CORS·SCRO
Canadian Operational Research Society
Société canadienne de recherche opérationnelle

# Decision Making in the Era of Large-Language Models: An Overview of Ner4Opt and Text2Zinc

*Canadian Operational Research Society (CORS), Edmonton, Canada, 2025*


Serdar Kadıoğlu

[1] Dept. of Computer Science, Brown University

[2] AI Center of Excellence, Fidelity Investments

skadio.github.io


BROWN

# Learning & Reasoning

## Data Science: ML/DL/NLP/LLMs/etc.

Focuses on **machine learning using historical data** to identify patterns and make predictions. Excels at pattern recognition, classification, and forecasting.

### System 1 - Predictive Models

- Learning from historical data patterns
- Probabilistic predictions and insights
- Ideal for unstructured problems
- Applications include recommendation systems, image recognition, and natural language processing

## Decision Science: OR/MIP/CP/SAT/LS/etc.

Focuses on **combinatorial satisfaction and optimization** using logical and mathematical models. Provides provable optimality and explicit reasoning.

### System 2 - Prescriptive Models

- Mathematical and logical formulations
- Provably optimal for deterministic environments
- Perfect for structured problems
- Applications include verification, planning, scheduling, routing, and resource allocation

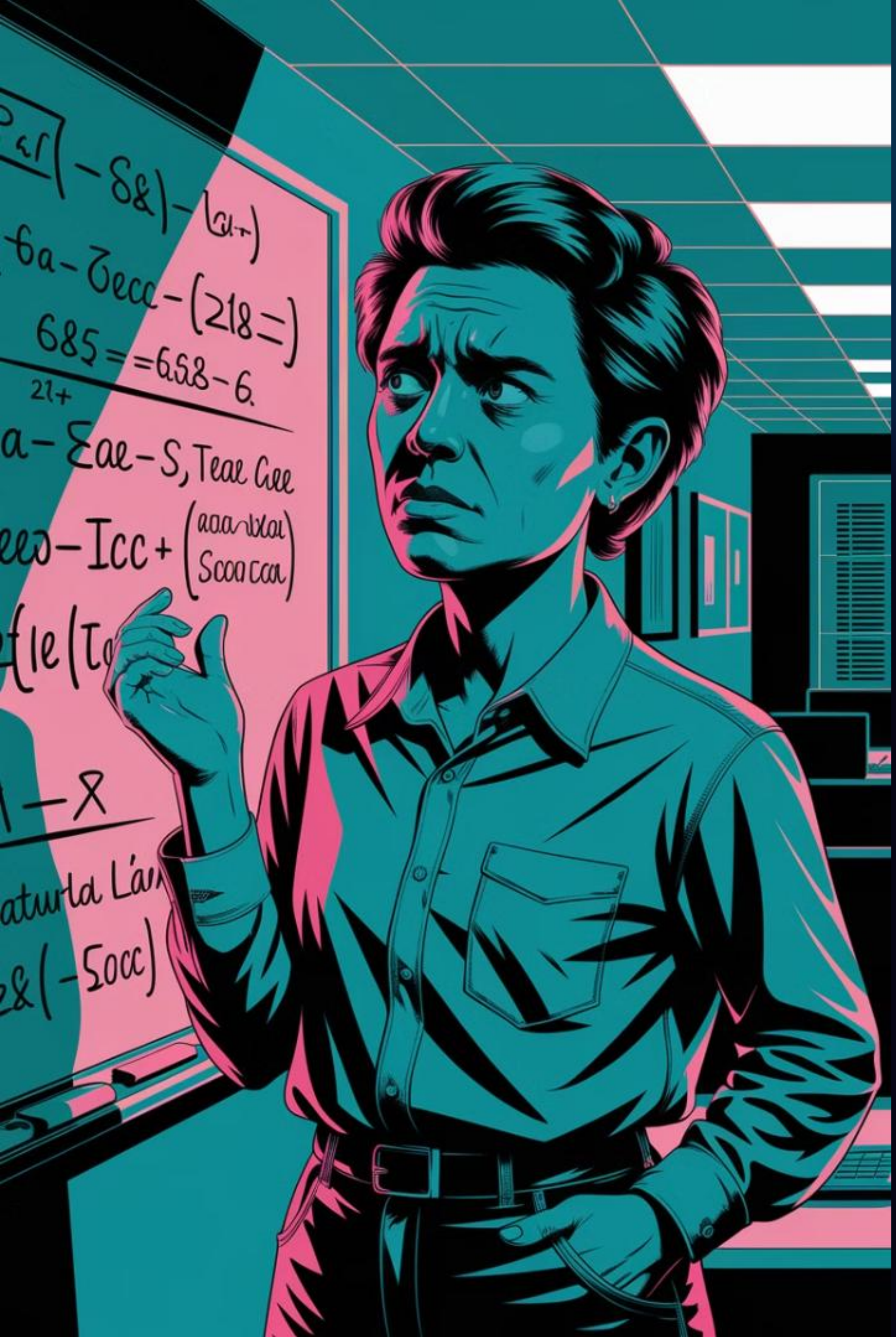
# Integration with Optimization Technology

## Existing ML-OR Integration

- Algorithm configuration procedures
- Variable and constraint selection
- Branching strategies
- Cut selection
- Node selection
- Tree-search configuration

## Emerging NLP-OR Integration

- Named entity recognition for optimization
- Natural language interfaces for solvers
- Automated model formulation
- Explanation generation
- Interactive modeling assistants
- Domain-specific optimization co-pilots



# The De-Facto Model-and-Run Strategy

1 
## Problem Description

Users **describe optimization problems** in natural language, which contains ambiguous references to variables, constraints, and objectives that must be precisely identified.

2 
## Model Formulation

Experts must **manually transform problem descriptions** into formal mathematical models, a process that requires specialized knowledge and is prone to errors.

3 
## Solution Finding

Once properly modeled, optimization solvers can find optimal solutions, but the **modeling barrier** remains a significant obstacle to wider adoption of optimization technology.

# Decision Making in the Era of Large-Language Models

## 1 Reasoning: Optimization

- Optimization technology and constraint solving techniques are powerful and have many applications.
- The cognitive barrier of translating problem descriptions into formal constraint models persists.

## 2 Learning: Large-Language Models

- LLMs have found success in many fields recently.
- However, they still face challenges in generating constraint models from free-form natural language text.

# Our Contributions

## Ner4Opt

A principled approach to **extracting components of optimization models** such as the objective, variables, and constraints from free-form natural language text.

## Text2Zinc

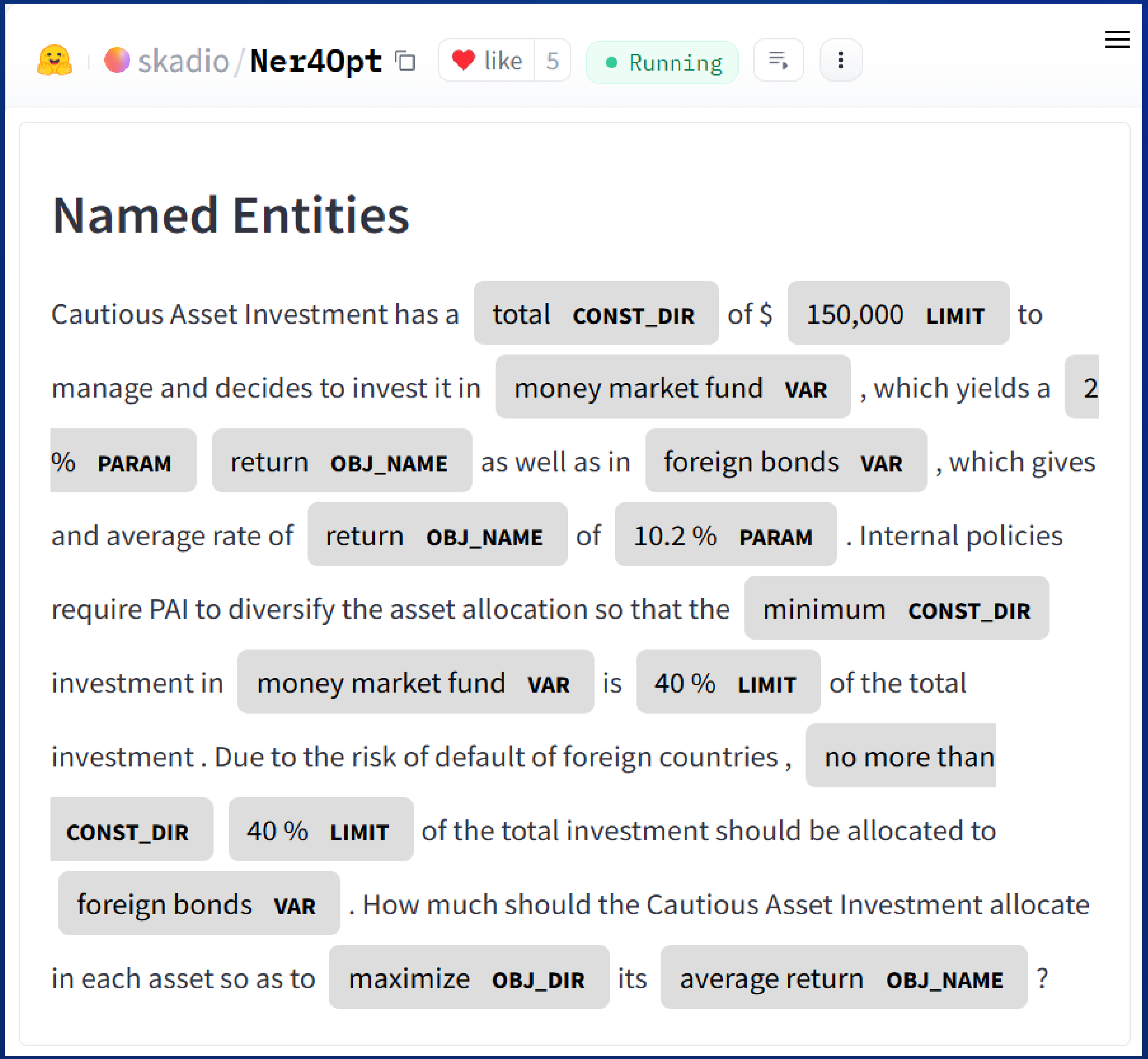
A **unified cross-domain dataset** curated to work with **LLM co-pilots** and an associated **leaderboard** to evaluate strategies to generate **MiniZinc models** from free-from natural language text.

