Bridging Natural Language and Optimization: An Overview of Ner4Opt and Text2Zinc

S. Kadıoğlu[1,2], A. Singirikonda[1], P. Dakle[2], K. Uppuluri[2], R. Politi[2], P. Raghavan[2], S. Rallabandi[2], R. Srinivasamurthy[2]

INFORMS Computing Society (ICS), Toronto, Canada, 2025

Serdar Kadıoğlu

[1] **Dept. of Computer Science, Brown University**

[2] **AI Center of Excellence, Fidelity Investments**

skadio.github.io

BROWN

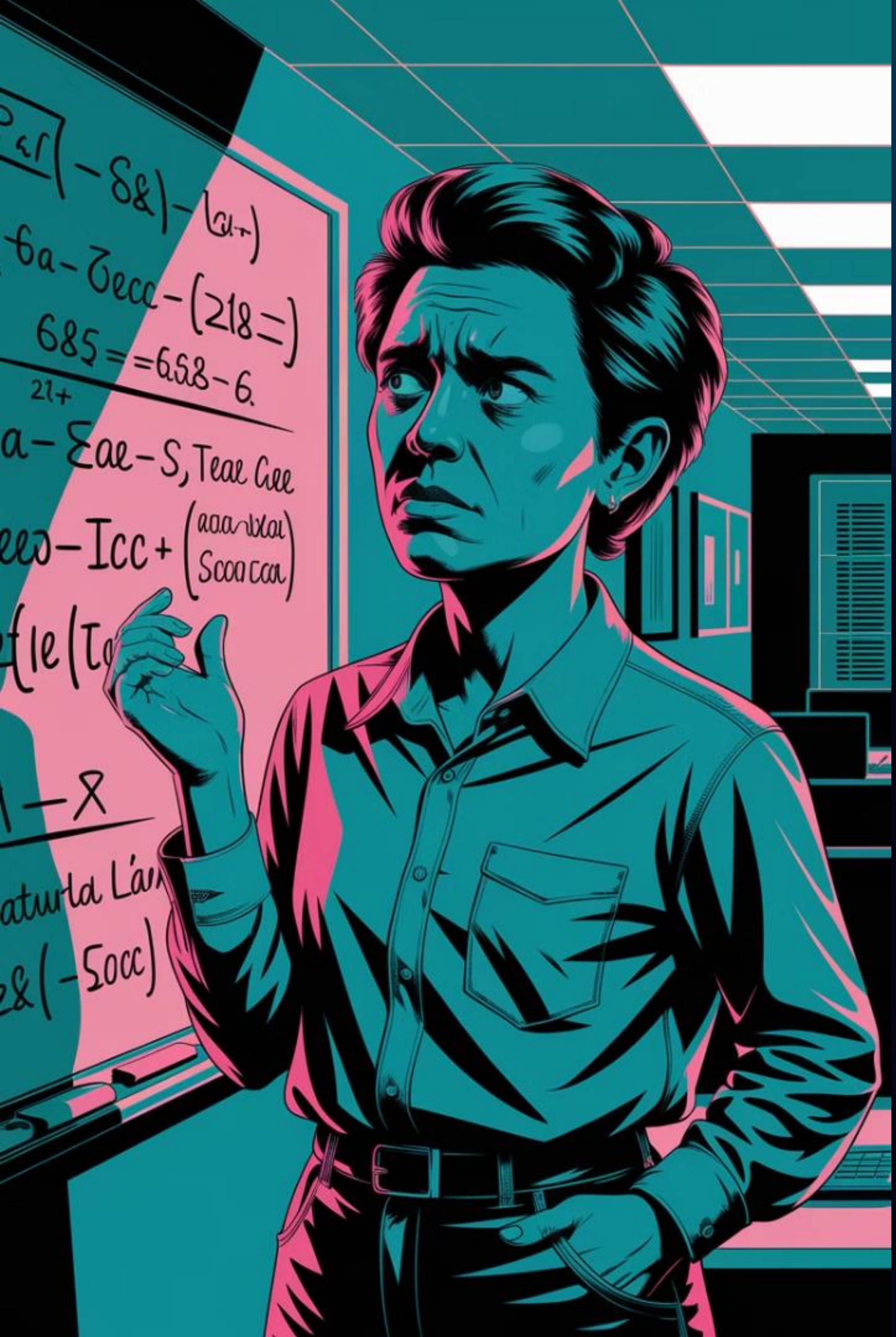
Integration with Optimization Technology

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 Decision Making & Optimization

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

2 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.

