AAAI 2024

The 38th Annual AAAI Conference on Artificial Intelligence
Bridge Program on Constraint Programming and Machine Learning
Vancouver, Canada

Large Language Models for Optimization Modelling

Serdar Kadıoğlu

Group VP, AI Center of Excellence, Fidelity Investments
Adj. Assoc. Prof., Computer Science Department, Brown University

skadio.github.io

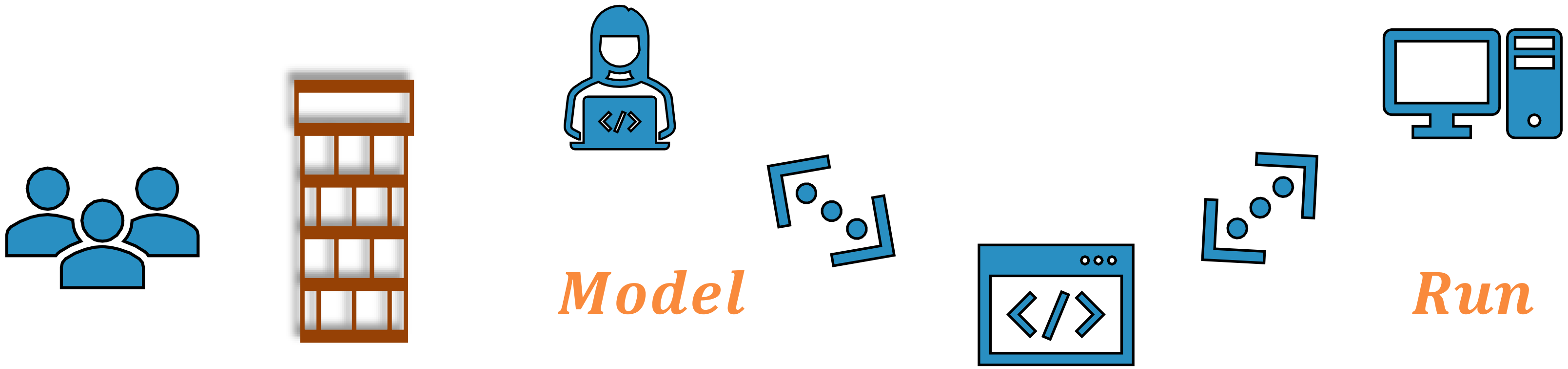
Introduction

Optimization Technology

❑ **Optimization** technology enjoys a wide range of applications

❑ Over the years, **dramatical speed-ups** enabled by theoretical and practical advances

❑ The overall **process** of modeling and solving problems remained the same for decades



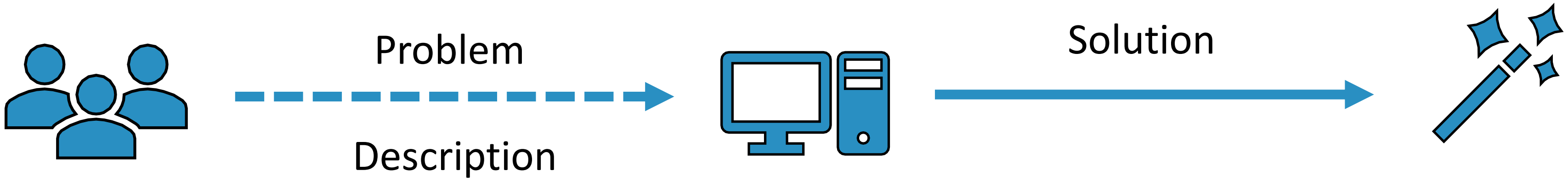
Model

Run

Introduction

Towards the Holy Grail

❑ Can we achieve the **Holy Grail** with Large Language Models?



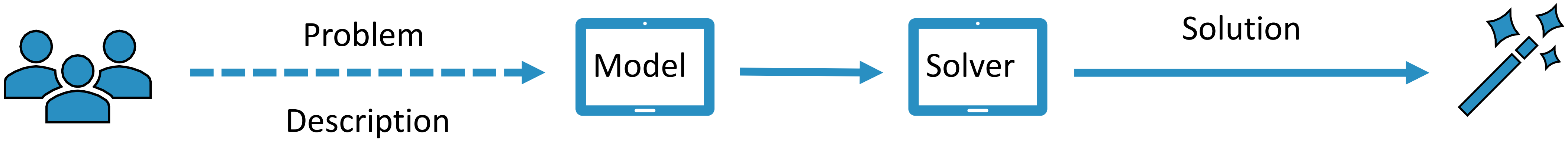
❑ LLMs still lack **reasoning** for solving combinatorial problems, even on simple puzzles

❑ We already know how to solve such problems! The bottleneck is to **model** them