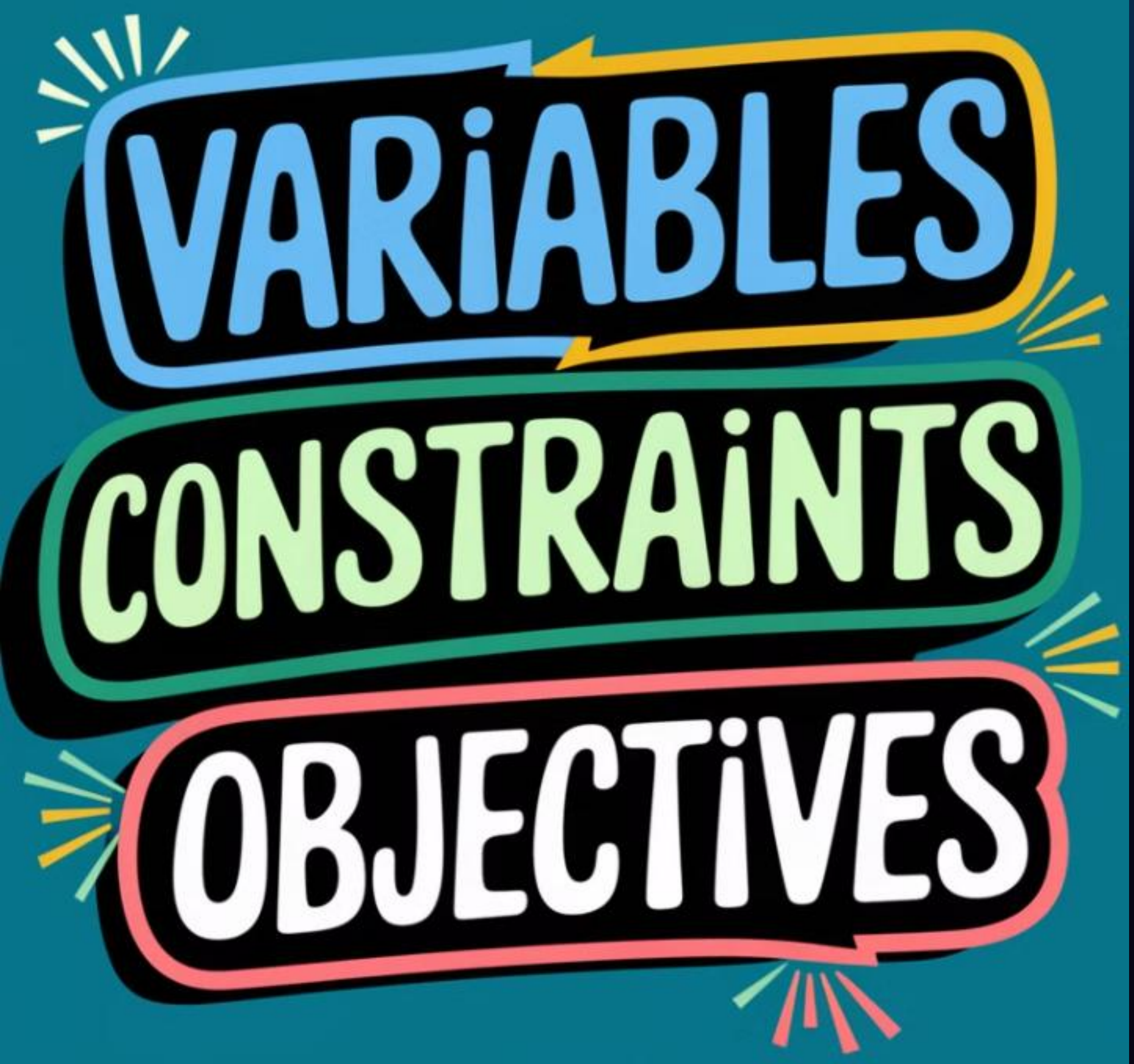
# Our Contributions

## Ner4Opt

A principled approach to **extracting components of optimization models** such as the objective, variables, and constraints from free-form natural language text.

Kadıoğlu et. al. Ner4Opt [Constraints'24] ACP YouTube






# Introducing Ner4Opt

## Named Entity Recognition

Ner4Opt extends traditional named entity recognition to identify optimization-specific components like **variables, parameters, constraints, limits, and objectives** from natural language text.

## Optimization Context

Unlike standard NER which focuses on people, places, and organizations, **Ner4Opt** targets elements needed for mathematical optimization models across **diverse application domains**.

## Modeling Assistance

By automatically extracting these entities, Ner4Opt helps bridge the gap between problem descriptions and formal optimization models, making **optimization technology more accessible**.

# Unique Challenges of LLM + Opt

## 1 Domain-Agnostic Generalization

## 2 Low Data Regime

A doctor can prescribe two types of medication for high glucose levels , a diabetic pill **VAR** and a diabetic shot **VAR** . Per dose , diabetic pill **VAR** delivers 1 **PARAM** unit of glucose reducing medicine and 2 **PARAM** units of blood pressure reducing medicine **OBJ_NAME** . Per dose , a diabetic shot **VAR** delivers 2 **PARAM** units of glucose reducing medicine and 3 **PARAM** units of blood pressure reducing medicine **OBJ_NAME** . In addition , diabetic pills **VAR** provide 0.4 **PARAM** units of stress and the diabetic shot **VAR** provides 0.9 **PARAM** units of stress . At most **CONST_DIR** 20 **LIMIT** units of stress can be applied over a week and the doctor must deliver at least **CONST_DIR** 30 **LIMIT** units of glucose reducing medicine . How many doses of each should be delivered to maximize **OBJ_DIR** the amount of blood pressure reducing medicine **OBJ_NAME** delivaered to the patient ?

Inherent ambiguity in entity boundaries and classifications creates challenges even for human annotators, placing an upper bound on achievable performance.

Optimization problems exhibit significant variability in linguistic patterns, problem structures, and application domains, making entity recognition more challenging.

# Technical Approaches to Ner4Opt

## Classical NLP

**Feature engineering** with Conditional Random Fields (CRF) leverages grammatical, morphological, and syntactic info.

Custom features like gazetteers and automata capture **optimization specific patterns**.

## Modern Language Models

**Transformer-based** approaches like RoBERTa and XLM-RB generate contextual embeddings that capture semantic relationships.

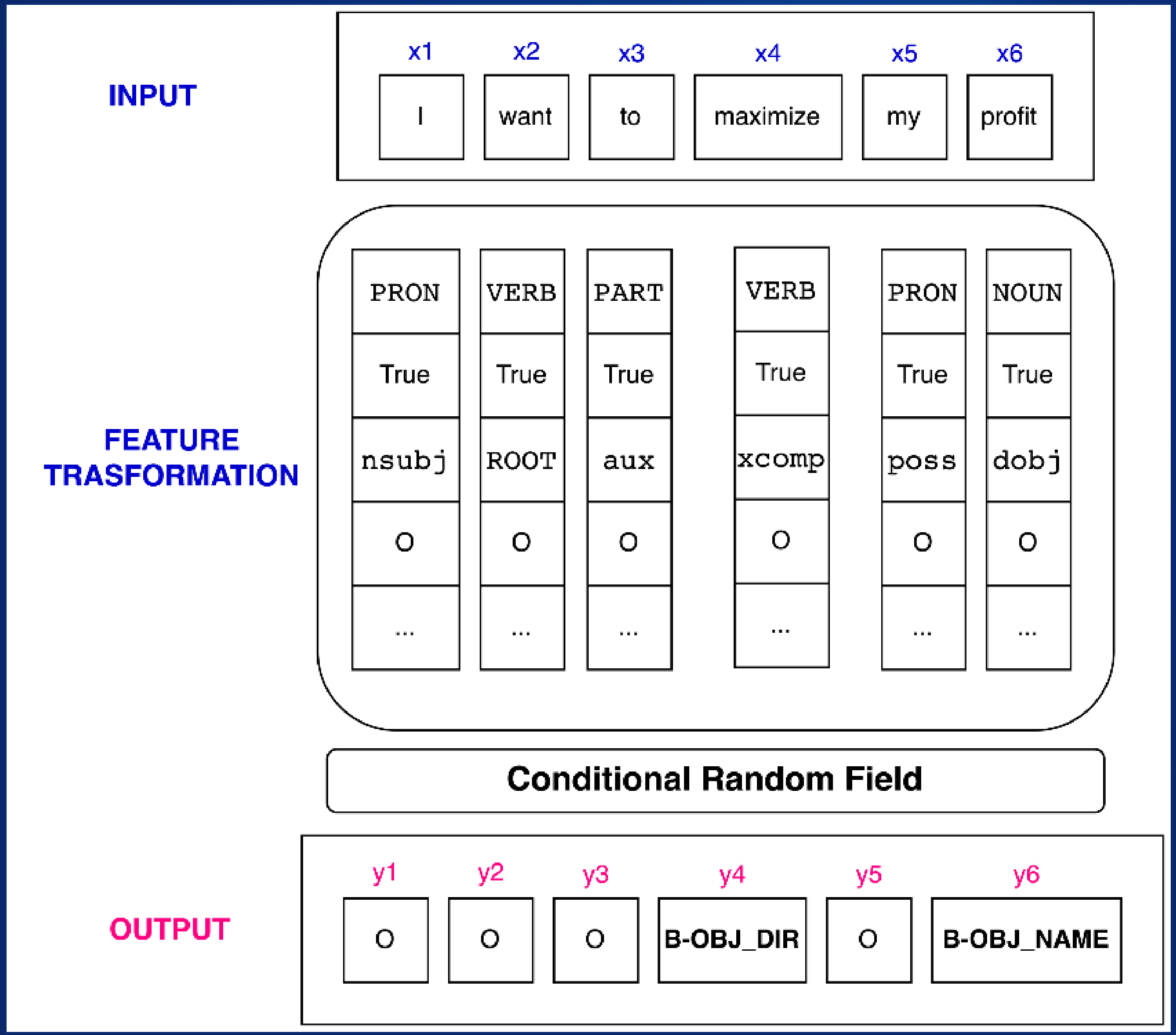
These models are **fine-tuned on optimization corpora** to improve domain-specific understanding.

## Hybrid Solutions

**Combination** of classical feature engineering with modern language models yields the best performance.

**Data augmentation** techniques address challenges like long-range dependencies and disambiguation between variables and objectives.

# Classical NLP Approach




## Feature Extraction

Extract linguistic properties of tokens, including **grammatical features** (part-of-speech, dependency relations), **morphological features** (prefixes, suffixes), and **syntactic features** (noun phrases).

## Model Training

Train the **CRF model** using maximum likelihood estimation on labeled examples, finding optimal weights for feature functions.

# Classical+: Feature Engineering for Optimization

### Gazetteer Features

Lookup tables serving as noisy priors to entity labels, capturing **common keywords** and phrases like "**maximize**" and "**minimize**" for objective direction, or "**at least**" and "**at most**" for constraint direction.

### Syntactic Features

Patterns capturing the unique syntactical properties of variables and objective names, such as **conjuncting noun chunks**, **prepositional chunks**, or elements connected by hyphens or quotes.

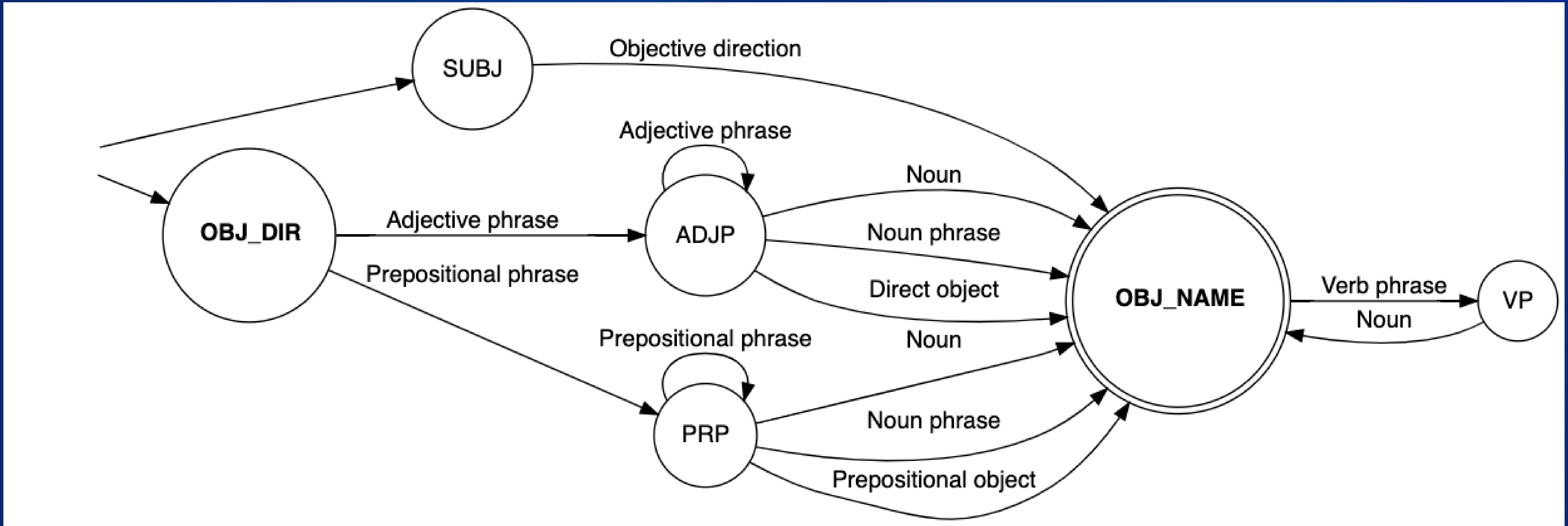
### Contextual Features

**Left and right contextual** information around each token with an appropriate window size, providing additional clues about entity types based on surrounding text.