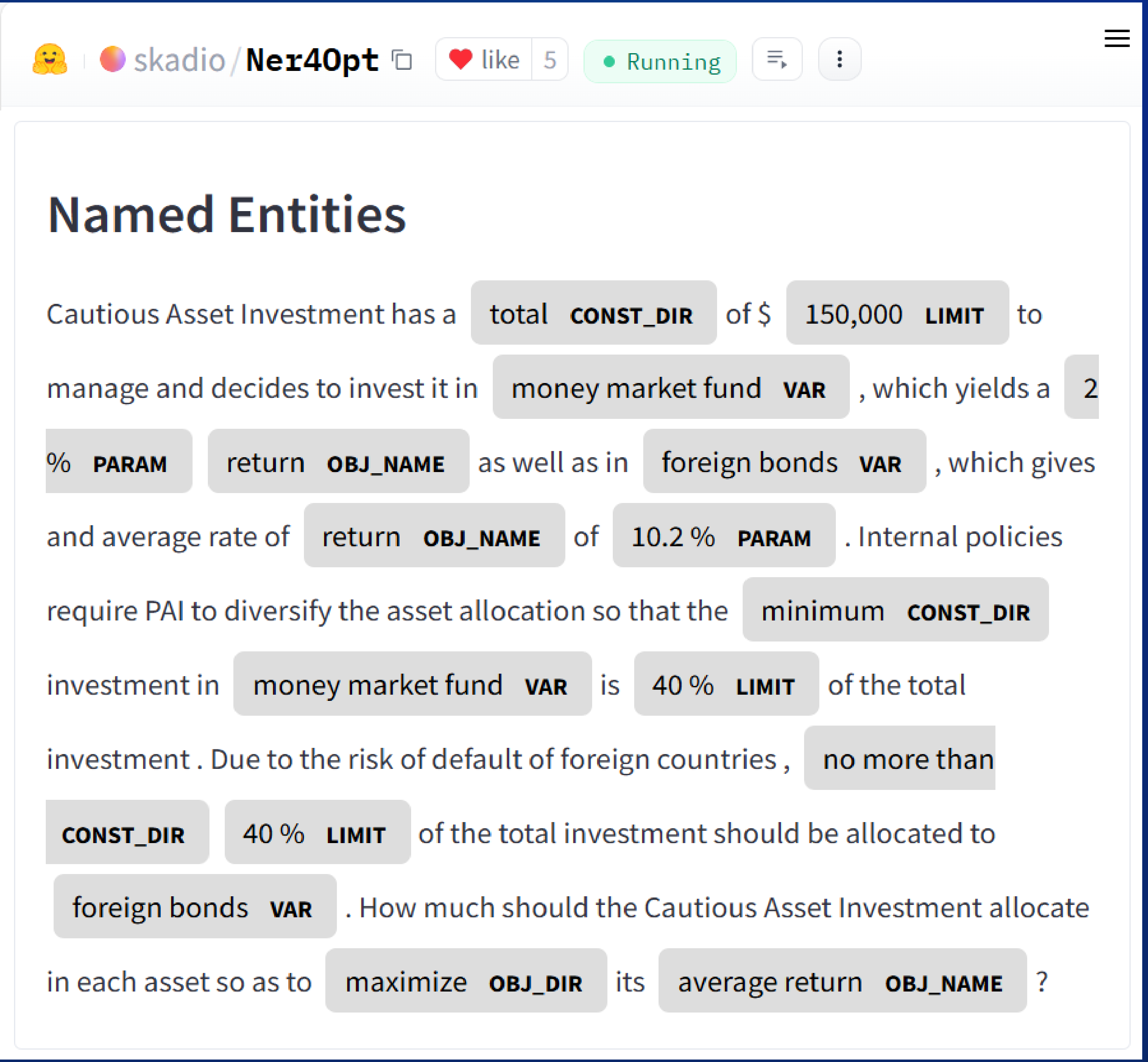
Text2Zinc

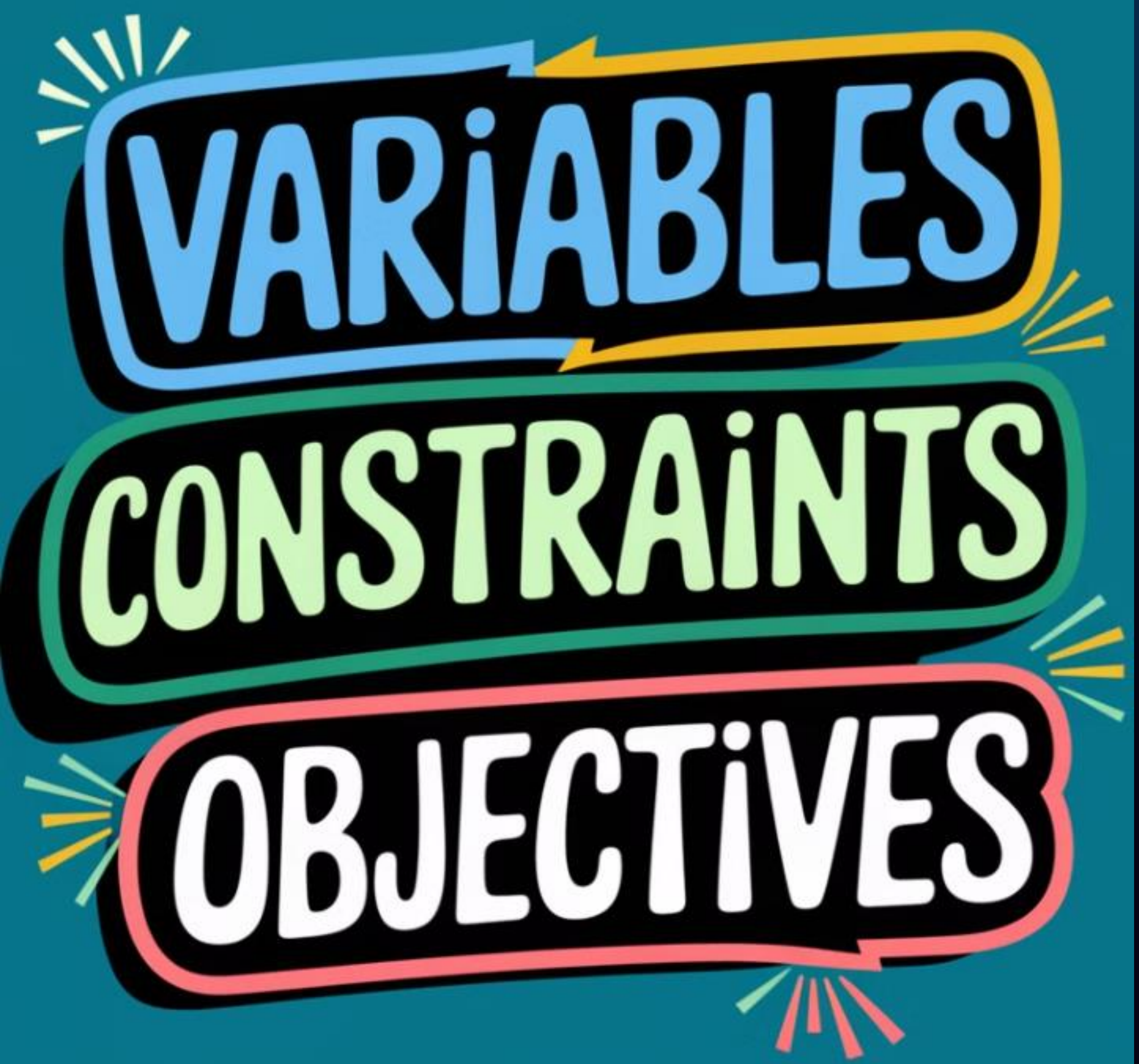
A **unified cross-domain dataset** curated to work with LLMs and an associated **leaderboard** to evaluate strategies to generate **MiniZinc models** from free-from natural language text.