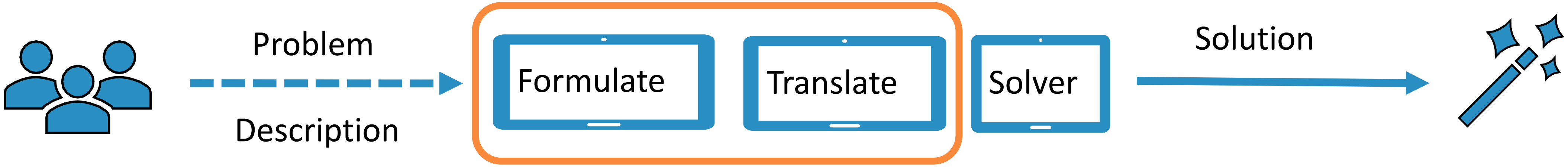
Introduction

Holy Grail 2.0

❑ **Holy Grail 2.0:** From natural language to constraint models



❑ Leverage LLM capabilities to model problems and then turn to powerful solving techniques

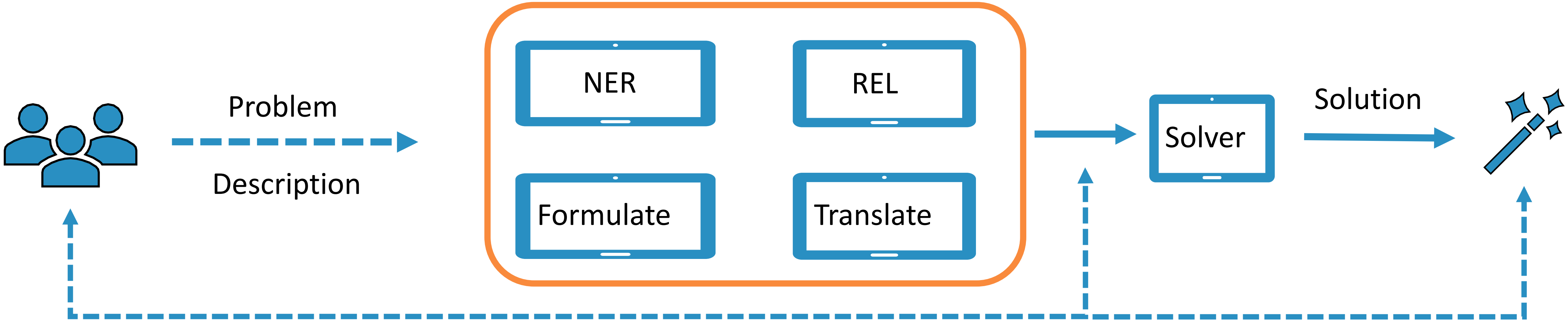


Holy Grail 2.0: From Natural Language to Constraint Models, CP 2023
D. Tsouros, H. Verhaeghe, S. Kadıoğlu, T. Guns

Introduction

Automated Modelling Assistant

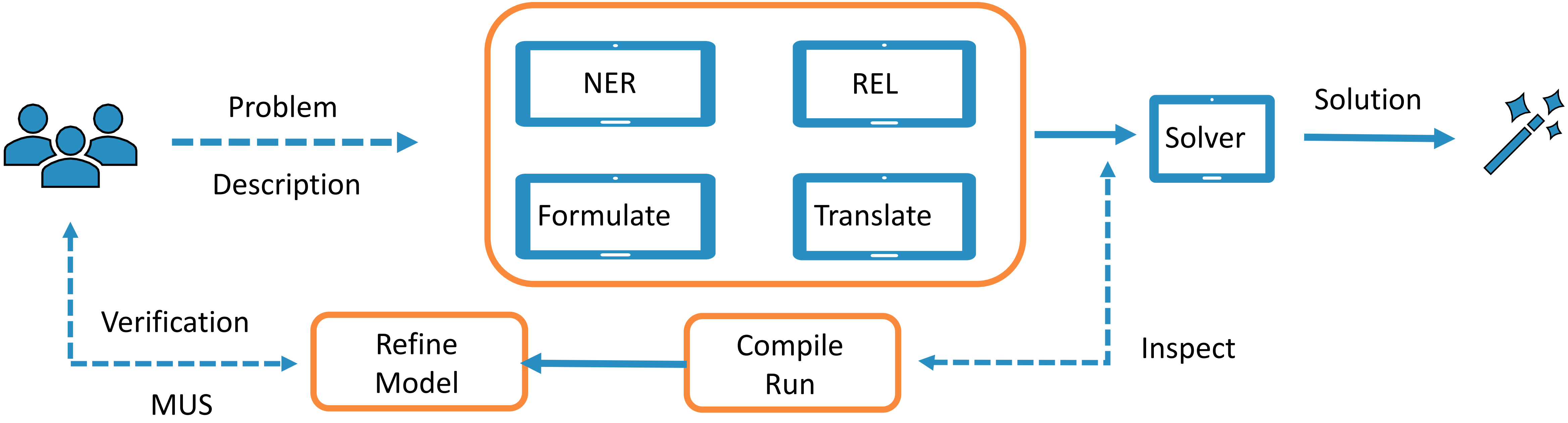
- ❑ Decompose into necessary **building blocks**
- ❑ LLMs and other technologies can be used in each block