### Automaton Features

**Regular automaton** designed to capture complex patterns for objective name extraction, such as "profit to be maximized" or "maximize the total monthly profit".

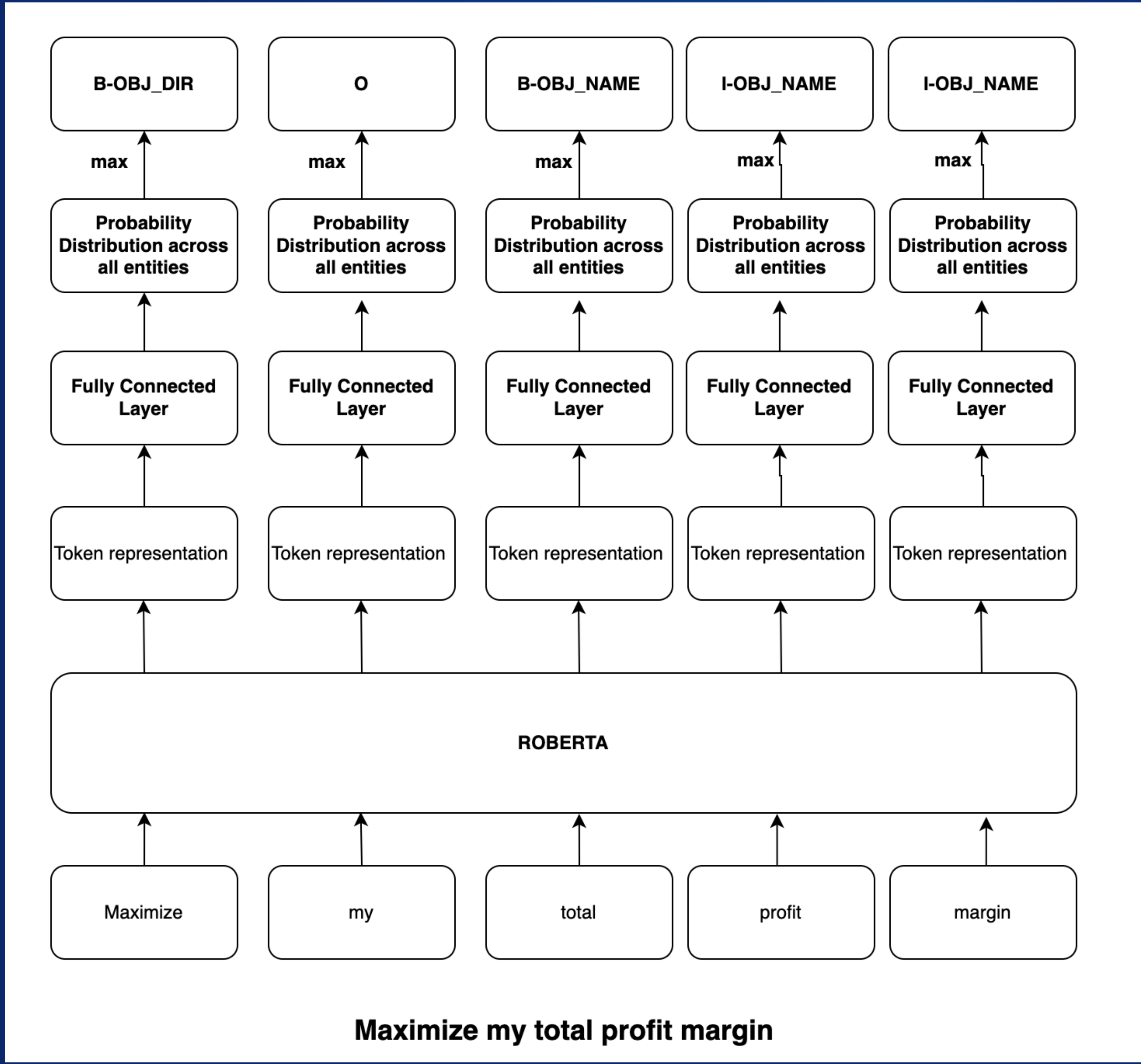
# Classical+: Feature Engineering for Optimization




profit SUBJ to be maximized OBJ_DIR

maximize OBJ_DIR the total monthly ADJP profit NOUN

# Modern NLP Approach




1 **Roberta**

LLM using the same transformer architecture as **BERT but with more robust training**. Employed its large version, which achieves state-of-the-art results on well-known NLP benchmarks.

2 **XLM-RB**

A self-supervised language model following the **RoBERTa architecture** with **multilingual** training. This was the state-of-the-art method on the benchmark dataset.

# Modern+: Training on Optimization Corpora

## Text Extraction

Extracting textual data from PDF versions of **optimization textbooks** to create a domain-specific corpus.

## Masked Language Modeling

**Continued pre-training** via masked language modeling by randomly masking 15% of words and training the model to predict them.

## Token Replacement Strategy

Replacing 80% of masked words with the **MASK token**, 10% with random words, and 10% with the original word to create robust training examples.

## Self-Supervised Training

Training the model in a **self-supervised fashion** to predict the masked words, helping it learn **optimization-specific vocabulary** and patterns.

# Data Augmentation Techniques

## Oversampling Infrequent Patterns

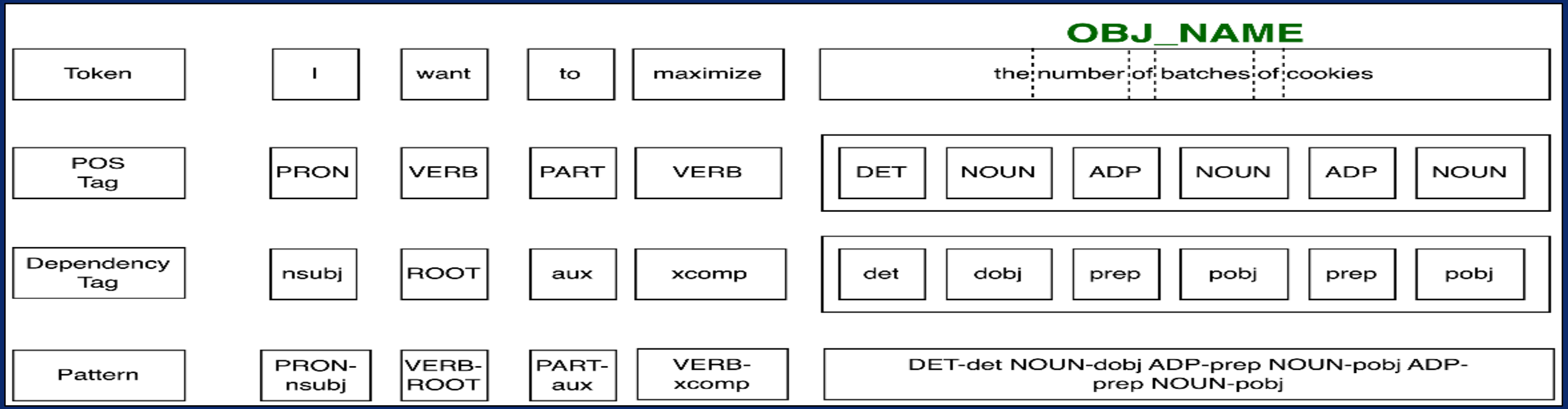
Identify and oversample **infrequent linguistic patterns** without manual inspection.

By extracting **part-of-speech** and **dependency tags** for each token and considering their union as a pattern, identify problem descriptions with rare patterns and duplicate them in the training data.

## L2 Augmentation

To address the challenge of disambiguating objective variables from other variables, introduce L2 augmentation.

Appends the **last two sentences** of problem descriptions to the beginning, helping the model identify the objective earlier in the text and maintain consistent labeling.



# Comparison Methods

## 1 Classical

The **baseline method** based on grammatical and morphological features, establishing a performance lower bound for comparison.

## 2 Classical+

An enhanced classical method incorporating **hand-crafted gazetteer**, syntactic, and contextual features to improve performance.

## 3 RoBERTa

**Transformer model** with strong performance across various language tasks, included for comparison. We use its large model variant.

## 4 XLM-RB

The previous state-of-the-art method on the dataset, based on the **XLM-RoBERTa transformer** architecture. We also evaluated its large variant, XLM-RL.

## 5 XLM-RB+

Our approach to **fine-tune XLM-RB** with optimization textbooks, creating a domain-specific language model.

## 6 Hybrid

Hybrid approach combining **Classical+ with XLM-RB+** and data augmentation techniques for optimal performance.

# Hybrid Modeling Approach

1

### Classical+ CRF

Provides the foundation with grammatical, morphological, gazetteer, syntactic, and contextual features.

2

### Transformer Predictions

Adds features from transformer models fine-tuned on optimization corpora to capture complex patterns.

3

### Data Augmentation

Incorporates oversampling of infrequent patterns and L2 augmentation to address specific challenges.

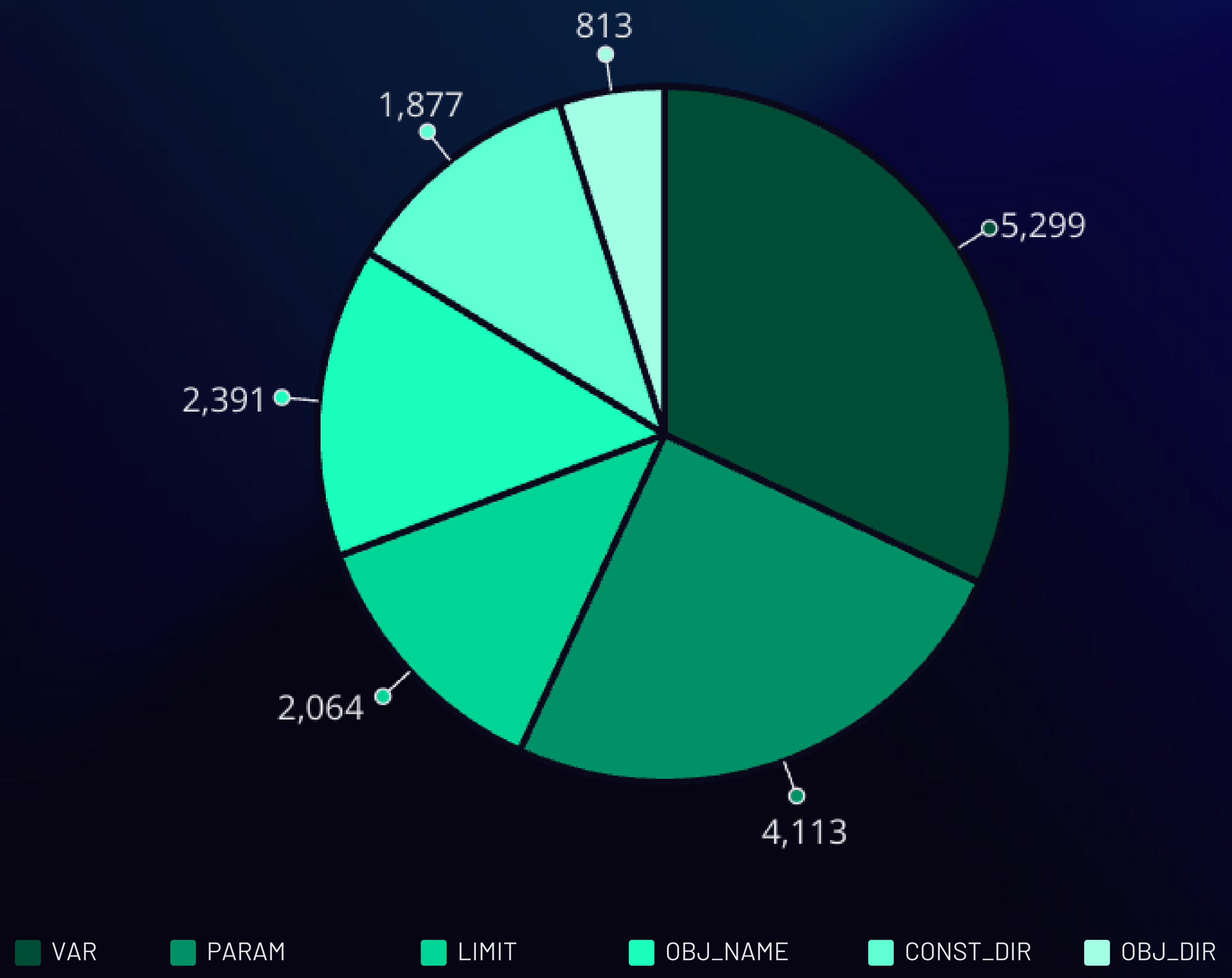
4

### Unified Model

Combines all components into a single system that outperforms individual approaches.