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.





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**.

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

Unique Challenges of Ner4Opt

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.

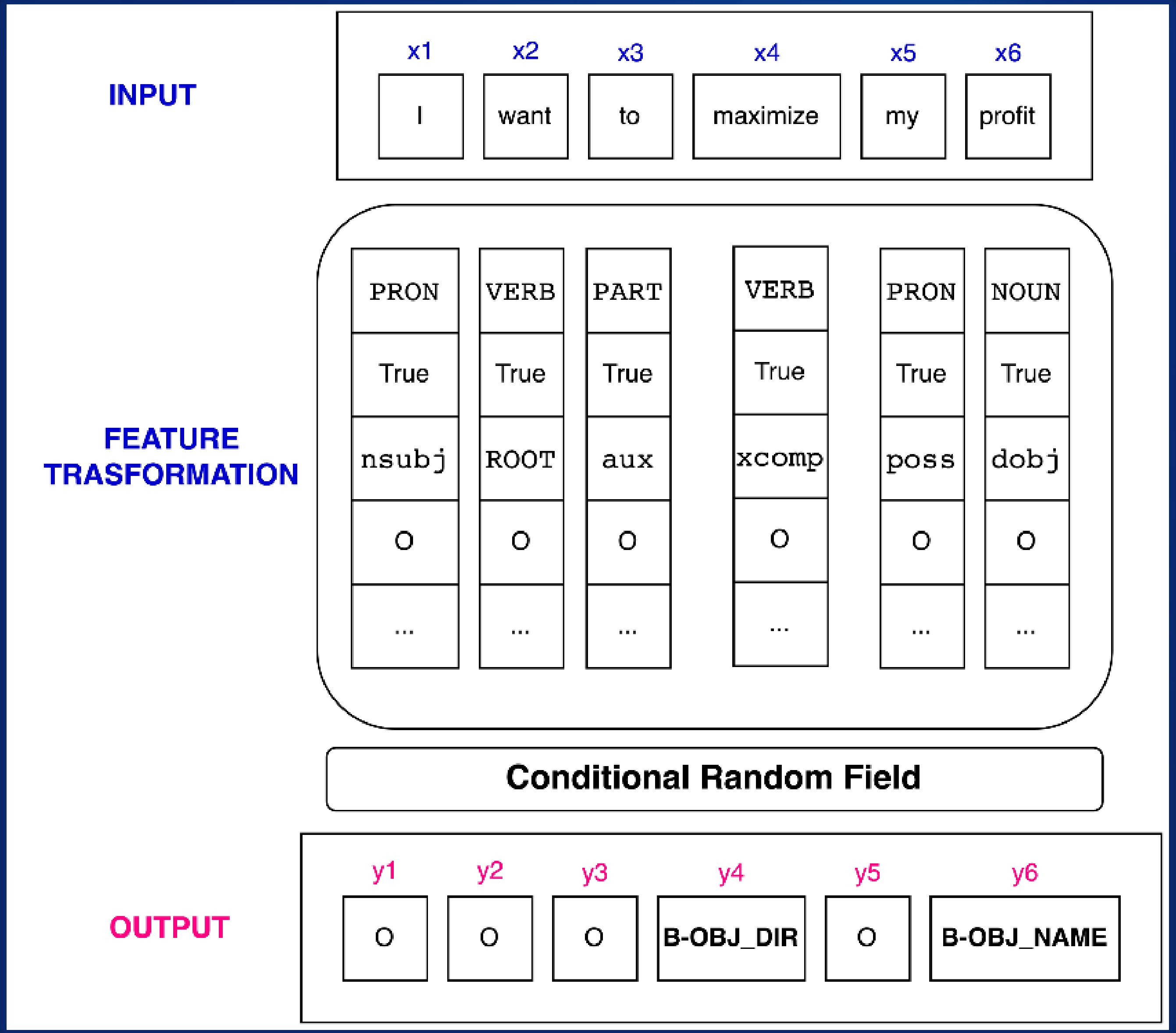
These models are **fine-tuned on optimization corpora** to improve domain-specific understanding.

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.

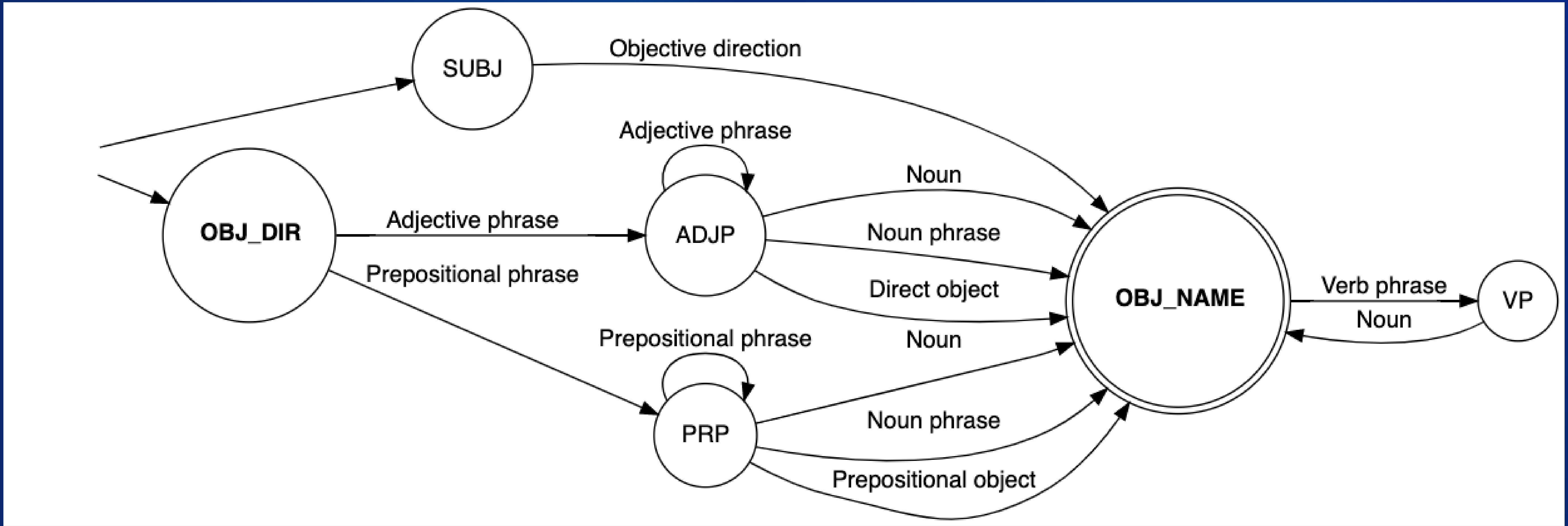
Contextual Features

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".

