured grounding	0.8195	**0.8795**	**0.6397**	0.7039	0.6737	**46/331**
Semantic-IR repair	0.8131	0.8483	0.4949	0.7160	0.6949	9/331

Coverage and TypeMatch are query-averaged slot coverage and type compatibility. InstReady denotes coarse readiness; Strict additionally requires a schema hit. F^G requires the correct schema, all gold scalar keys, compatible types, and complete stored-reference scalar tolerance; every row uses denominator 331. IR denotes intermediate representation. Typed greedy and constrained assignment share ratio-aware patterns but differ in token/mention construction, context, scoring, abstention, and decoding; their comparison does not isolate global search. Other methods use intrinsic enriched extractors. Exact20 (on hits), the query-averaged within-tolerance slot fraction, uses 291 comparable schema-hit queries except for semantic-IR repair (285); noncomparable cases are excluded, as in Table 2.

6 Discussion

6.1 Interpreting RQ1–RQ3 Together

Target-schema identification, readiness, and complete stored-reference scalar agreement answer different diagnostic questions. Strong target identification does not ensure complete scalar agreement. Readiness checks whether enough eligible slots receive type-compatible values; the residual partition locates the first failure under stricter gates, while readiness-conditioned agreement evaluates outputs that pass the coarse checks. These summaries locate numerical failures without establishing their semantic causes.

The contrasting readiness and complete-agreement outcomes also show why optimizing one intermediate score can be misleading. Abstention changes coverage and the conditional Exact20 population, while only fixed-candidate, fixed-score comparisons isolate decoder behavior. The 49-query controls and exact-objective diagnostics distinguish slot-level gains, whole-instance outcomes, pruning failures, and incorrect mappings preferred by the score. Their finite observations do not establish that global search, learned representations, or mathematical constraints are generally ineffective.

These results support joint reporting of the endpoint sequence, supplied information, and conditional populations, with causal attribution limited to controls that hold other components fixed. This approach complements Text2Opt-Bench's

Table 6 One-to-one factorial controls on 49 queries and 277 gold scalar slots, distinct from the historical 331-query configurations. Each pair shows greedy → exact maximum-sum assignment. Typed and Rich denote the typed-token and richer contextual mention parsers; Ordinal uses numerical ranks of type preferences, Context uses contextual scores, and No marker bonus removes percent/currency bonuses.

Pool	Scorer	E slots	T slots	E full	T full
Typed	Ordinal	71 → 71	87 → 86	3 → 3	3 → 3
Typed	Context	106 → 136	117 → 145	3 → 4	5 → 5
Typed	No marker bonus	97 → 137	111 → 146	3 → 4	5 → 5
Rich	Ordinal	67 → 67	81 → 81	3 → 3	3 → 3
Rich	Context	100 → 133	111 → 142	3 → 4	5 → 5
Rich	No marker bonus	97 → 135	111 → 144	3 → 4	5 → 5

E slots and T slots count numerically exact and tolerance-correct scalar assignments, respectively. E full and T full count queries for which every gold scalar key is assigned a type-compatible value and all values meet the corresponding numerical criterion. Pairs compare decoders within each pool/scorer configuration; no best-value ranking is imposed across configurations.

modeling–binding analysis [10], Refai and Ahmed's component-level formulation evaluation [22], and OPT-Engine's complexity-based diagnosis [7]. Its specific contribution is readiness-conditioned complete reference agreement and fixed-score attribution in supplied-slot grounding. Provenance-bearing bindings and explicit abstention may aid practical review, but their deployment effectiveness remains unmeasured.

6.2 Where Numerical Diagnostics Stop

Stored-reference agreement does not establish semantic fidelity: equal values may come from different occurrences, and the historical tolerance can accept materially different small proportions. Formulation equivalence additionally concerns variable mappings, feasible sets, and objectives [28]. The supplementary studies illustrate these limits; they do not constitute a fourth central research question.