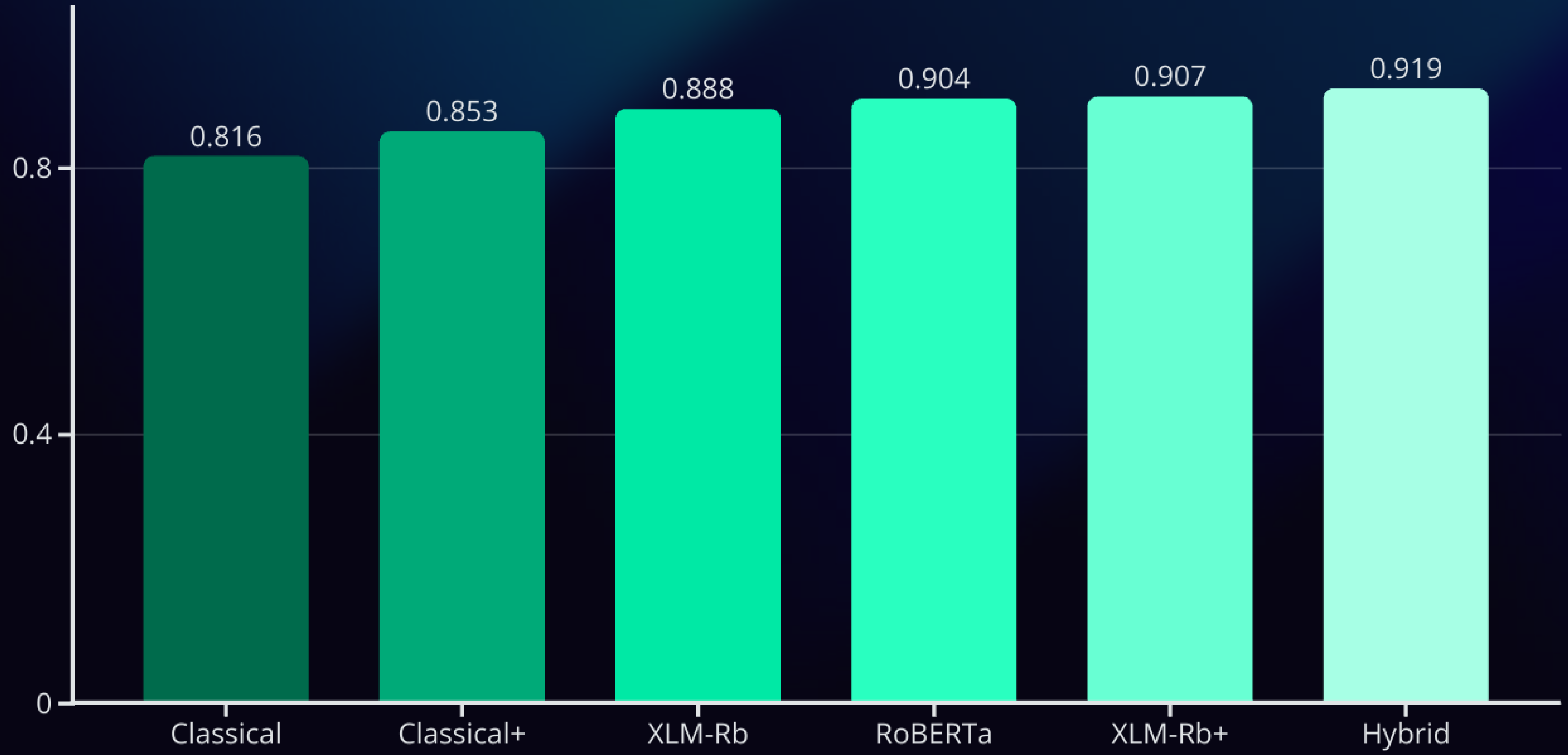
# Experimental Setup

- Experiments on a benchmark dataset of **linear programming word problems**.
- This dataset contains **1,101 samples annotated** with six entity types: variable (VAR), parameter (PARAM), limit (LIMIT), constraint direction (CONST_DIR), objective direction (OBJ_DIR), and objective name (OBJ_NAME).
- The problems in the dataset span **six domains** grouped into **source domains**: advertising, investment, sales **target domains**: production, science, transportation.
- Training set consists samples **only from source domains**, while development and test sets include samples from both source and target domains in a 1:3 ratio.
- Variables (VAR) are the most common entity type, followed by parameters (PARAM) and objective names (OBJ_NAME). Objective direction (OBJ_DIR) is the least frequent entity type.

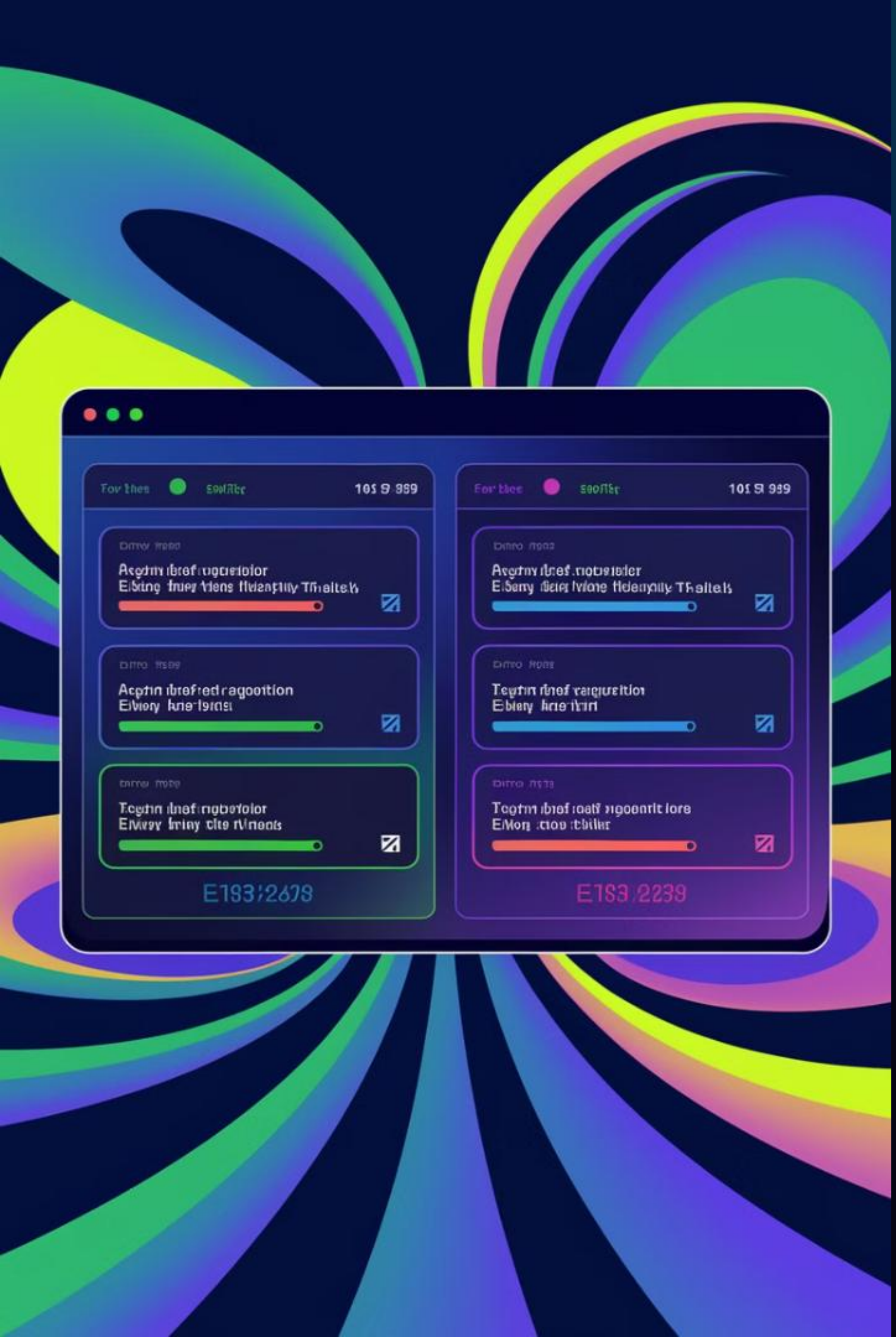



Ramamonjison et. al., NL4Opt Competition: Formulating Optimization Problems Based on Natural Language Descriptions

# Experimental Results




The **Hybrid Approach** combining classical feature engineering with optimization-fine-tuned language models achieves the best performance with a micro-averaged F1 score of **0.919**. This represents a significant improvement over the baseline classical approach (0.816) and the previous state-of-the-art (0.888). The **most challenging entity** to identify is the objective name (OBJ_NAME), where the hybrid approach shows the largest improvement over other methods.



# Comparison with Large Language Models (GPT-4)

1

## Zero-Shot GPT-4

Direct application of GPT-4 without examples achieves only **0.546** F1 score, struggling with entity boundaries and disambiguation.

2

## Few-Shot Learning

Adding examples improves performance significantly, with five examples reaching **0.838** F1 score, demonstrating the importance of in-context learning.

3

## Hybrid Approach

**Our dedicated Ner4Opt** hybrid solution (**0.919** F1) still outperforms even few-shot GPT-4, highlighting the value of specialized approaches for optimization tasks.

# Ner4Opt for Modeling Assistants

**44.44%**

**Without Annotations**

GPT-4 with problem description only
MiniZinc model generation

**65.66%**

**With Ner4Opt Annotations**

GPT-4 with problem description + Ner4Opt
MiniZinc model generation

# Ner4Opt Open-Source Library

## Library Features

**Simple API** for extracting optimization entities from text, with options to select different model types and confidence thresholds.

## OBIE Output Format

**Returns** a list of dictionaries, each containing entity information including start/end indices, text, entity type, and confidence score.

## Pre-trained Resources

Source code, training protocols, and **pre-trained models** are all publicly available through GitHub and Hugging Face.