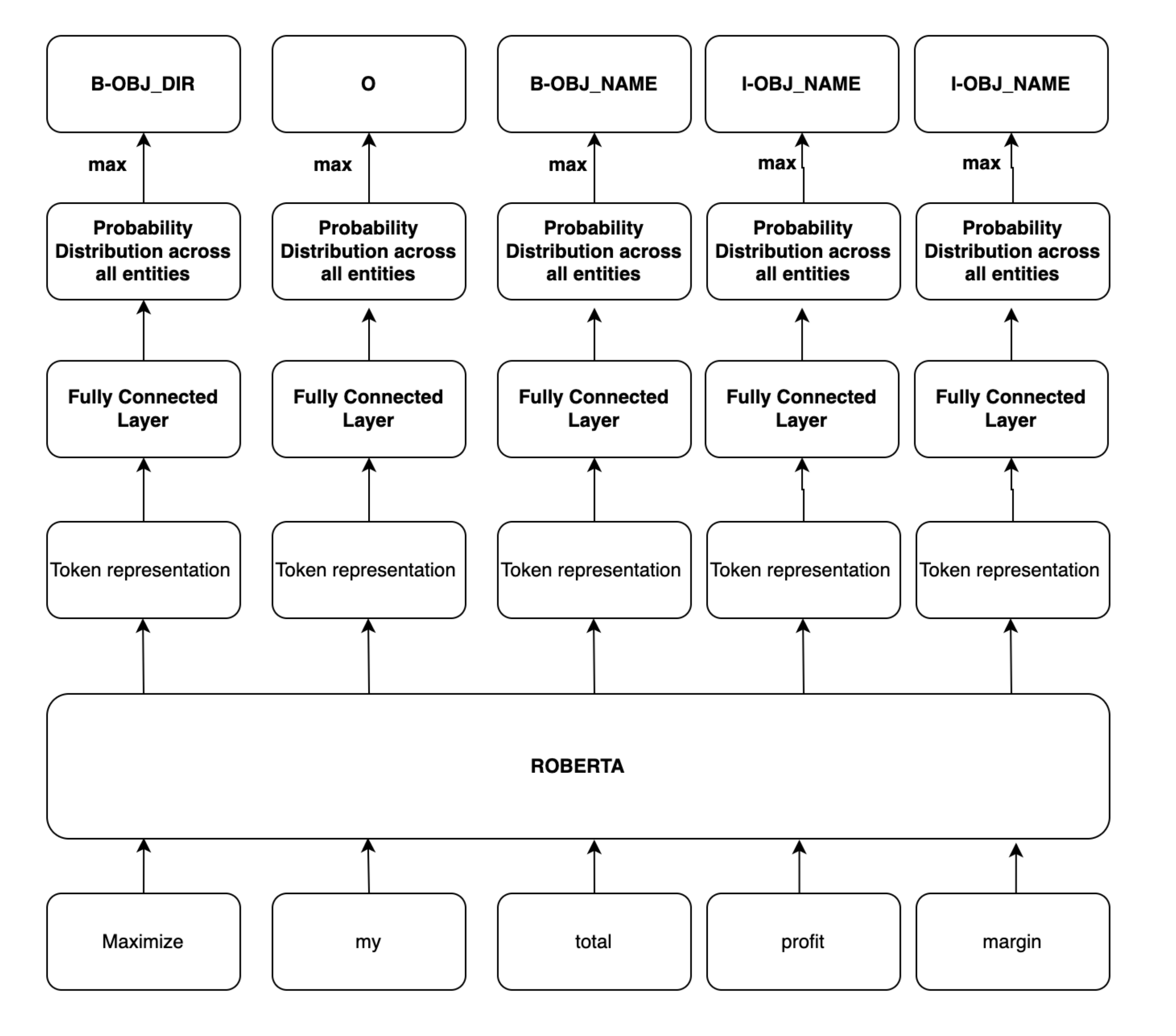
Classical+: Feature Engineering for Optimization



profit SUBJ to be maximized OBJ_DIR

maximize OBJ_DIR the total monthly ADJP profit NOUN

Modern NLP Approach



Maximize my total profit margin

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. The researchers 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.