The 217-relationship study covers three services and 16 selected NL4Opt source problems. Historical repair and recovery yield 631/651 structurally valid judgments and 197 complete cases with chance-corrected inter-rater agreement, Fleiss' $\kappa = 0.371$. The 119 determinate unanimous, 77 disputed, and one unresolved complete cases describe agreement among assertions, not service correctness. No independently accepted human labels are available, and recovery changes the conditions and complete-case membership. Online Resource 1, Section S7.2, preserves the full validity and agreement accounting.

The author-reviewed semantic contrast set (D1) comprises 12 pairs containing 24 author/AI-assisted samples across eight categories: ten meaning-changing pairs and two paraphrases. They illustrate bounds, units, roles, and logical relations that numerical overlap cannot resolve. They provide neither provider predictions nor independently blinded gold, and they do not explain measured pipeline error frequencies. Online Resource 1, Section S10.2, preserves all final decisions and annotation dependencies.

The formulation comparison on the same 18 selected problems (common-18), OptMATH extraction check, and historical 60/269/20-case proxies/executions address separate questions (Online Resource 1, Sections S5, S10.1, and S8.1). Heterogeneous generation protocols preclude a common leaderboard, and PaMOP objective agreement remains withheld because of unresolved precision. Extraction recall does not establish full-framework transfer, nor do the subset proxies establish numerical correctness. Their positive and negative findings, single-run timings, and auxiliary learning histories remain available as supporting evidence without determining the core conclusions.

6.3 Limitations and Threats to Validity

Task and reference validity. Section 3.2 defines the reference-assisted task in an exposed, known

catalog; its scores do not identify autonomous performance. The scalar endpoint excludes structured parameters and model equivalence. Metadata defects can omit keys: query 102 exposes three of seven, and query 293 has no eligible slots. The supplementary reference overlay is not maintainer-certified gold, and the absolute 0.20 tolerance below unit magnitude remains permissive. Section S4.2 documents source-based corrections without rewriting historical predictions.

Sampling and dependence. The evidence relies primarily on one benchmark and its reused templates. Grouping and the query-291 exclusion address known dependence without creating unseen templates; the medium-overlap stratum contains only four queries. Historical learned scorers used evaluation-origin splits with confirmed group crossings, and statistical families were specified retrospectively. Selected proxy, common-18, and relationship populations do not support population-wide rankings. Model agreement and assisted author labels cannot serve as independent truth under the semantic evidence conditions described above.

Historical reconstruction. Unrecovered records include original headline scalar maps, complete manifests and outcomes for the 60/269/20-case subsets, full historical ranked lists, and some rendered prompts and provider settings. BGE-M3 lacks an authenticated hash seed, device, and environment; PaMOP lacks sufficient full-precision returns; and corrected OptMATH per-instance bootstrap counts are missing. Later caches cannot replace these observations. Public reanalysis is limited to the released records described below.

6.4 Reanalysis, Availability and Future Validation

The public `https://github.com/SoroushVahidi/combinatorial-opt-agent/releases/tag/jdsa-evidence-v2.0.0` package supports deterministic aggregation of saved records, grouped statistical checks, and sanitized ledger/projection analysis. Online Resource 1, Section S4.3, gives its exact Hugging Face revision and source hashes. Projections alone cannot support the raw source-dependent checks that passed locally. Complete local originals from the evidence-validation audit (M5), private reviews, and gated or permission-unclear third-party content remain nonpublic; historical v1.2 remains a separate frozen release.

The autonomous-inventory and slot-participation feasibility audits (G1–G3) found neither a validated autonomous inventory path nor an informative native fixed-slot filter-removal comparison. Online Resource 1, Section S6, preserves their static participation accounting; these findings are not measured zero effects. Stronger validation requires autonomous inventory discovery, retained occurrence-level predictions, independently reviewed templates, and full-precision execution records. The separate independent relationship-review protocol remains incomplete (Section S7.4), and D1 does not complete it. These are requirements for future studies, not experiments performed here.