Introduction

Conversational Constraint Solving: Acquisition – Automation – Explanation

- ❑ What if the user needs **explanation** for the results?
  - ○ Problem is unsatisfiable
  - ○ User not satisfied with the solution
- ❑ What if **additional constraints** need to be added?
  - ○ Constraint acquisition



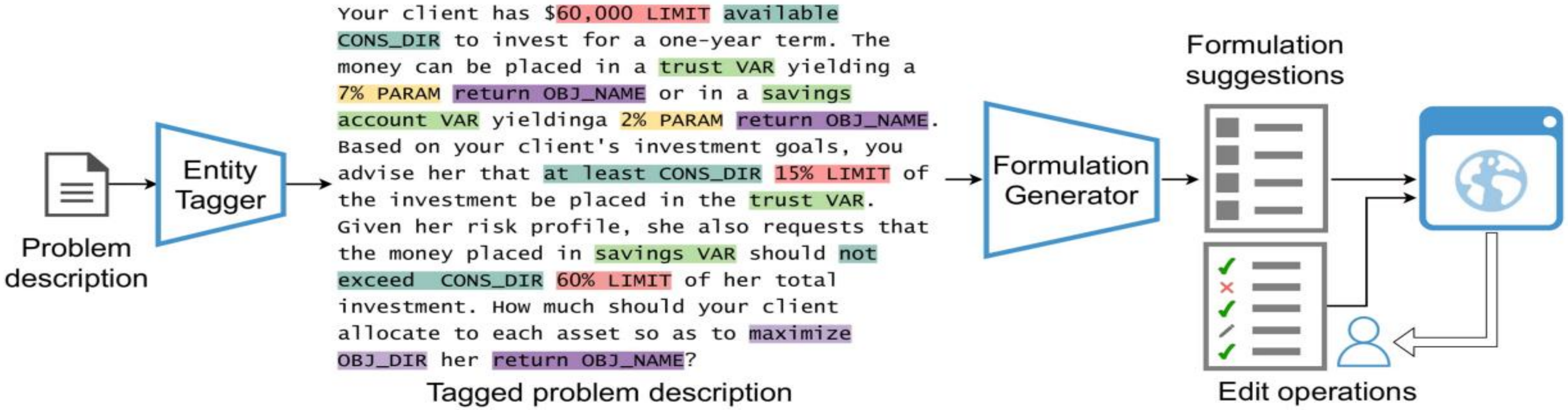
Introduction

Recent NL4OPT Challenge

❑ **NL4OPT** was proposed @ EMNLP'22

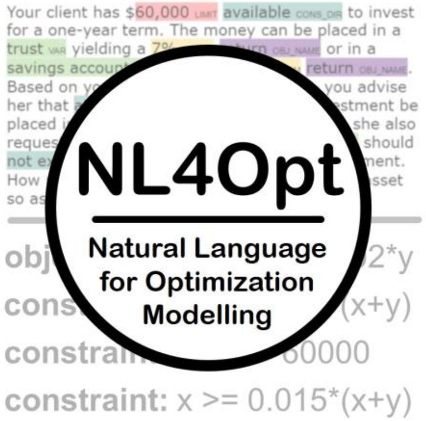
❑ Two subtasks were considered: **NER** and **Formulate**

❑ The first dataset for these problems was introduced, used in **NL4OPT Challenge** @ NeurIPS'22



Your client has $60,000 LIMIT available CONS_DIR to invest for a one-year term. The money can be placed in a trust VAR yielding a 7% PARAM return OBJ_NAME or in a savings account VAR yieldinga 2% PARAM return OBJ_NAME. Based on your client's investment goals, you advise her that at least CONS_DIR 15% LIMIT of the investment be placed in the trust VAR. Given her risk profile, she also requests that the money placed in savings VAR should not exceed CONS_DIR 60% LIMIT of her total investment. How much should your client allocate to each asset so as to maximize OBJ_DIR her return OBJ_NAME?