**pip install ner4opt**

**https://huggingface.co/spaces/skadio/ner4opt**

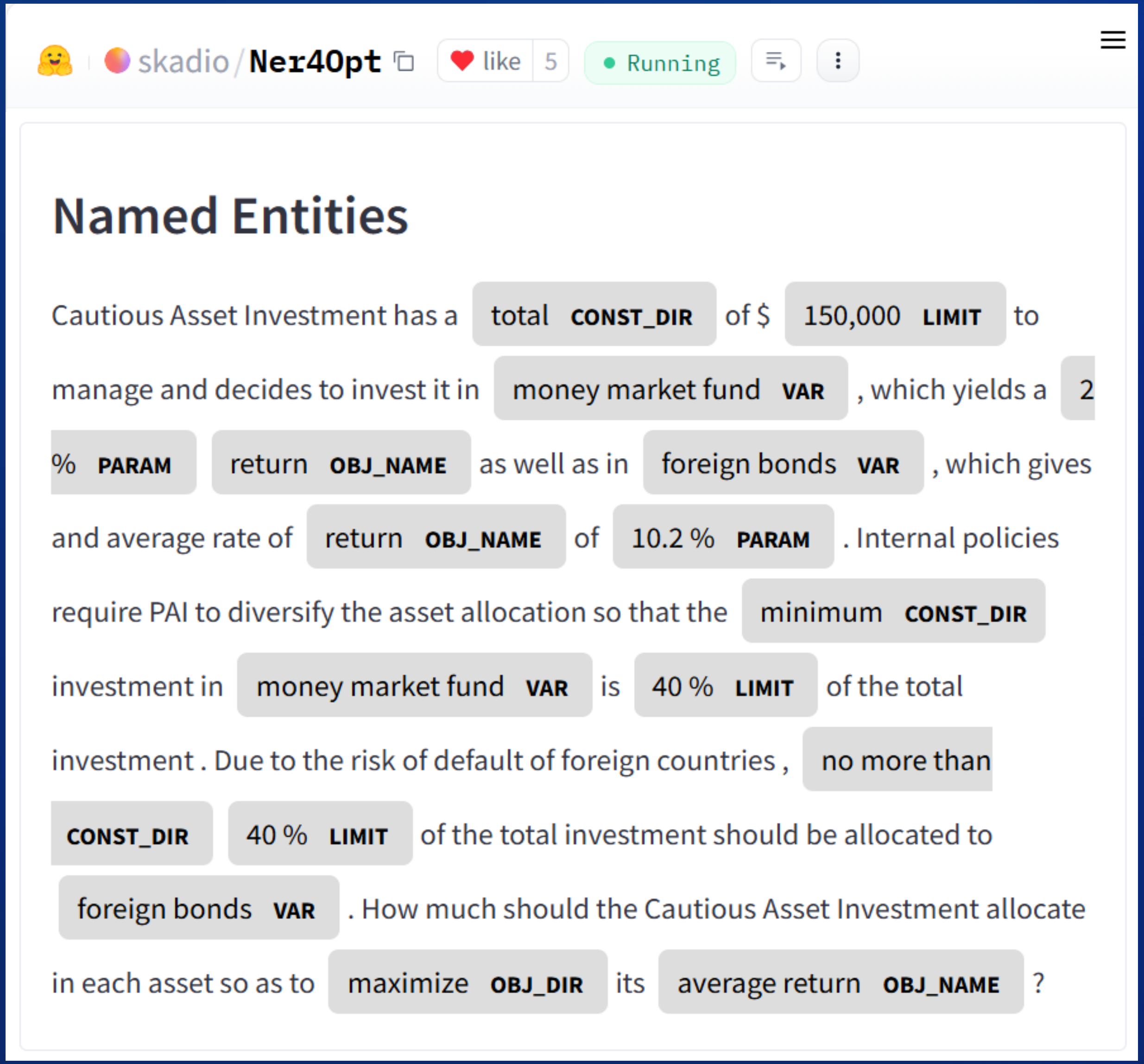
# Interactive Demo

## User Interface

Demo interface with a model selection panel, **a text input field for problem description** and an output panel that highlights identified entities with color-coded tags.

## Model Selection

Users can choose between **different model** types (lexical, lexical-plus, semantic, hybrid) to compare performance on the same input text.




**Hugging Face Spaces Demo**

**https://huggingface.co/spaces/skadio/ner4opt**

# ChatOpt Integration

## Integration

Designed for easy integration with optimization solvers and modeling frameworks, already used in the **Holy Grail 2.0** constraint programming modeling assistant.


- *Holy Grail 2.0: From Natural Language to Constraint Models, CP 2023, D. Tsouros, H. Verhaeghe, S. Kadıoğlu, T. Guns*
- *Constraint modelling with LLMs using in-context learning, CP 2024, K. Michailidis, D. Tsouros, T. Guns*

# ChatOpt Integration

**Integra**

Designed
modeling
constraint

You are designing an office space with two types of desks: long desks and short desks. You can spend at most $2000. Long desks cost $300, take up 10 square feet of space, and seat 6 employees. Short desks cost $100, take up 4 square feet of space, and seat 2 employees. The office can have at most 200 square feet of desks. How many of each desk should you buy in order to maximize the seating availability?

U

B

Variables:

- Number of long desks: Long
- Number of short desks: Short

Objective: Maximize seating availability: 6 * Long + 2 * Short

Constraints:

- Total cost should not exceed $2000: 300 * Long + 100 * Short <= 2000
- Total space taken by desks should not exceed 200 square feet: 10 * Long + 4 * Short <= 200

Pseudo Model

Generate An Executable CPMpy Model

- *Holy Grail 2.0: From Natural Language to Constraint Models, CP 2023, D. Tsouros, H. Verhaeghe, S. Kadıoğlu, T. Guns*
- *Constraint modelling with LLMs using in-context learning, CP 2024, K. Michailidis, D. Tsouros, T. Guns*

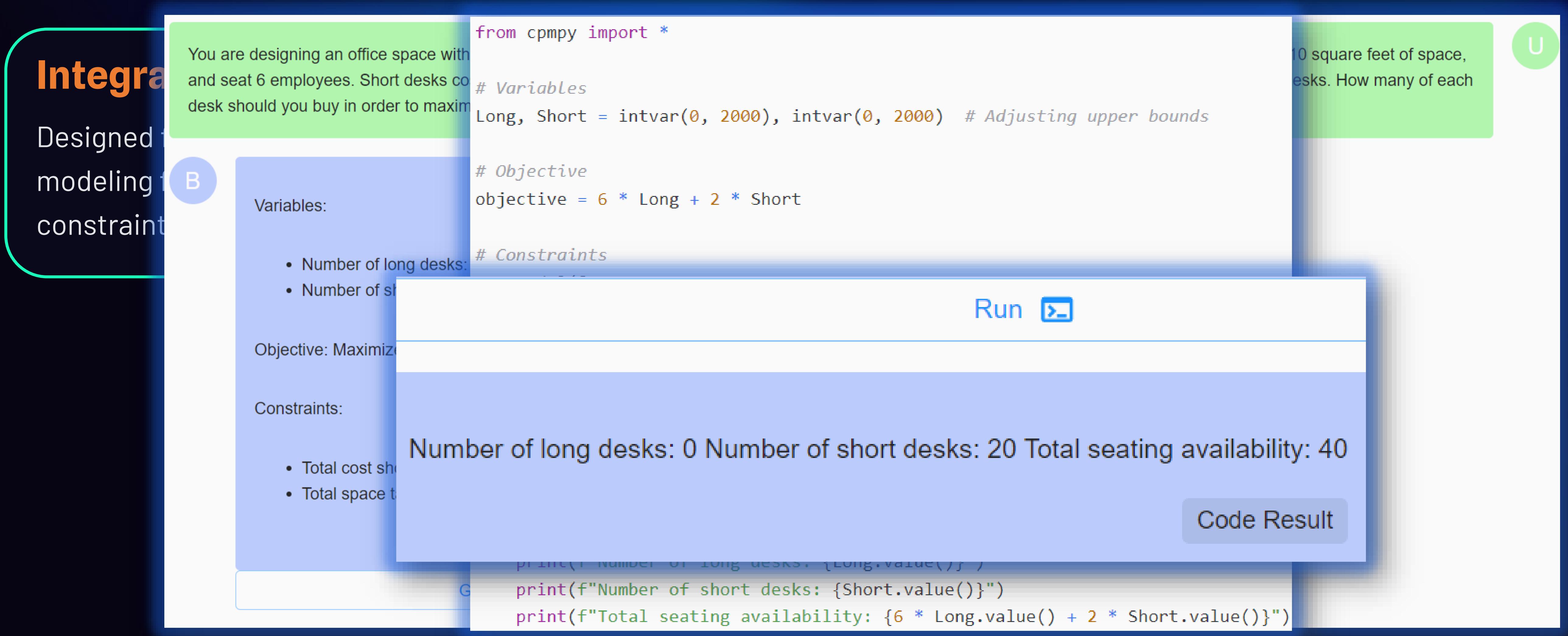
# ChatOpt Integration

**Integra**

Designed f
modeling f
constraint


- *Holy Grail 2.0: From Natural Language to Constraint Models, CP 2023, D. Tsouros, H. Verhaeghe, S. Kadıoğlu, T. Guns*
- *Constraint modelling with LLMs using in-context learning, CP 2024, K. Michailidis, D. Tsouros, T. Guns*

# ChatOpt Integration




- *Holy Grail 2.0: From Natural Language to Constraint Models, CP 2023, D. Tsouros, H. Verhaeghe, S. Kadıoğlu, T. Guns*
- *Constraint modelling with LLMs using in-context learning, CP 2024, K. Michailidis, D. Tsouros, T. Guns*

# Future Directions and Applications

### Integration with Solvers

Embedding Ner4Opt directly into optimization platforms to enable natural language interfaces for model creation.

### Interactive Modeling

Developing conversational interfaces that use Ner4Opt to clarify ambiguities and refine optimization models through dialogue.


### Domain Adaptation

Extending the approach to specialized fields like supply chain, finance, and healthcare with domain-specific entity types.

### Text2Zinc Dataset & Leaderboard

A unified dataset curated to work with LLMs and an associated leaderboard to evaluate strategies to generate MiniZinc models from natural language text.

**Ner4Opt**

**pip install ner4opt**

**Text2Zinc**

**https://huggingface.co/datasets/skadio/text2zinc**

# Future Directions and Applications

Constraints (2024) 29:261–299
https://doi.org/10.1007/s10601-024-09376-5

Check for updates

## NER4OPT: named entity recognition for optimization modelling from natural language

Serdar Kadıoğlu[1,2] · Parag Pravin Dakle[1] · Karthik Uppuluri[1] · Regina Politi[2] · Preethi Raghavan[1] · SaiKrishna Rallabandi[1] · Ravisutha Srinivasamurthy

optimiz
langua

Deve
that u
and ref

alized
and
entity

oard

with
oard to
MiniZinc
xt.

# Our Contributions

## Ner4Opt

A principled approach to **extracting components of optimization models** such as the objective, variables, and constraints from free-form natural language text.

## Text2Zinc

A **unified cross-domain dataset** curated to work with **LLM co-pilots** and an associated **leaderboard** to evaluate strategies to generate **MiniZinc models** from free-from natural language text.

# Text2Zinc: Motivation

## Driving Progress

Datasets and benchmarks **fuel progress** in various domains: Computer Vision, NLP, and SAT, CP, MIP, RecSys, etc.

## Room for Improvement

Current problem datasets have **potential for improvement** for integration with language models.

## Structured Information & Metadata

Models and natural language descriptions of problems have been documented heavily but seldom occur together. Crucial **metadata is unavailable**.