7 Conclusion

Readiness must be interpreted alongside complete scalar agreement. TF–IDF and BGE-M3 identify the target schema for 301 and 313 of 331 queries, respectively (RQ1), yet ratio-aware TF–IDF achieves complete stored-reference scalar agreement on only seven of its 265 ready outputs. The ordered residual partition identifies numerical values as the dominant remaining failure category (RQ2). Matched procedures reach 46 complete successes while changing multiple components; frozen-score controls improve slot counts without increasing the five full-tolerance successes in their separate 49-query population (RQ3).

The contribution is a joint diagnostic account of retrieval, readiness, and complete scalar agreement, with explicit assistance, conditional populations, and limits on attribution. The findings remain conditional on supplied structure, exposed populations, and archived observations. Public saved-record reanalysis makes these measurements inspectable within the documented reconstruction and semantic-validation limits; it establishes neither autonomous formulation accuracy nor independent semantic truth.

Supplementary information. Online Resource 1 contains secondary results, statistical and recovery protocols, evidence provenance and the twelve semantic contrast pairs. Its section, table and equation numbers carry an S prefix.

Statements and Declarations

Funding. This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors. The external large-language-model baseline experiments used Microsoft Azure OpenAI services; associated usage was supported by the standard Azure for Students educational credit available to the author. NJIT/Wulver infrastructure access and complimentary Secure ShellFish access were institutional and in-kind assistance. The archived relationship-labeling runs identify providers but do not establish which research-credit awards paid for their use.

Competing interests. The author declares that there are no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Ethics approval and consent to participate. Not applicable; this study did not involve human participants, human data, or animals. No independently accepted human reference-annotation study was completed for the 217 relationships. D1 comprises author-reviewed synthetic contrasts with language-model assistance, rather than participant data; the separate independent validation protocol remains uncompleted.

Consent for publication. Not applicable; this study did not involve participant data.

Author contributions. Soroush Vahidi: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Writing – review & editing, Visualization. This single-author paper uses the standard CRediT taxonomy to describe the contributions of the sole author.

Data availability. The Hugging Face evidence at https://huggingface.co/datasets/SoroushVahidi/nl-optimization-instantiation-metrics, revision f940128722l1ab6a828147bb138fbe8ab695b982, directory releases/jdsa-evidence-v2.0.0, publishes sanitized metrics, assignment score/choice projections, relationship label/validity ledgers, common-18 numerical outcomes and author/AI-assisted D1 cases. Historical v1.2 is preserved separately at revision 24a047fa5d661be6cfcf75980ac8f62e6c942243. Author-owned evidence is CC BY-NC-SA 4.0 without relicensing third-party material. Complete M5 originals, raw provider prose and private reviews remain local. NLP4LP requires separate gated access; OptMATH source licensing and provider-specific raw-output redistribution permissions remain unresolved. The package supplies upstream-access instructions, not those source corpora.

Code availability. The project repository is https://github.com/SoroushVahidi/combinatorial-opt-agent. The immutable jdsa-evidence-v2.0.0 release linked in Section 6.4 contains MIT-licensed verification code, manifests and pinned reanalysis dependencies. The baseline manuscript sources, article and Online Resource 1 are archived in https://github.com/SoroushVahidi/combinatorial-opt-agent/releases/tag/jdsa-manuscript-v1.0.0, with a separate claim-location crosswalk and audit reports; they do not overwrite the evidence release. The numerical-summary derivation is recorded separately from its authenticated inputs. The current corrected sources and matching PDFs are available at https://github.com/SoroushVahidi/combinatorial-opt-agent/tree/main/manuscript/jdsa-stage3; this remains a pre-submission author-review version.

Reproducibility. The public package reproduces supported saved-record calculations, not the original inference experiments. Sanitized relationship and common-18 projections support ledger/numerical reanalysis but omit raw responses, programs and source witnesses needed for full validation. Missing observations and runtime details are listed in Section 6.3; no newly generated output substitutes for them. All editorial and consistency calculations use authenticated records without model inference or solver experiments. Online Resource 1, Section S4.3, gives exact revisions, commands and evidence classifications.