Tagged problem description



NL4Opt
Natural Language for Optimization Modelling
constraint: x >= 0.015*(x+y)

Ramamonjison et al., Augmenting Operations Research with Auto-Formulation of Optimization Models from Problem Descriptions, EMNLP 2022
Ramamonjison et al., NL4Opt Competition: Formulating Optimization Problems Based on Their Natural Language Descriptions, NeurIPS 2022

Introduction

Related work

- ❑ **LGPSolver:** Solving logic grid puzzles automatically, ACL’20
- ❑ Automatic formulation and optimization of linear problems from a structured paragraph, ICSCT’21
- ❑ **Optimus:** optimization modeling using MIP/LLM, AhmadiTeshnizi et. al. 2023
- ❑ 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
- ❑ **IBM modelling assistant** with templatized domain applications
- ❑ **OptiGuide:** LLMs for Supply Chain Optimization, Microsoft, 2023
- ❑ **MeetMate:** Enabling interactive decision support using LLMs and CP, Lawless et al., 2023
- ❑ Towards an Automatic Optimization Model Generator Assisted with GPT, Almonacid, 2023
- ❑ **LM4OPT** A Unreiviling the potential of LLMs for Optimization Modelling, Tasnim et. al., 2024
- ❑ **SciBERT** A pretrained language model for scientific text, Beltagy et. al., EMNLP’19
- ❑ **seq2sql:** generating structured queries from NL using RL and LLMs, Zhong et. al. 2017
- ❑ Learning natural language interfaces with neural models, Li Dong, PhD thesis, 2019
- ❑ …

Demo: Ner4Opt & ChatOpt

Ner4Opt Hugging Face Spaces (CPAIOR’23)
https://huggingface.co/spaces/skadio/Ner4Opt

Modeling Assistant Demo (CP’23)
https://chatopt.cs.kuleuven.be

ChatOpt deep-dive

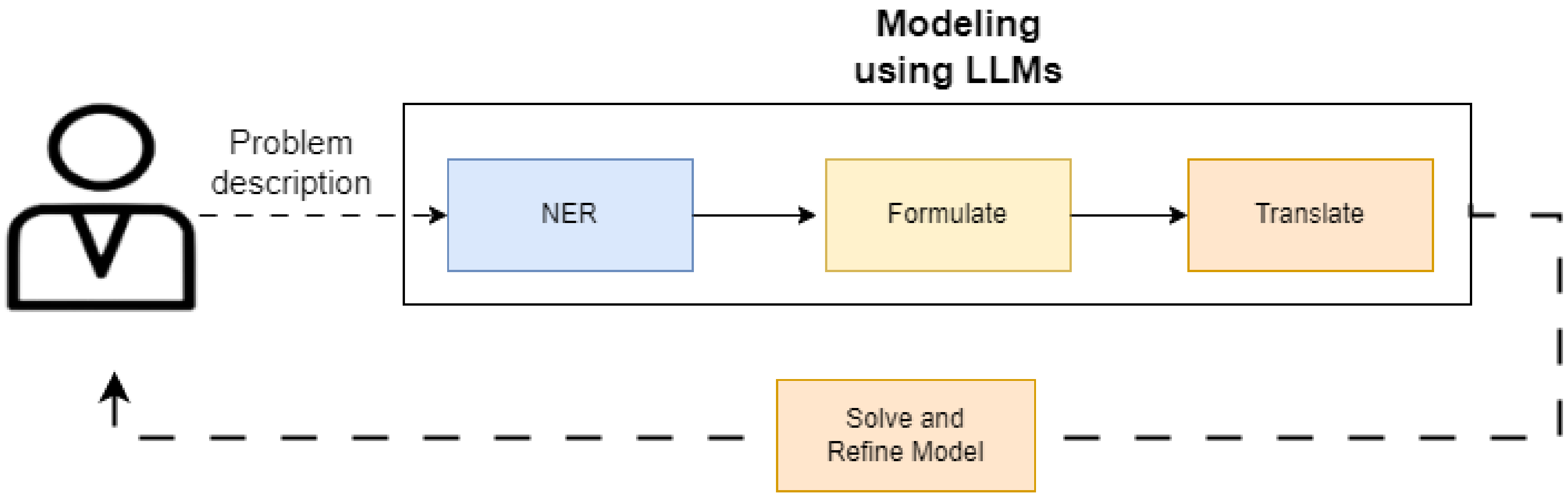
Ner4Opt deep-dive

What’s next?

Holy Grail 2.0: From Natural Language to Constraint Models, CP 2023
D. Tsouros, H. Verhaeghe, S. Kadıoğlu, T. Guns

ChatOpt

What’s under the hood?