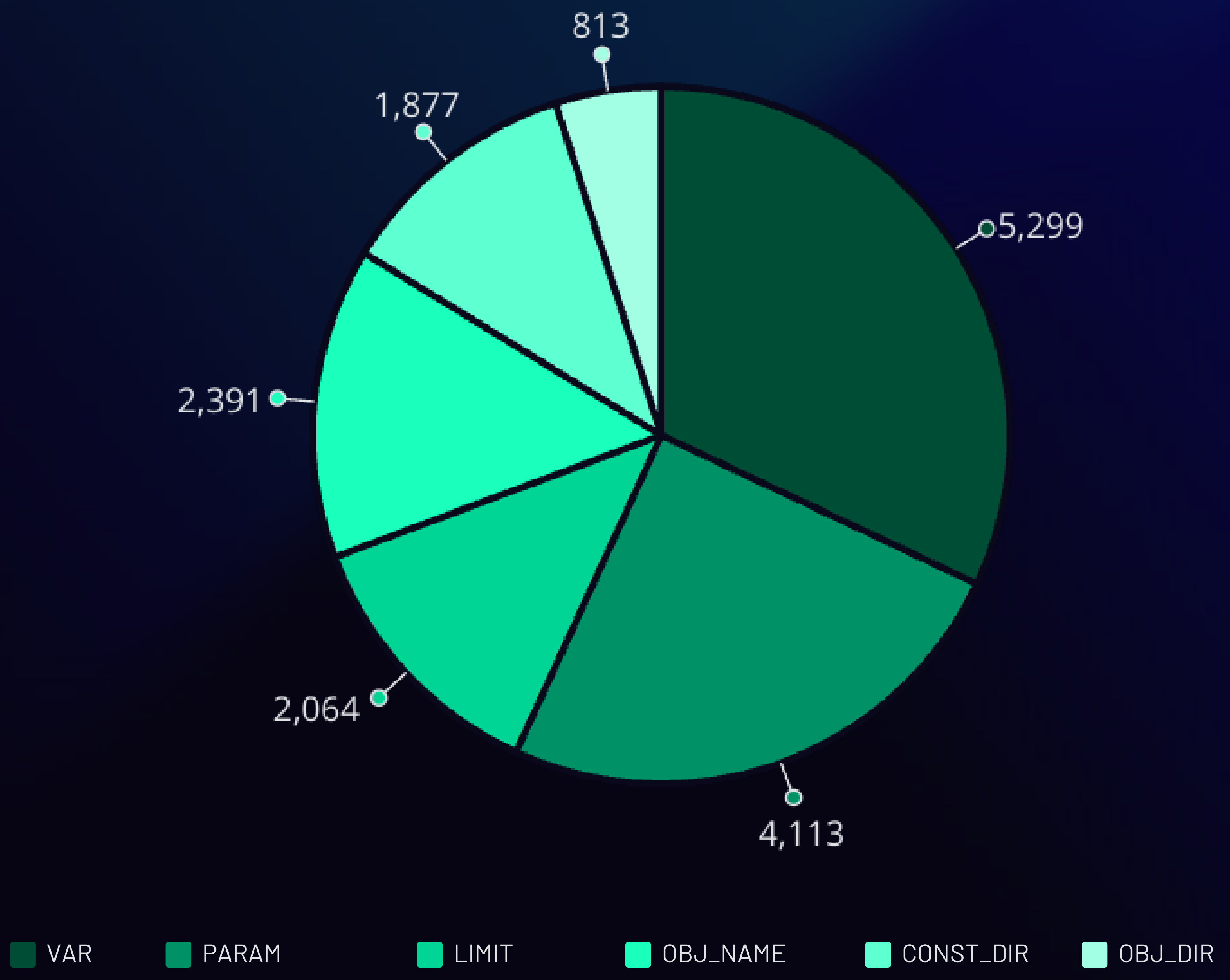
4

Unified Model

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

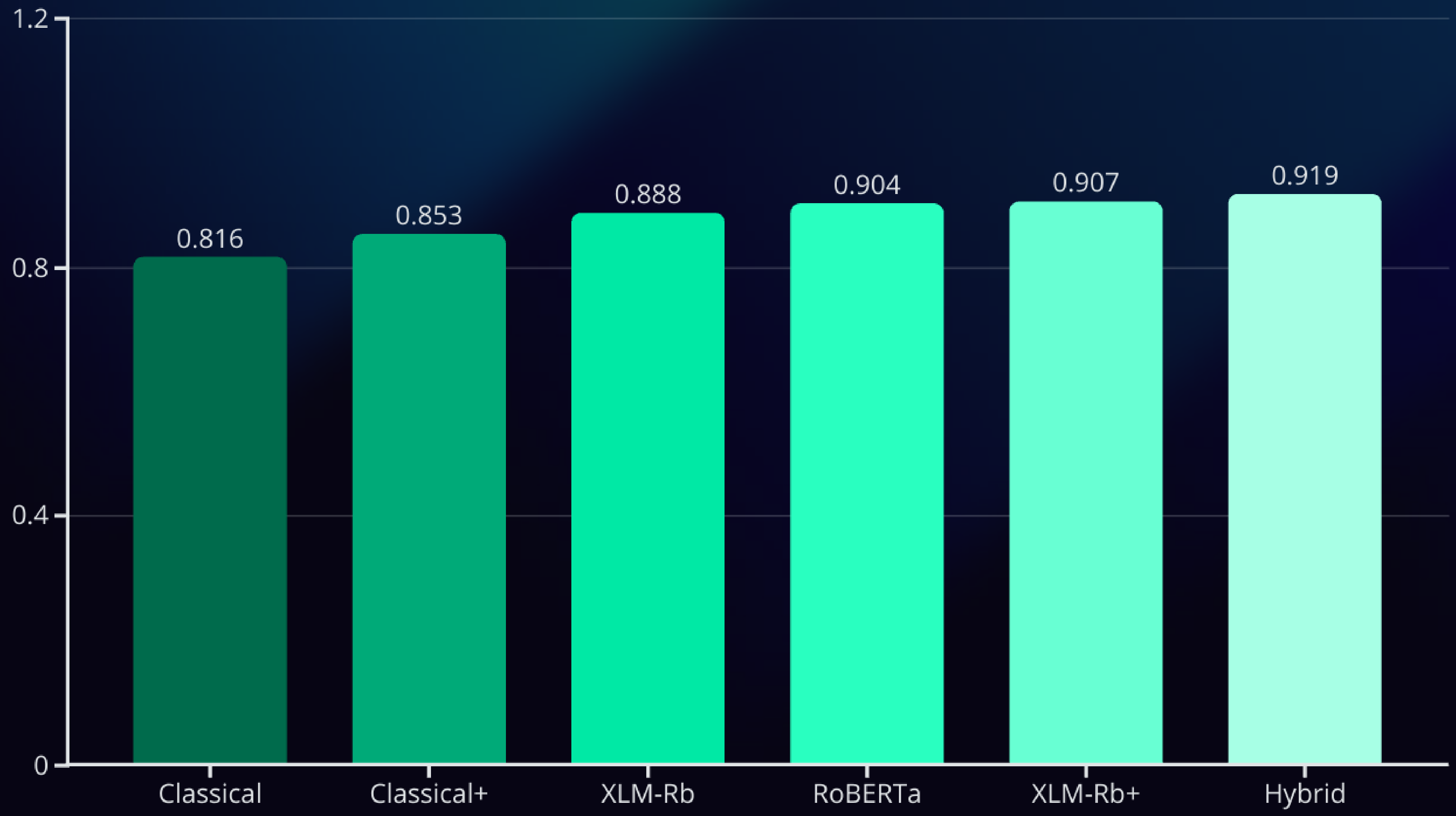
Experimental Setup

- 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.