Declaration of generative AI and AI-assisted technologies in the writing process. The author used ChatGPT and Gemini to improve readability and language. OpenAI Codex assisted manuscript editing, evidence organization and deterministic validation. The author reviewed and edited the original work and remains responsible for its content.

Declaration of AI assistance in software development. The author used Claude, Cursor and GitHub Copilot for coding assistance and reviewed, verified, tested and edited the resulting code. OpenAI Codex later assisted code corrections and offline verification scripts. The author retains

responsibility for the implementation and reported results. Scientific LLM calls are identified separately; the evaluated scalar-grounding procedure uses no generative LLM inference.

References

[1] AhmadiTeshnizi A, Gao W, Udell M (2024) NLP4LP: A dataset for natural language to linear programming and mixed-integer linear programming. Hugging Face dataset repository, https://huggingface.co/datasets/udell-lab/NLP4LP, released with OptiMUS (ICML 2024); gated dataset

[2] AhmadiTeshnizi A, Gao W, Udell M (2024) OptiMUS: Scalable optimization modeling with (MI)LP solvers and large language models. In: Proceedings of the 41st International Conference on Machine Learning, Proceedings of Machine Learning Research, vol 235. PMLR, pp 577–596, https://proceedings.mlr.press/v235/ahmaditeshnizi24a.html

[3] Ahmed T, Choudhury S (2025) OPT2CODE: A retrieval-augmented framework for solving linear programming problems. Natural Language Processing Journal 13:100185. https://doi.org/10.1016/j.nlp.2025.100185

[4] Almasian S, Kazakova V, Göldner P, et al (2023) CQE: A comprehensive quantity extractor. In: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, pp 12845–12859, https://doi.org/10.18653/v1/2023.emnlp-main.793

[5] Astorga N, Liu T, Xiao Y, et al (2025) Autoformulation of mathematical optimization models using LLMs. In: Proceedings of the 42nd International Conference on Machine Learning, Proceedings of Machine Learning Research, vol 267. PMLR, pp 1864–1886, https://proceedings.mlr.press/v267/astorga25a.html

[6] Chen J, Xiao S, Zhang P, et al (2024) M3-Embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. In: Findings of the Association for Computational Linguistics: ACL 2024. Association for Computational Linguistics, pp 2318–2335, https://doi.org/10.18653/v1/2024.findings-acl.137

[7] Chen Y, Cheng C, Sun Y, et al (2026) OPT-Engine: Benchmarking the limits of LLMs in optimization modeling via complexity scaling. In: Proceedings of the 43rd International Conference on Machine Learning, Proceedings of Machine Learning Research, vol 306. PMLR, pp 13807–13834, https://proceedings.mlr.press/v306/chen26o.html

[8] Deerwester S, Dumais ST, Furnas GW, et al (1990) Indexing by latent semantic analysis. Journal of the American Society for Information Science 41(6):391–407. https://doi.org/10.1002/(SICI)1097-4571(199009)41:6<391::AID-ASI1>3.0.CO;2-9

[9] Deng X, Awadallah AH, Meek C, et al (2021) Structure-grounded pretraining for text-to-SQL. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Association for Computational Linguistics, pp 1337–1350, https://doi.org/10.18653/v1/2021.naacl-main.105

[10] Gao Z, Ge A, Berenbeim AM, et al (2026) Models can model, but can't bind: Structured grounding in text-to-optimization. In: Conference on Language Models (COLM 2026), pre-proceedings, https://proceedings.colm.cc/paper_files/paper/2026/hash/6feb9b30798abcfae937760d183605e1-Abstract-Conference.html

[11] Holm S (1979) A simple sequentially rejective multiple test procedure. Scandinavian Journal of Statistics 6(2):65–70. https://www.jstor.org/stable/4615733

[12] Huang C, Tang Z, Hu S, et al (2025) ORLM: A customizable framework in training large models for automated optimization modeling. Operations Research 73(6):2986–3009. https://doi.org/10.1287/opre.2024.1233