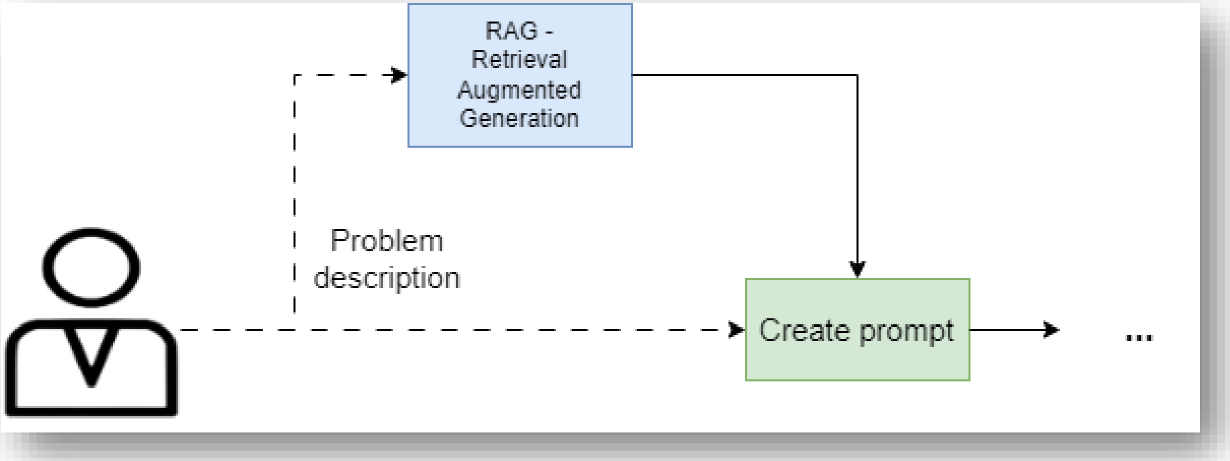


- ❑ Ongoing research
  - ○ Large Language Models used for each step
  - ○ In-context Learning and Chain-of-thought used
- ❑ Current state in the beta version:
  - ○ No REL step yet, experimenting with NER
  - ○ Still not there for the goal of conversational constraint solving

ChatOpt: LLMs as CP modellers

What’s under the hood?