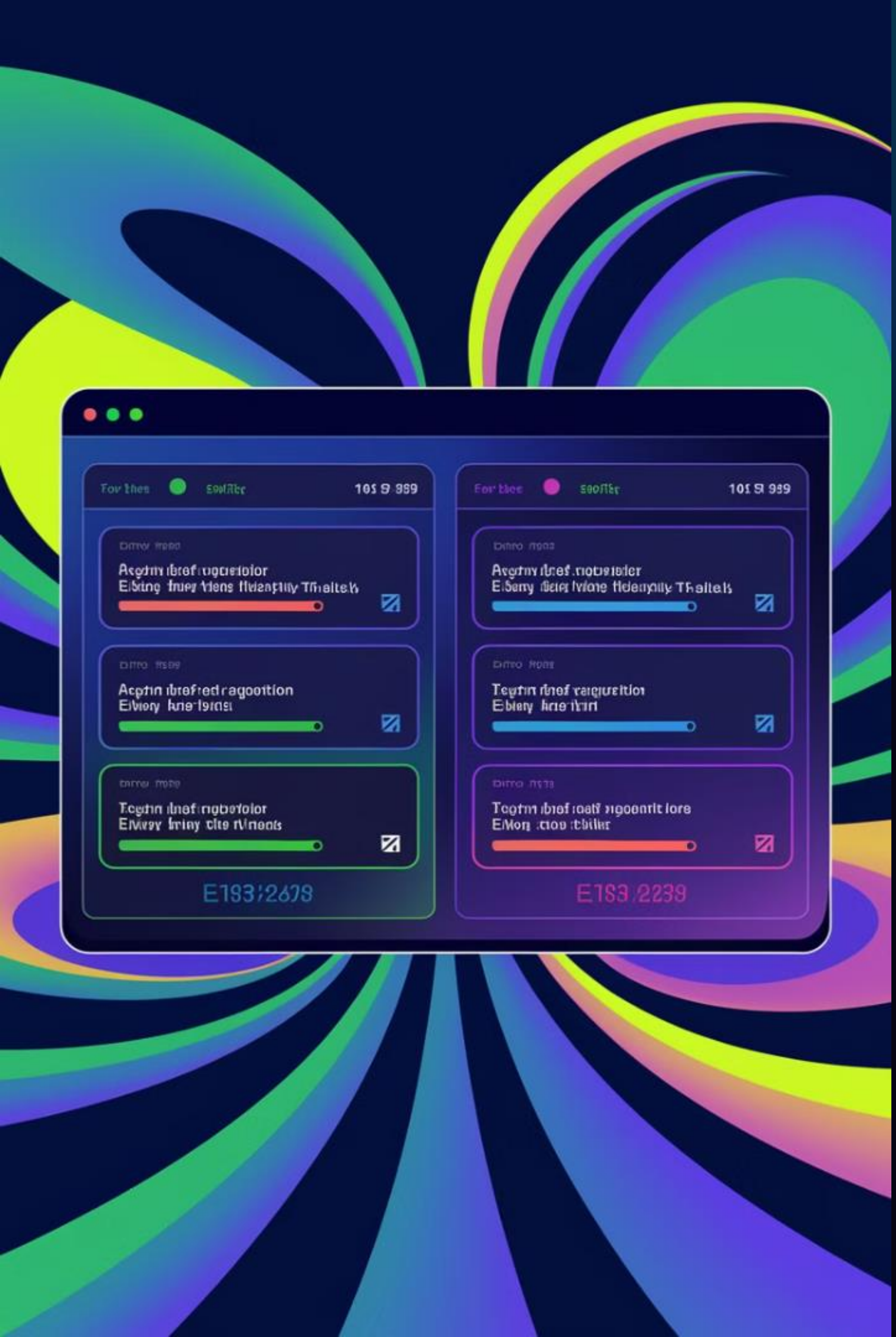


Ramamonjison et. al., NL4Opt Competition: Formulating Optimization Problems Based on Natural Language Descriptions

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.