# Existing Resources

### NL4OPT

- Linear programming problems
- No separation between problem description and data
- Relatively easy instances

### NLP4LP

- Extends NL4OPT
- Introduces MIPs
- Evaluated with GurobiPy and cvxpy

### ComplexOR

- Standard OR Problems
- Evaluated with GurobiPy

Optimization

### Logic Grid Puzzles

- Introduces satisfaction problems in the form of logic grid puzzles

### CSPLib

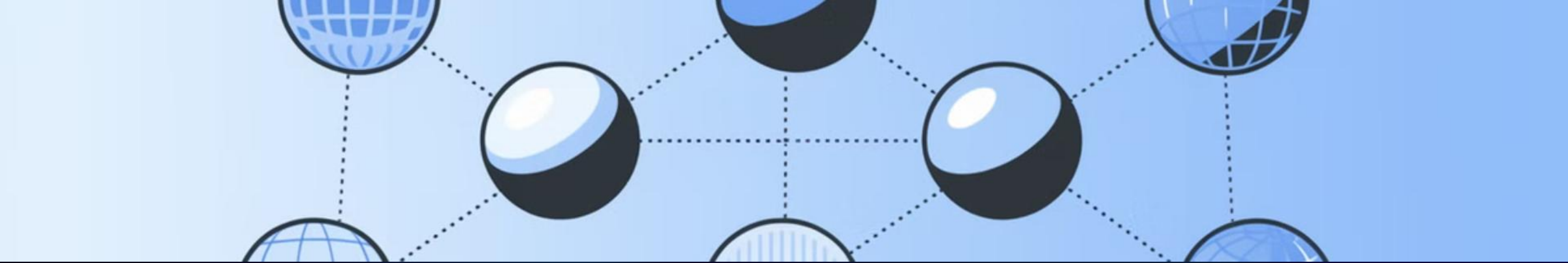
- CP and Satisfaction problems
- Not designed to work with ML or LLMs

### Hakank's Models

- Extensive set of constraint models in various languages
- Does not capture metadata

Satisfaction

** Massive thank you to the community for contributing these valuable resources!*



# Text2Zinc: Addressing Dataset Gaps

## 1

### Cross-Domain

- Focus on combining both **optimization & satisfaction** problems.
- Incorporates LP, MIP, CP problems.

## 2

### Unified Format

- Unifies existing datasets.
- **Clear separation** of problem description & instance data.

## 3

### Solver Agnostic

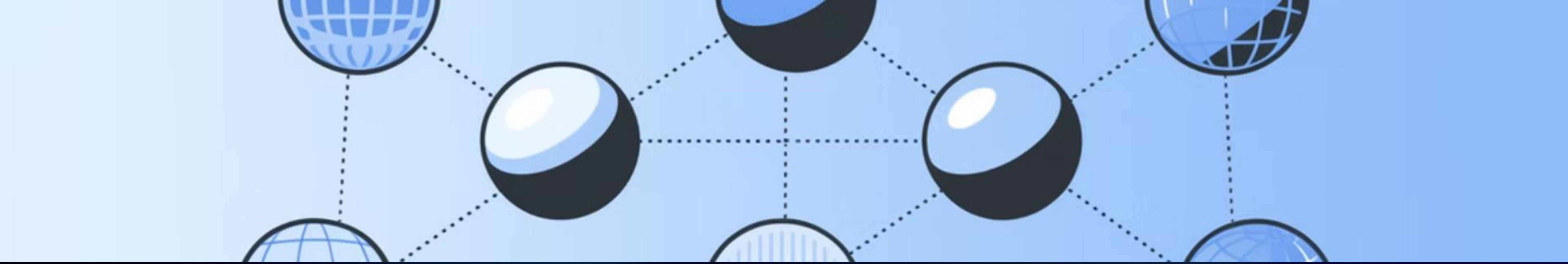
- Enables **solver agnostic** approaches.
- MIP, CP, SAT, LCG through MiniZinc.

## 4

### Data Augmentation

- Clear and concise descriptions.
- Input and output specification.
- **Metadata** generation.
- Manual verification.

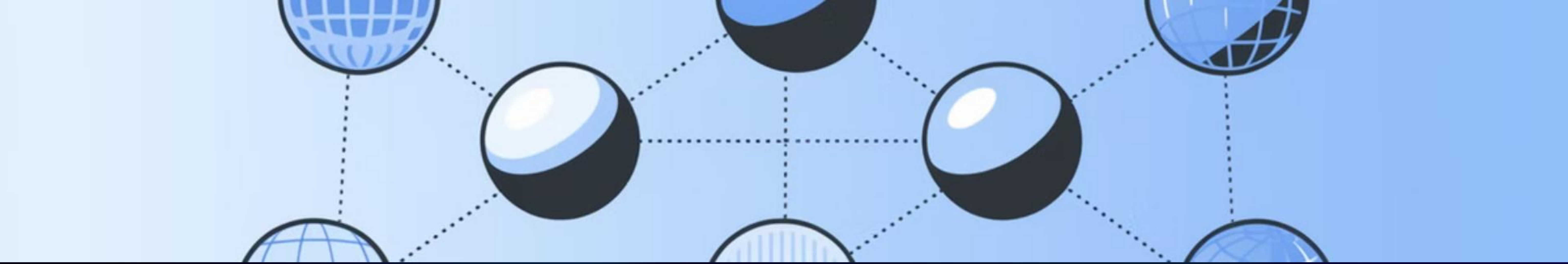
S. Kadıoğlu et. al. Text2Zinc: A Cross-Domain Dataset for Modeling Optimization and Satisfaction Problems in MiniZinc



# Text2Zinc: Example Timetabling Problem – Description


```
  "description": "Lecture timings need to be scheduled for courses across a limited number of periods. Each
     course requires a specific number of lectures and can only be assigned to certain periods due to
     availability constraints. Some courses have conflicts due to having common students and cannot be scheduled
     at the same time. Additionally, there is a limited number of rooms that can be used and thus a maximum
     number of lectures that can occur simultaneously. How can we allocate lectures to periods while ensuring all
      constraints are met?",
  "identifier": "or_lp_ip_scheduling_problem_2",
  "metadata": {
   "name": "Timetable Problem", "domain": "Scheduling", "objective": "satisfy", "source": "hakank", "constraints
       ": [
     "forall", "<=", "+", "=", "sum"]
  }
}
```

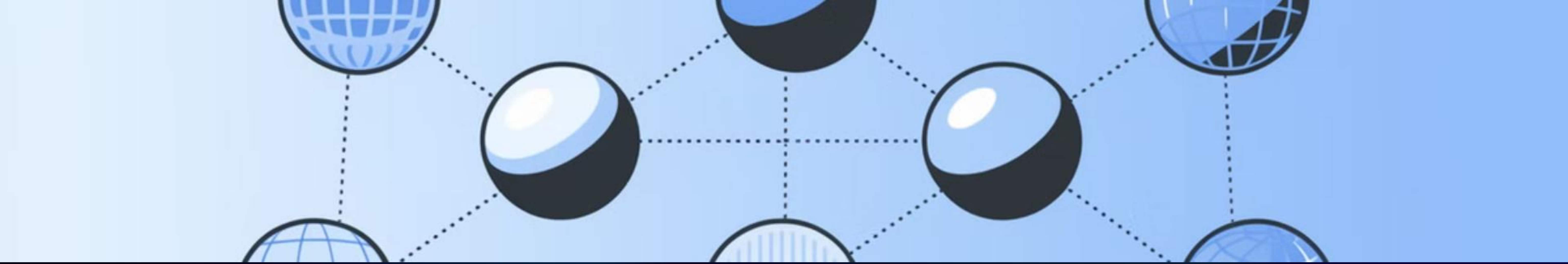

**Figure 2** **An example input with description, parameters, metadata, and output fields.**



# Text2Zinc: Example Timetabling Problem – Model

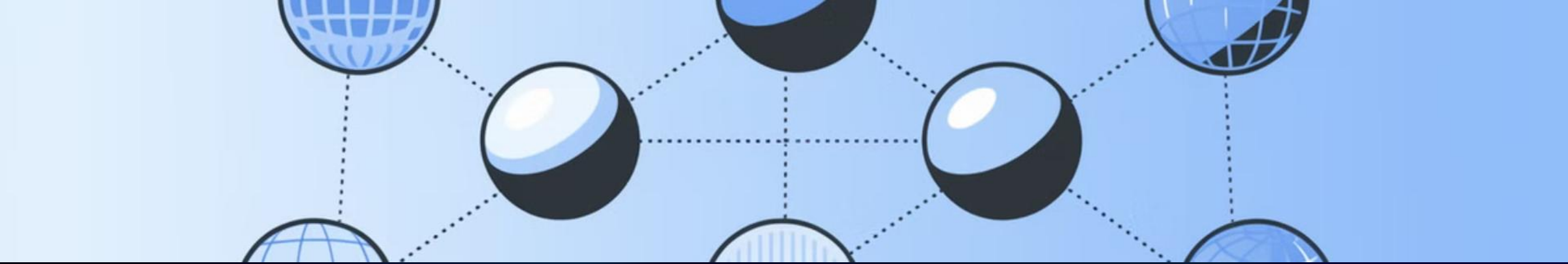
model.mzn

```
include "globals.mzn";

% Input parameters
int: courses;
int: periods;
int: rooms;

array[1..courses, 1..periods] of int: available;
array[1..courses, 1..courses] of int: conflict;
array[1..courses] of int: requirement;

% Decision variables
array[1..courses, 1..periods] of var 0..1: timetable;
```



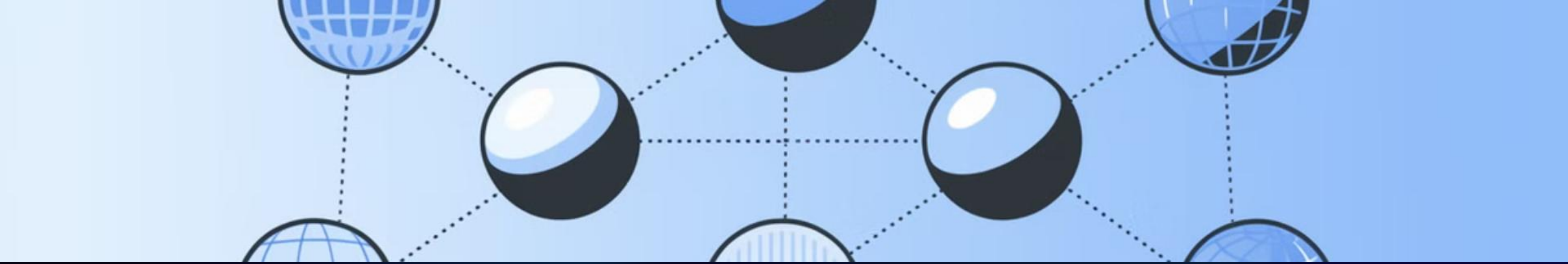
# Text2Zinc: Example Timetabling Problem – Model

```
constraint
   % 1. Conflicts: Courses with common students must not be scheduled at the same time
   forall(c1, c2 in 1..courses where c1 < c2) (
      if conflict[c1, c2] = 1 then
         forall(p in 1..periods) (
            timetable[c1, p] + timetable[c2, p] <= 1
         )
      else
         true
      endif
   )
   % 2. Availabilities: Courses can only be scheduled in available periods
   /\
   forall(c in 1..courses, p in 1..periods) (
      if available[c, p] = 0 then
         timetable[c, p] = 0
```



# Text2Zinc: Example Timetabling Problem – Model

```
% 3. Rooms: At most 'rooms' lectures can be scheduled per period
/\
forall(p in 1..periods) (
   sum([timetable[c, p] | c in 1..courses]) <= rooms
)
% 4. Number of lectures per course must match the requirement
/\
forall(c in 1..courses) (
   sum([timetable[c, p] | p in 1..periods]) = requirement[c]
);
```