❑ In-Context Learning

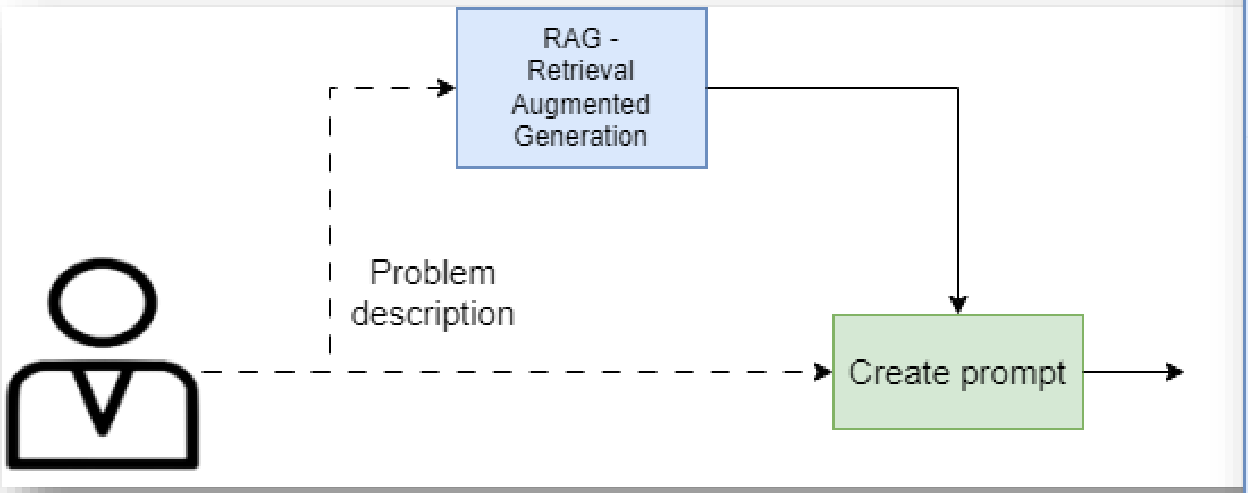


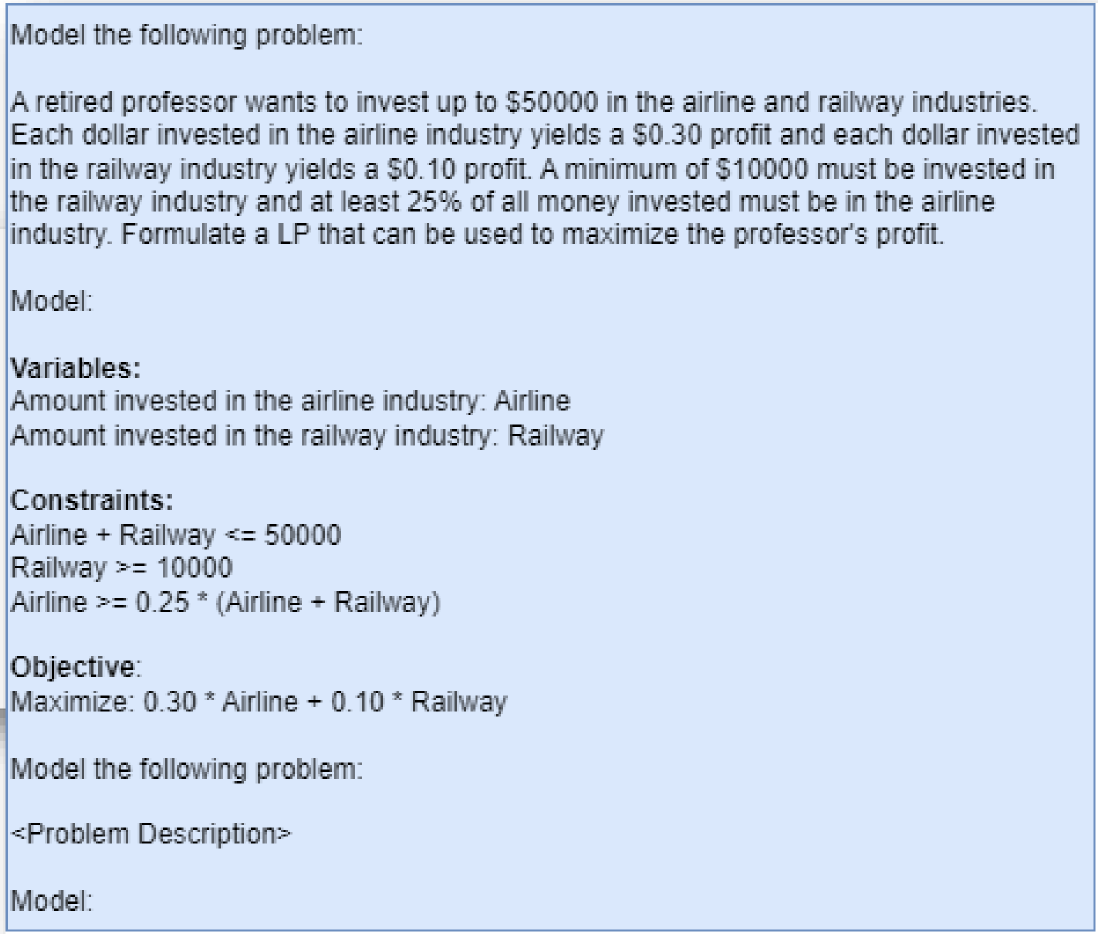
❑ Dynamically selecting the examples (shots) based on the current problem:

- ○ Random selection
- ○ RAG:
  - ○ Similarity selection: Select the most similar ones (cosine similarity)
  - ○ Maximal Marginal Relevance (MMR): Balance diversity and relevance in example selection

ChatOpt: LLMs as CP modellers

In-Context Learning





Model the following problem:

A retired professor wants to invest up to $50000 in the airline and railway industries. Each dollar invested in the airline industry yields a $0.30 profit and each dollar invested in the railway industry yields a $0.10 profit. A minimum of $10000 must be invested in the railway industry and at least 25% of all money invested must be in the airline industry. Formulate a LP that can be used to maximize the professor's profit.

Model:

Variables:
Amount invested in the airline industry: Airline
Amount invested in the railway industry: Railway

Constraints:
Airline + Railway <= 50000
Railway >= 10000
Airline >= 0.25 * (Airline + Railway)

Objective:
Maximize: 0.30 * Airline + 0.10 * Railway

Model the following problem:

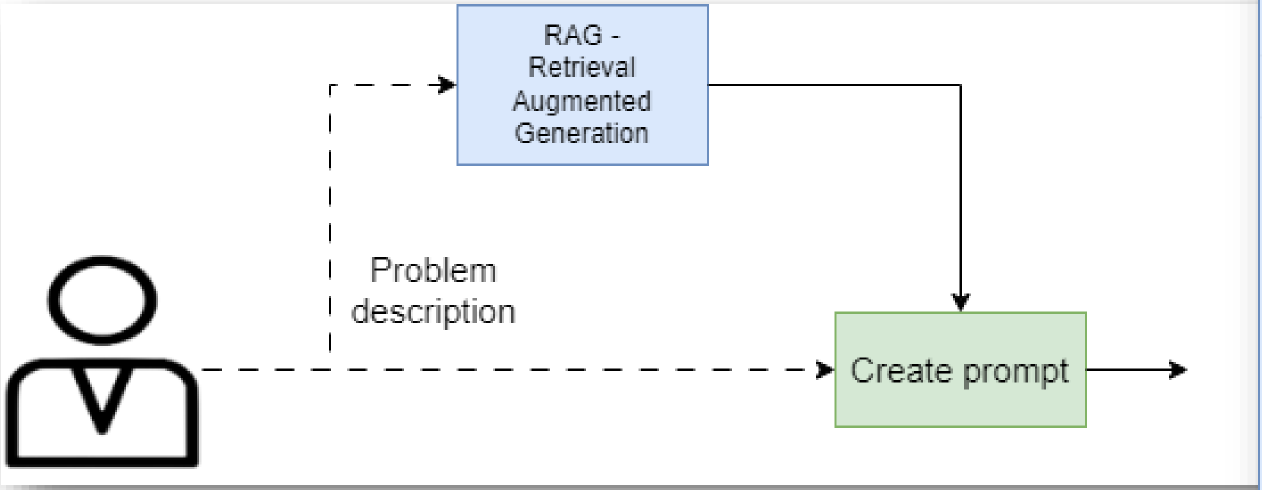
<Problem Description>

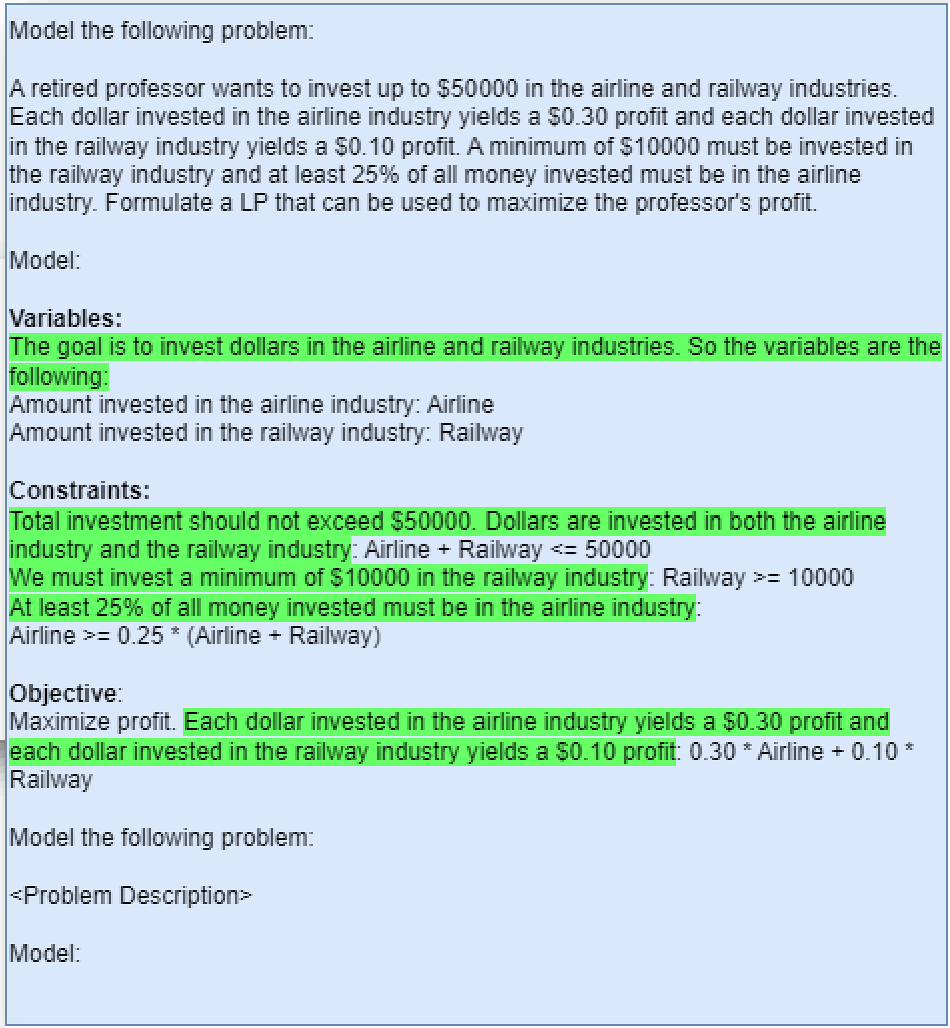
Model:

ChatOpt: LLMs as CP modellers

Chain-of-thought

- ❑ Augment the description of the solution to the given problem(s) with explanation
- ❑ That is, using chain-of-thought





Model the following problem:

A retired professor wants to invest up to $50000 in the airline and railway industries. Each dollar invested in the airline industry yields a $0.30 profit and each dollar invested in the railway industry yields a $0.10 profit. A minimum of $10000 must be invested in the railway industry and at least 25% of all money invested must be in the airline industry. Formulate a LP that can be used to maximize the professor's profit.