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

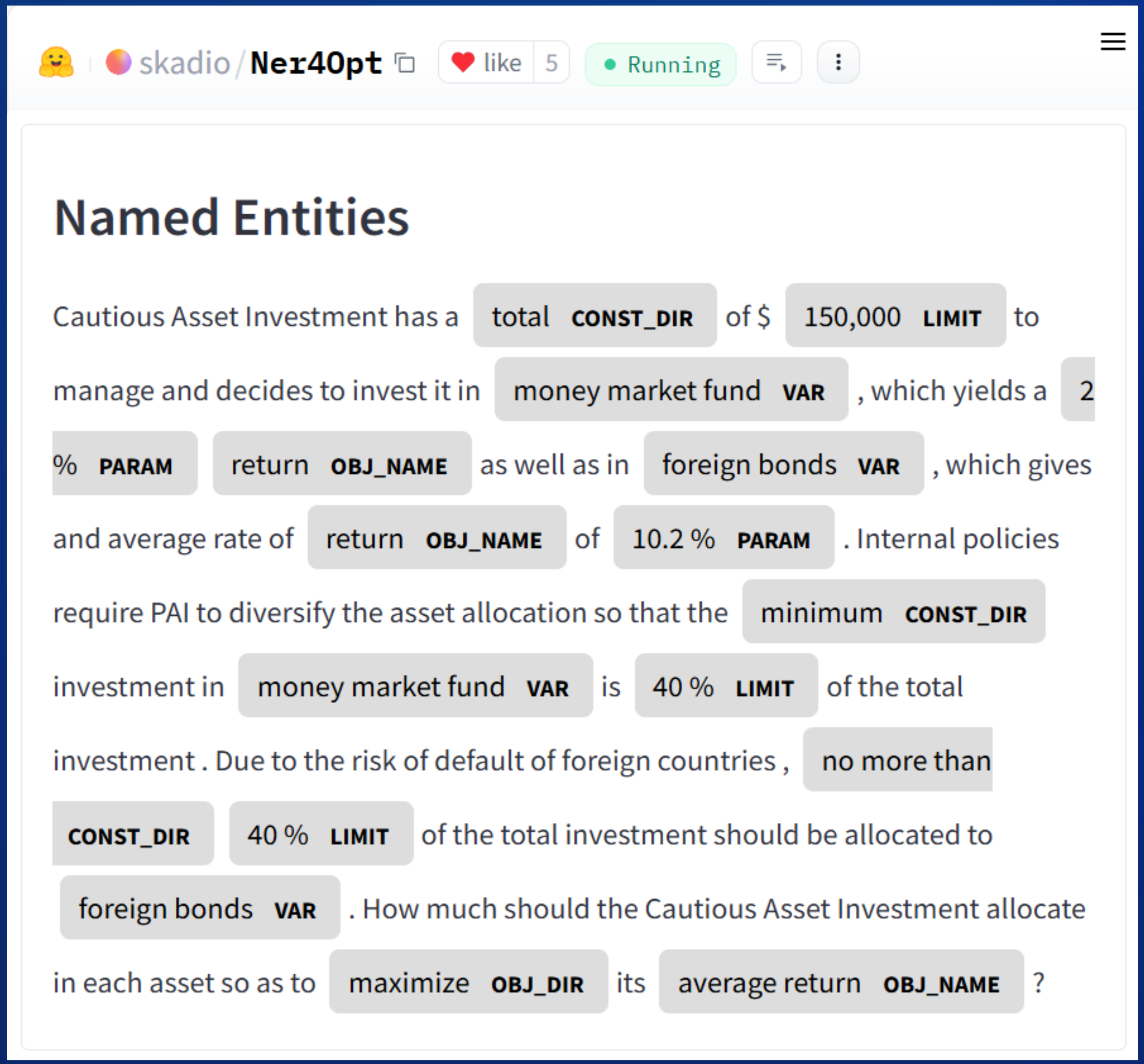
Interactive Demo

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

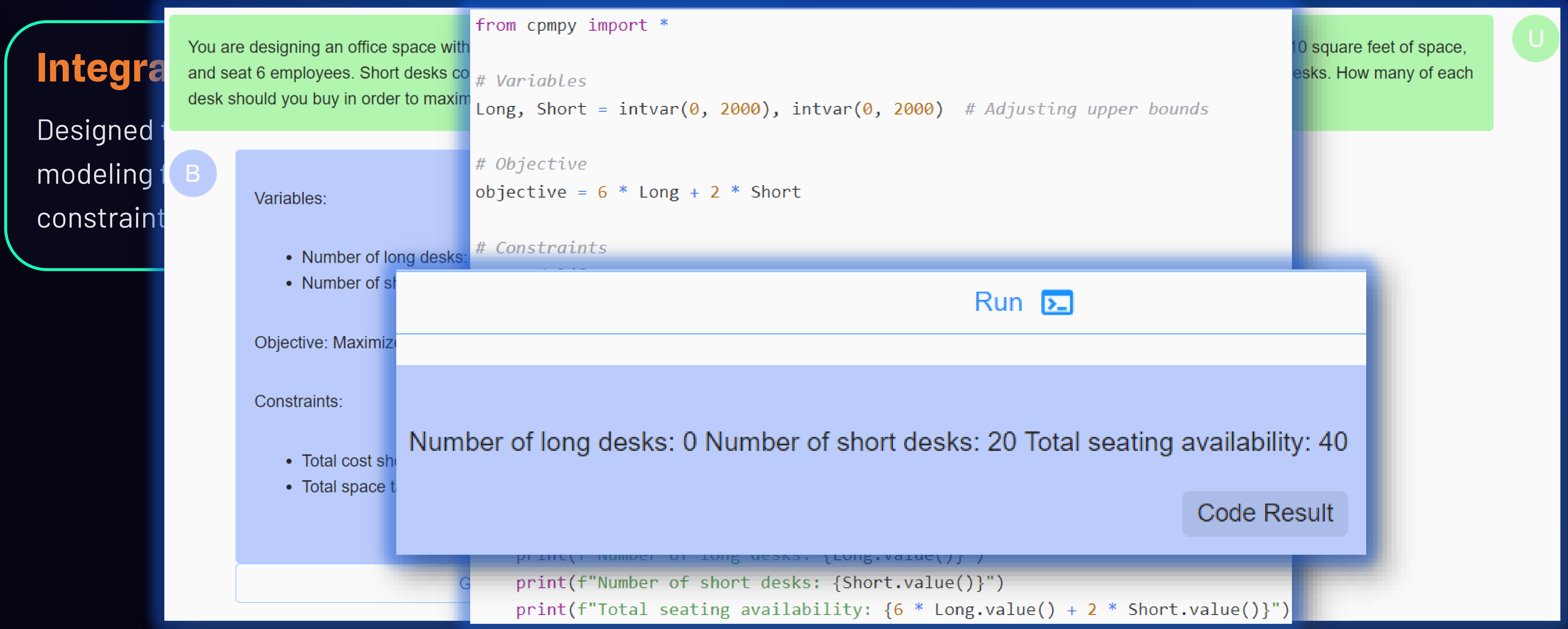
Users can choose between **different model** types (lexical, lexical-plus, semantic, hybrid) to compare performance on the same input text.