# Text2Zinc: Example Timetabling Problem – Input & Output

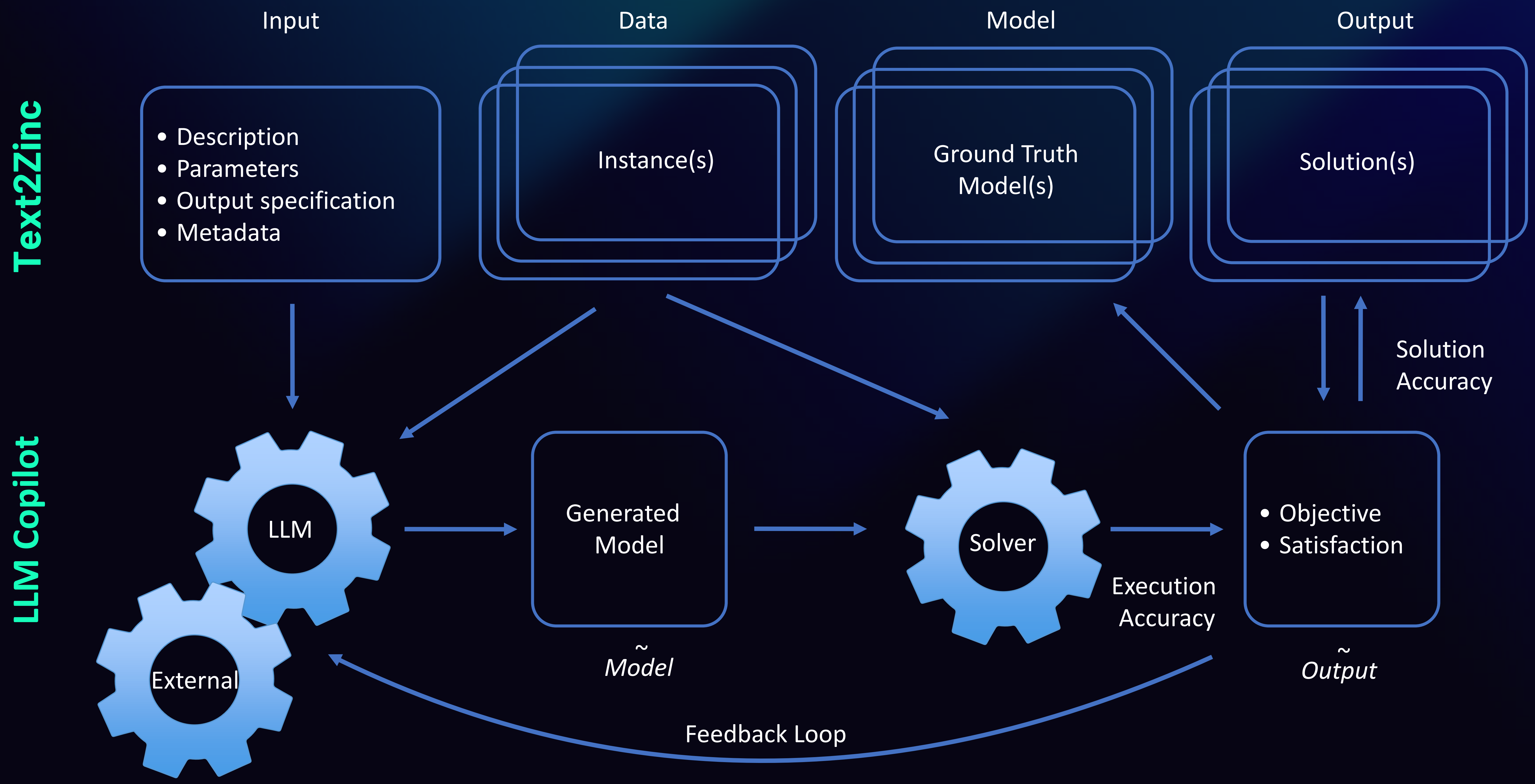
data.dzn

```
int: courses = 5;
int: periods = 20;
int: rooms = 2;
array[1..courses, 1..periods] of int:
     available = array2d(1..courses,
    1..periods, [
    % 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6
       7 8 9 0
      0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1,
         1, 0, 1, 1, 0, 1, 1, 1, 1,
      1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1,
         1, 1, 1, 1, 1, 1, 1, 1, 1,
      0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1,
         1, 0, 1, 1, 1, 1, 0, 1, 1,
      1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1,
         1, 1, 1, 1, 1, 1, 0, 1, 1,
      1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
         1, 1, 1, 1, 1, 1, 1, 1, 1
```

output.json


```
{
"timetable": [
    [0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0,
        1, 0, 1, 0, 0, 0, 0, 0, 0],
    [1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0,
        0, 1, 0, 1, 1, 1, 1, 1, 1],
    [0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1,
        1, 0, 1, 1, 1, 1, 0, 1, 1],
    [0, 0, 1, 0, 0, 0, 1, 1, 1, 1, 1,
        0, 0, 0, 0, 0, 0, 0, 0, 0],
    [1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 1, 0, 0, 0, 0, 1, 0, 0]
  ]
  }
```

# Text2Zinc: A Unified Approach



# Text2Zinc Statistics

**64**

### Linear Programming

Continuous variables
with linear constraints

**31**

### Mixed Integer Programming

Continuous
and discrete variables

**15**

### Constraint Programming

Global
constraints and logic

Our dataset includes **550+ instances** of mixed of **LP, MIPs, and CP problems** across various domains
Providing a comprehensive benchmark for natural language to constraint model translation.

# Text2Zinc Initial Co-Pilot Approaches

### Out-of-the-box LLM

Vanilla prompting, zero-shot, few-shot performance

Single vs. Multi-Call


### Chain-of-Thought

Improved reasoning through step-by-step problem-solving

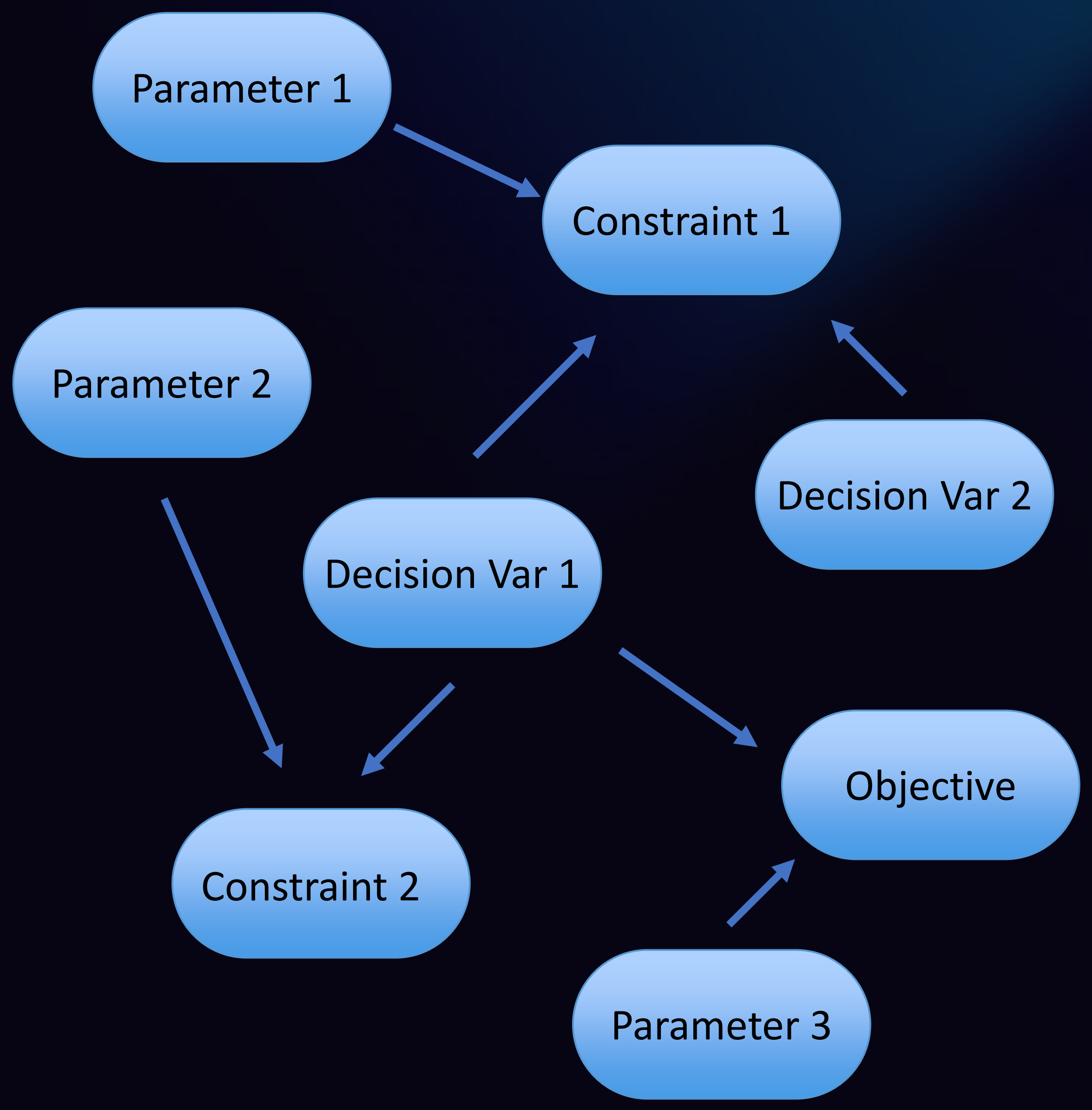
### Knowledge Graph

Leveraging structured knowledge as intermediary representation

### Structured Prediction

Grammar-based model generation to enforce LLM output

# Knowledge Graphs Representation




### Knowledge Graph Integration

LLM constructs knowledge graph which is used to solve problem.

### Intermediary Representation

Condenses important information about entities into human and machine readable format.

### Structured Representation

Incorporates referential information about parameters, decision variables, constraints, and objective.

# Text2Zinc Initial Results (GPT-4)

| Solution Approach | Execution Accuracy | Solution Accuracy | LLM Calls |
|---|---|---|---|
| Baseline | 0.33 | 0.17 | 1 |
| Chain-of-Thought (CoT) | 0.57 | 0.28 | 1 |
| Knowledge Graph | 0.48 | 0.26 | 2 |
| CoT + Code Validation | 0.57 | **0.28** | 2 |
| CoT + Grammar Validation | 0.63 | 0.23 | 3 |
| CoT + Code & Grammar Val. | **0.70** | 0.25 | 3 |
| Compositional | 0.44 | 0.20 | 4 |
| Compositional + Code Val. | 0.44 | 0.21 | 5 |

**Hugging Face Text2Zinc Leaderboard**

# Text2Zinc Initial Results (GPT Reasoning)

| Solution Approach | Execution Accuracy | Solution Accuracy | LLM Calls |
|---|---|---|---|
| Baseline | 0.33 | 0.17 | 1 |
| Chain-of-Thought (CoT) | 0.57 – **0.61** | 0.28 – **0.35** | 1 |
| Knowledge Graph | 0.48 | 0.26 | 2 |
| CoT + Code Validation | 0.57 – **0.81** | **0.28** – **0.41** | 2 |
| CoT + Grammar Validation | 0.63 | 0.23 | 3 |
| CoT + Code & Grammar Val. | **0.70** – **0.74** | 0.25 – **0.40** | 3 |
| Compositional | 0.44 | 0.20 | 4 |
| Compositional + Code Val. | 0.44 | 0.21 | 5 |

[Hugging Face Text2Zinc Leaderboard]

# General Observations

### 1 Execution-Solution Gap

Consistently lower solution accuracies across strategies indicate **complexity of optimization** expertise

### 2 Complication Issues

Syntax errors are primary cause of execution failures, attributed to LLM's limited exposure to **MiniZinc's specialized syntax**

### 3 Information Sweet Spot

Both too little and too much information can be detrimental, suggesting an optimal **level of context** exists