[13] Kadıoğlu S, Dakle PP, Uppuluri K, et al (2024) Ner4Opt: Named entity recognition for optimization modelling from natural language. Constraints 29(3–4):261–299. https://doi.org/10.1007/s10601-024-09376-5

[14] Kuhn HW (1955) The Hungarian method for the assignment problem. Naval Research Logistics Quarterly 2(1–2):83–97. https://doi.org/10.1002/nav.3800020109

[15] Lhoest Q, Villanova del Moral A, Jernite Y, et al (2021) Datasets: A community library for natural language processing. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. Association for Computational Linguistics, pp 175–184, https://doi.org/10.18653/v1/2021.emnlp-demo.21

[16] Lu H, Xie Z, Wu Y, et al (2025) OptMATH: A scalable bidirectional data synthesis framework for optimization modeling. In: Proceedings of the 42nd International Conference on Machine Learning, Proceedings of Machine Learning Research, vol 267. PMLR, pp 40769–40802, https://proceedings.mlr.press/v267/lu25o.html

[17] Pan X, Fang J, Wu F, et al (2025) Guiding large language models in modeling optimization problems via question partitioning. In: Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence (IJCAI-25), pp 2657–2665, https://doi.org/10.24963/ijcai.2025/296

[18] Pedregosa F, Varoquaux G, Gramfort A, et al (2011) Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12:2825–2830

[19] Punyakanok V, Roth D, Yih Wt, et al (2004) Semantic role labeling via integer linear programming inference. In: COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics. COLING, pp 1346–1352, https://aclanthology.org/C04-1197/

[20] Ramamonjison R, Li H, Yu T, et al (2022) Augmenting operations research with auto-formulation of optimization models from problem descriptions. In: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: Industry Track. Association for Computational Linguistics, pp 29–62, https://doi.org/10.18653/v1/2022.emnlp-industry.4

[21] Ramamonjison R, Yu T, Li R, et al (2022) NL4Opt competition: Formulating optimization problems based on their natural language descriptions. In: Proceedings of the NeurIPS 2022 Competitions Track, Proceedings of Machine Learning Research, vol 220. PMLR, pp 189–203, https://proceedings.mlr.press/v220/ramamonjison23a.html

[22] Refai D, Ahmed M (2026) A component-level evaluation framework for diagnosing LLM errors in optimization modelling. Neurocomputing 695:133981. https://doi.org/10.1016/j.neucom.2026.133981

[23] Robertson S, Zaragoza H (2009) The probabilistic relevance framework: BM25 and beyond. Foundations and Trends in Information Retrieval 3(4):333–389. https://doi.org/10.1561/1500000019

[24] Roy S, Roth D (2017) Unit dependency graph and its application to arithmetic word problem solving. In: Proceedings of the AAAI Conference on Artificial Intelligence, https://doi.org/10.1609/aaai.v31i1.10959

[25] Scholak T, Schucher N, Bahdanau D (2021) PICARD: Parsing incrementally for constrained auto-regressive decoding from language models. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, pp 9895–9901, https://doi.org/10.18653/v1/2021.emnlp-main.779

[26] Wang B, Shin R, Liu X, et al (2020) RAT-SQL: Relation-aware schema encoding and linking for text-to-SQL parsers. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Association

for Computational Linguistics, pp 7567–7578, https://doi.org/10.18653/v1/2020.acl-main.677

[27] Xiao Z, Wang YJ, Han X, et al (2026) DeepOR: A deep reasoning foundation model for optimization modeling. In: Proceedings of the Fortieth AAAI Conference on Artificial Intelligence (AAAI-26), pp 34052–34060, https://doi.org/10.1609/aaai.v40i40.40699

[28] Zhai H, Lawless C, Vitercik E, et al (2025) EquivaMap: Leveraging LLMs for automatic equivalence checking of optimization formulations. In: Proceedings of the 42nd International Conference on Machine Learning, Proceedings of Machine Learning Research, vol 267. PMLR, pp 74288–74305, https://proceedings.mlr.press/v267/zhai25a.html