Hugging Face Spaces Demo

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

ChatOpt Integration



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

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 LLMs 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 mixed integer programming
- 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

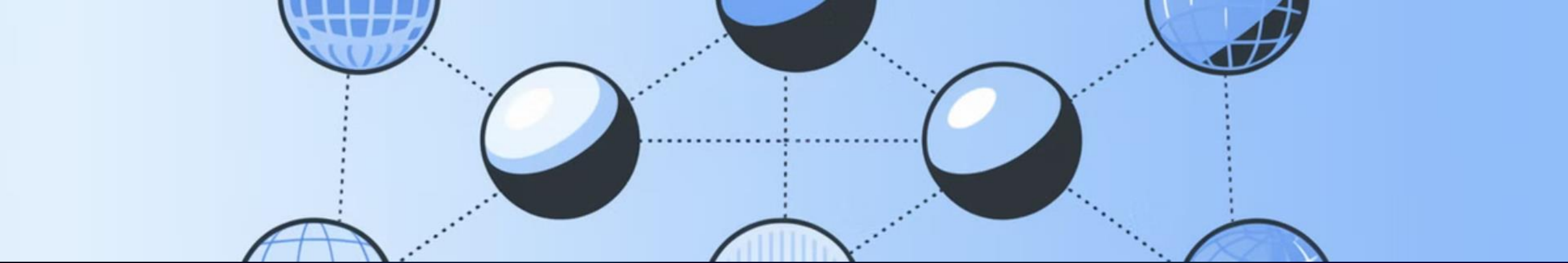
- CP and Satisfaction problems
- Not designed to work with ML or LLMs

Hakank's Models

- Extensive set of constraint programming 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

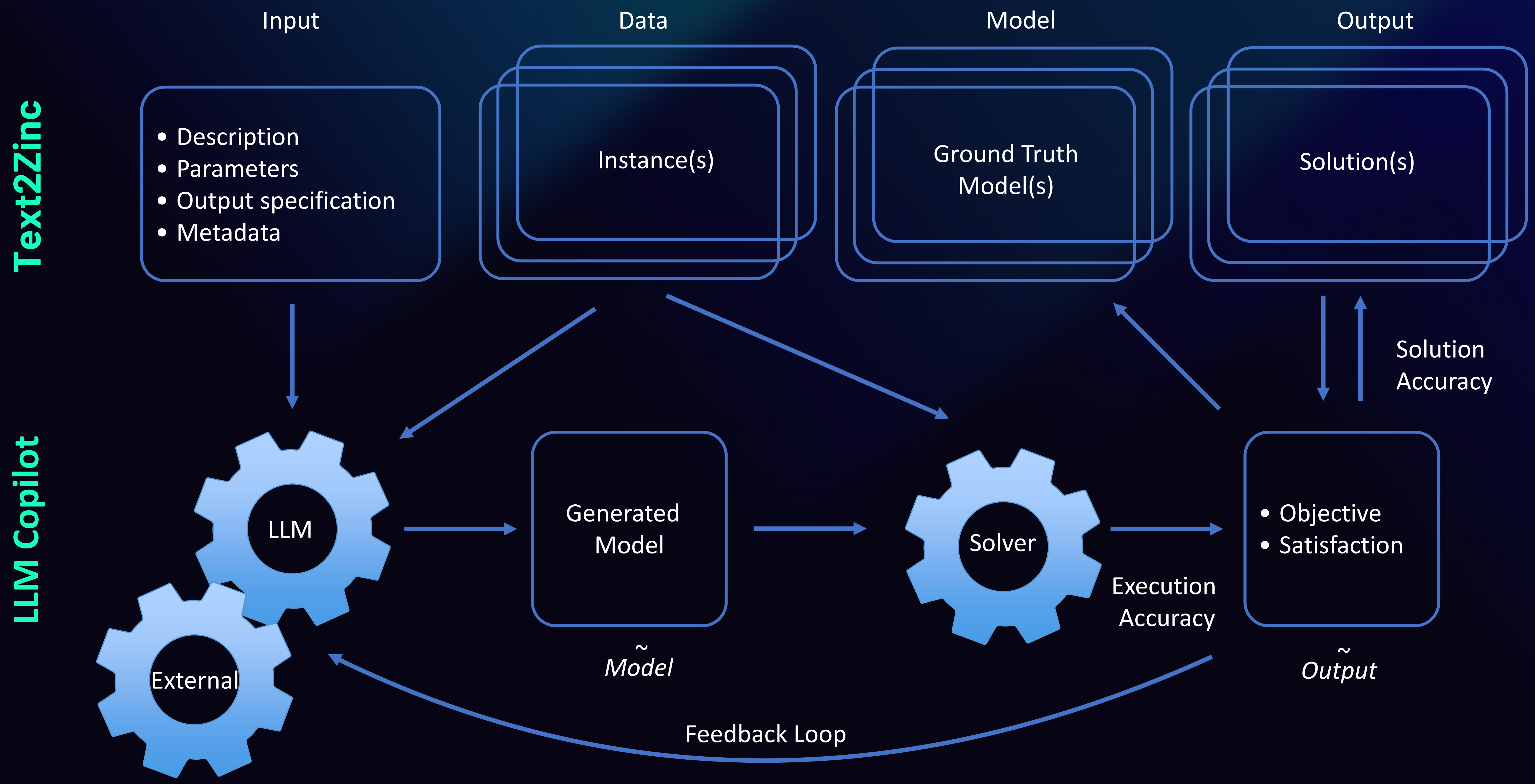
- Enables **solver agnostic** approaches.
- MIP, CP, SAT, LCG through MiniZinc.

4

Data Augmentation

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

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

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

Text2Zinc Initial 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

Evaluation

Assessment of execution and solution accuracy

Text2Zinc Initial Results

Solution Approach	Execution Accuracy	Solution Accuracy
Out-of-the-box Prompting	0.1904	0.0634
+ Data & examples	0.3650	0.1904
+ Shape information	0.1904	0.1269
+ Knowledge Graph	0.3492	0.1111
Chain-of-Thought (COT)	0.4285	0.1746
+ In-context examples	0.5873	**0.2539**
+ Shape information	0.5555	0.2063
Multi-Call + Composition	**0.6031**	0.2222

Hugging Face Text2Zinc Leaderboard

Future Directions and Applications

Integration with Solvers

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

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.

Interactive Modeling

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

What's Next?

1 Ner4Opt & Text2Zinc

- Ner4Opt and Text2Zinc dataset is now available on Hugging Face
- Providing a valuable resource for researchers and practitioners in the field

2 Performance

- Explore other approaches
- Contribute to the Text2Zinc leaderboard to establish comprehensive benchmarks

3 Contributions

We encourage contributions to the dataset through new problems

[Hugging Face Text2Zinc Dataset & Leaderboard]

[skadio.github.io]