### 4 Reasoning vs. Structure

Superior **performance of CoT and compositional** approaches suggests how information is processed matters more than quantity provided

# Future Directions

1. **Call-to-Action: Dataset Expansion**

   Encourage community contributions to create more comprehensive resources

2. **Intermediate Representations**

   Explore alternative representations like named entities and semantic graphs

3. **Agentic Frameworks**

   Investigate potential of agentic approaches in capturing nuances of optimization modeling problems

4. **Model Improvements**

   Develop specialized LLMs for optimization and satisfaction tasks

# Future Directions

informs
https://pubsonline.informs.org/journal/ijds

INFORMS JOURNAL ON DATA SCIENCE

Vol. 00, No. 0, Xxxxx 0000, pp. 000–000

ISSN 2694-4022, EISSN 2694-4030

## Text2Zinc for Large-Language Models as Modeling Co-Pilots

Serdar Kadıoğlu
Department of Computer Science, Brown University; AI Center of Excellence, Fidelity Investments, serdark@cs.brown.edu

Akash Singirikonda
Department of Computer Science, Brown University, akash_singirikonda@brown.edu

Karthik Uppuluri
AI Center of Excellence, Fidelity Investments, karthik.uppuluri@fmr.com

Hugging Face Text2Zinc Dataset & Leaderboard

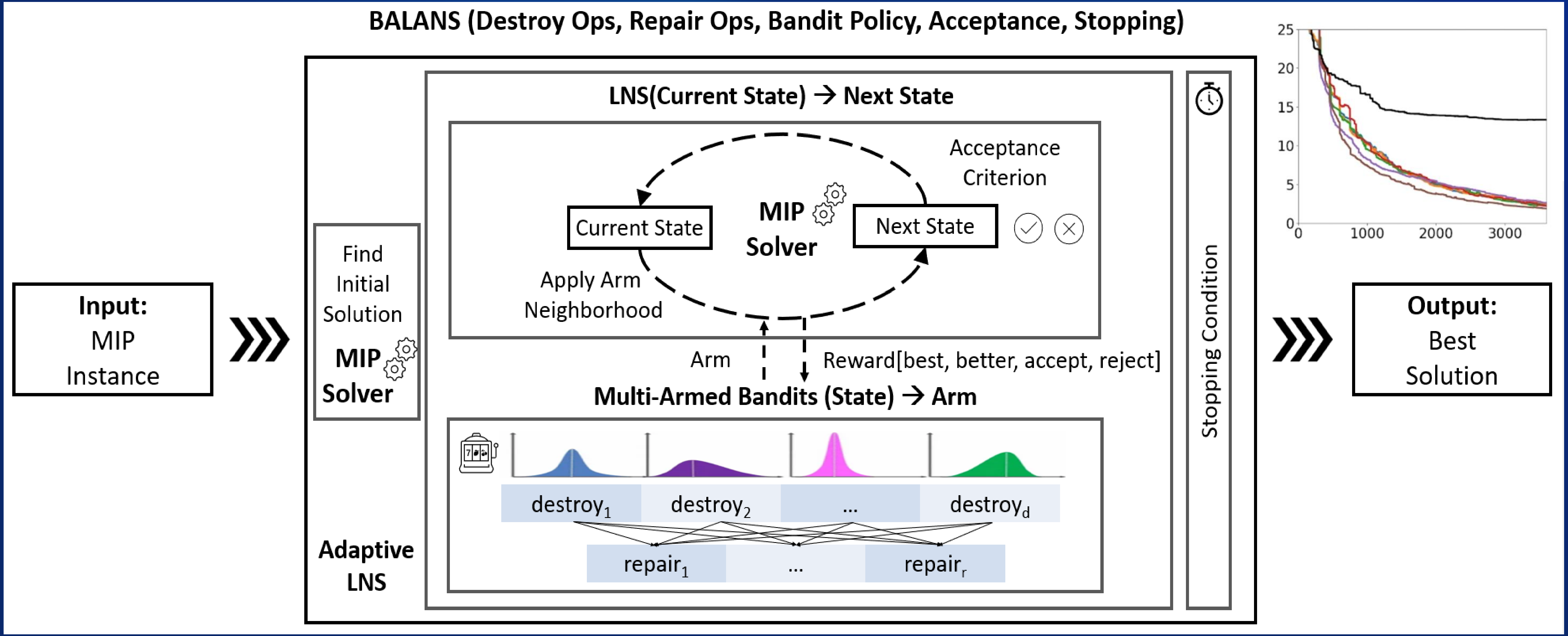
# Emerging Literature

- **Co-Pilot Position Papers:** Holy Grail 2.0 (Tsouros et.al. 2023), Co-Pilot Manifesto (Wasserkrug et al. 2025)
- Learning natural **language interfaces** with neural models, Li Dong, PhD thesis, 2019
- **LGPSolver:** Solving logic grid puzzles automatically, ACL'20
- Automatic formulation and optimization of **linear problems** from a structured paragraph, ICSCT'21
- Early efforts (Ramamonjison et al. 2022) focused on linear programming problems using entity recognition and logical forms
- Synthesizing **MIP models** from NL, Qingyang Li et. al., 2023
- **Latex2Solver** turns input .tex files into optimization models + symbolic model UI, Ramamonjison et.al, ACL'23
- **OptiGuide:** LLMs for Supply Chain Optimization, Microsoft, 2023
- **MeetMate:** Enabling interactive decision support using LLMs and CP, Lawless et al., 2023
- **Logic.py:** Software verification through logic (Kesseli et. al. 2025)
- Towards an Automatic Optimization Model Generator Assisted with GPT, Almonacid, 2023
- **LM4OPT:** Unveiling the Potential of LLMs in Formulating Mathematical Optimization Problems, 2024
- **Agents:** Multi-agent chain-of-experts (Xiao et al. 2023) **Optimus** (AhmadiTeshnizi et al. 2024) specific to Gurobi and cvxpy
- **Data scarcity** for LPs in PuLP addressed by data augmentation, leveraging CodeT5 (Prasath and Karande 2023)
- **OR-Instruct:** Training custom LLMs through solver specific OR-Instruct (Huang et al. 2024a).
- **MAMO benchmark** (Huang et al. 2024b), focusing on LLMs' mathematical modeling processes rather than solution correctness
- **Streamliners:** LLMs for generating streamliners in CP using MiniZinc (Voboril et al. 2024)
- **RAG:** In-context learning and RAG to build CPMPY constraint models (Michailidis et al. 2024)
- **Privacy:** Domain-specific applications, focusing on supply chain optimization while preserving data privacy (Li et al. 2023)
- **Infeasible:** Diagnosing infeasible optimization problems through interactive conversations (Chen et al. 2023)
- **Generation:** optimization prob;ems from scratch (Jiang et al. 2025)
- **MCP:** Model context protocol to integrate LLMs and with symbolic solvers (Szeider 2025)

# Emerging Literature

- **Co-Pilot Position Papers:** Holy Grail 2.0 (Tsouros et.al. 202
- Learning natural **language interfaces** with neural models,
- **LGPSolver:** Solving logic grid puzzles automatically, ACL'2
- Automatic formulation and optimization of **linear problem**
- Early efforts (Ramamonjison et al. 2022) focused on linear
- Synthesizing **MIP models** from NL, Qingyang Li et. al., 2023
- **Latex2Solver** turns input .tex files into optimization mode
- **OptiGuide:** LLMs for Supply Chain Optimization, Microsoft,
- **MeetMate:** Enabling interactive decision support using LL
- **Logic.py:** Software verification through logic (Kesseli et. a
- Towards an Automatic Optimization Model Generator Assis
- **LM4OPT:** Unveiling the Potential of LLMs in Formulating M
- **Agents:** Multi-agent chain-of-experts (Xiao et al. 2023) **Op**
- **Data scarcity** for LPs in PuLP addressed by data augment
- **OR-Instruct:** Training custom LLMs through solver specifi
- **MAMO benchmark** (Huang et al. 2024b), focusing on LLMs
- **Streamliners:** LLMs for generating streamliners in CP usir
- **RAG:** In-context learning and RAG to build CPMPY constrai
- **Privacy:** Domain-specific applications, focusing on supply
- **Infeasible:** Diagnosing infeasible optimization problems th
- **Generation:** optimization prob;ems from scratch (Jiang et
- **MCP:** Model context protocol to integrate LLMs and with sy

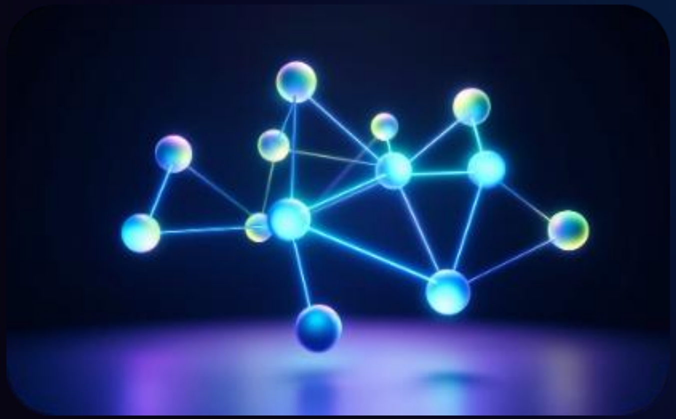
# Balans: Bandit-based ALNS for MIP Solving




**To appear at IJCAI'25 in August, 2025**

# AI Center of Excellence @ Fidelity

The 40th Annual AAAI Conference on Artificial Intelligence
JANUARY 20 – JANUARY 27, 2026 | SINGAPORE



## Optimization & Decision Systems

[Constraints'24, CPAIOR'23, NeurIPS'22] **Text2Zinc** & **Ner4Opt**
LLM copilots for optimization
github.com/skadio/ner4opt

[ArXiv'24] **Balans** – Adaptive meta optimization solver
github.com/skadio/balans

[ArXiv'24] **iCBS**: Pruning LLVMs
Improved conflict-based search
github.com/amazon-science/icbs

## Explainable & Responsible AI

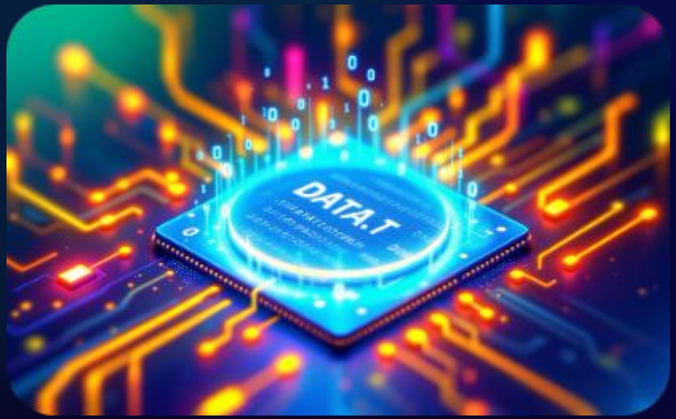
[AAAI'25, MAKE'23] **BoolXAI**
Explainable AI with Boolean formulas
github.com/fidelity/boolxai

[ACM'24, CPAIOR'23, ICMLA'21]
**Jurity** Fairness & bias mitigation
github.com/fidelity/jurity

## Machine Learning & Recommendations

[AAAI'24] **Mab2Rec** - Multi-armed bandit recommender systems
github.com/fidelity/mab2rec

[IJAIT'21] **MABWiser**
Contextual multi-armed bandits
github.com/fidelity/mabwiser



## Text Embeddings & Data Processing at Scale

[AI Magazine'23, AAAI'22] **Seq2Pat**
Sequential pattern mining
github.com/fidelity/seq2pat

[AAAI'21] **TextWiser**
NLP/text featurization
github.com/fidelity/textwiser

[CPAIOR'22] **Selective**
Tabular feature selection
github.com/fidelity/selective

**skadio.github.io**