Model:

Variables:
The goal is to invest dollars in the airline and railway industries. So the variables are the following:
Amount invested in the airline industry: Airline
Amount invested in the railway industry: Railway

Constraints:
Total investment should not exceed $50000. Dollars are invested in both the airline industry and the railway industry: Airline + Railway <= 50000
We must invest a minimum of $10000 in the railway industry: Railway >= 10000
At least 25% of all money invested must be in the airline industry:
Airline >= 0.25 * (Airline + Railway)

Objective:
Maximize profit. Each dollar invested in the airline industry yields a $0.30 profit and each dollar invested in the railway industry yields a $0.10 profit: 0.30 * Airline + 0.10 * Railway

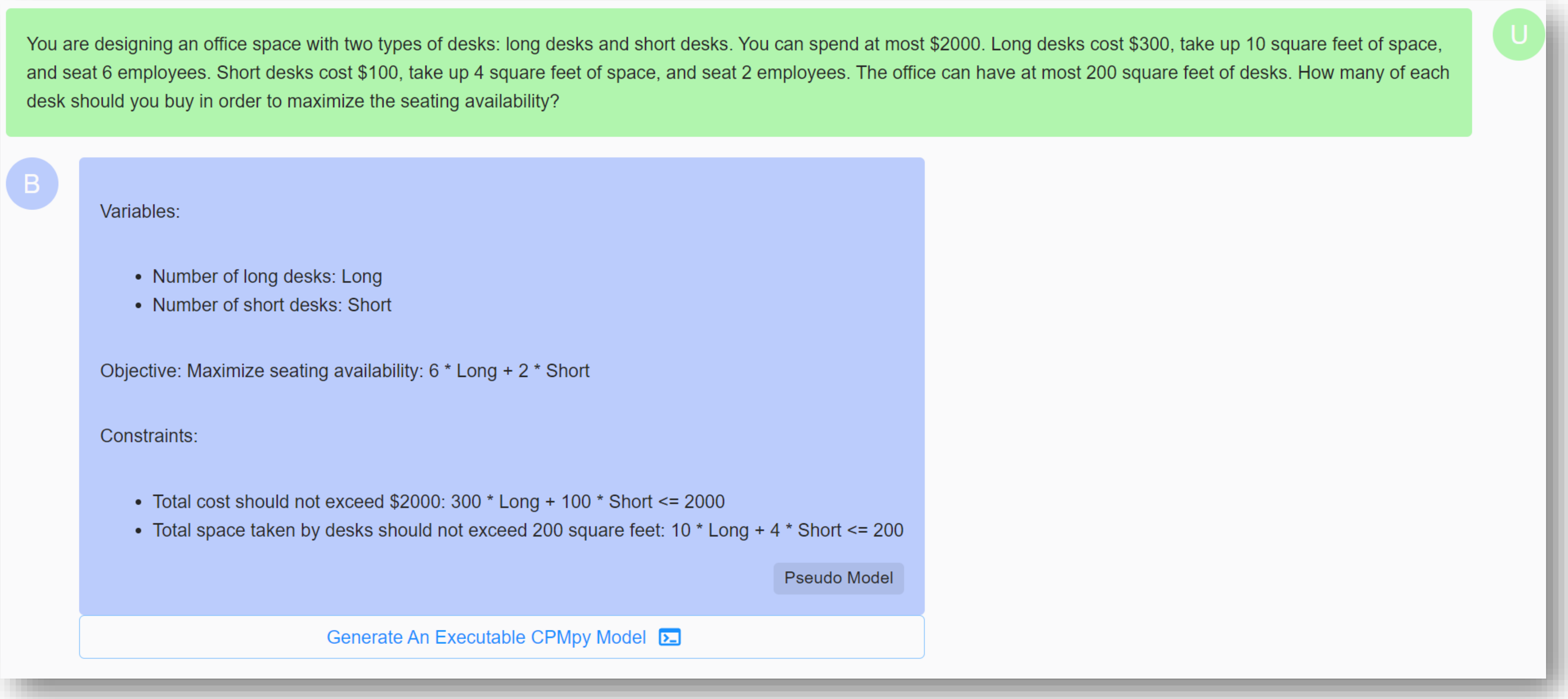
Model the following problem:

<Problem Description>

Model:

ChatOpt: Step – I

Produce the pseudo-model



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?

Variables:

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

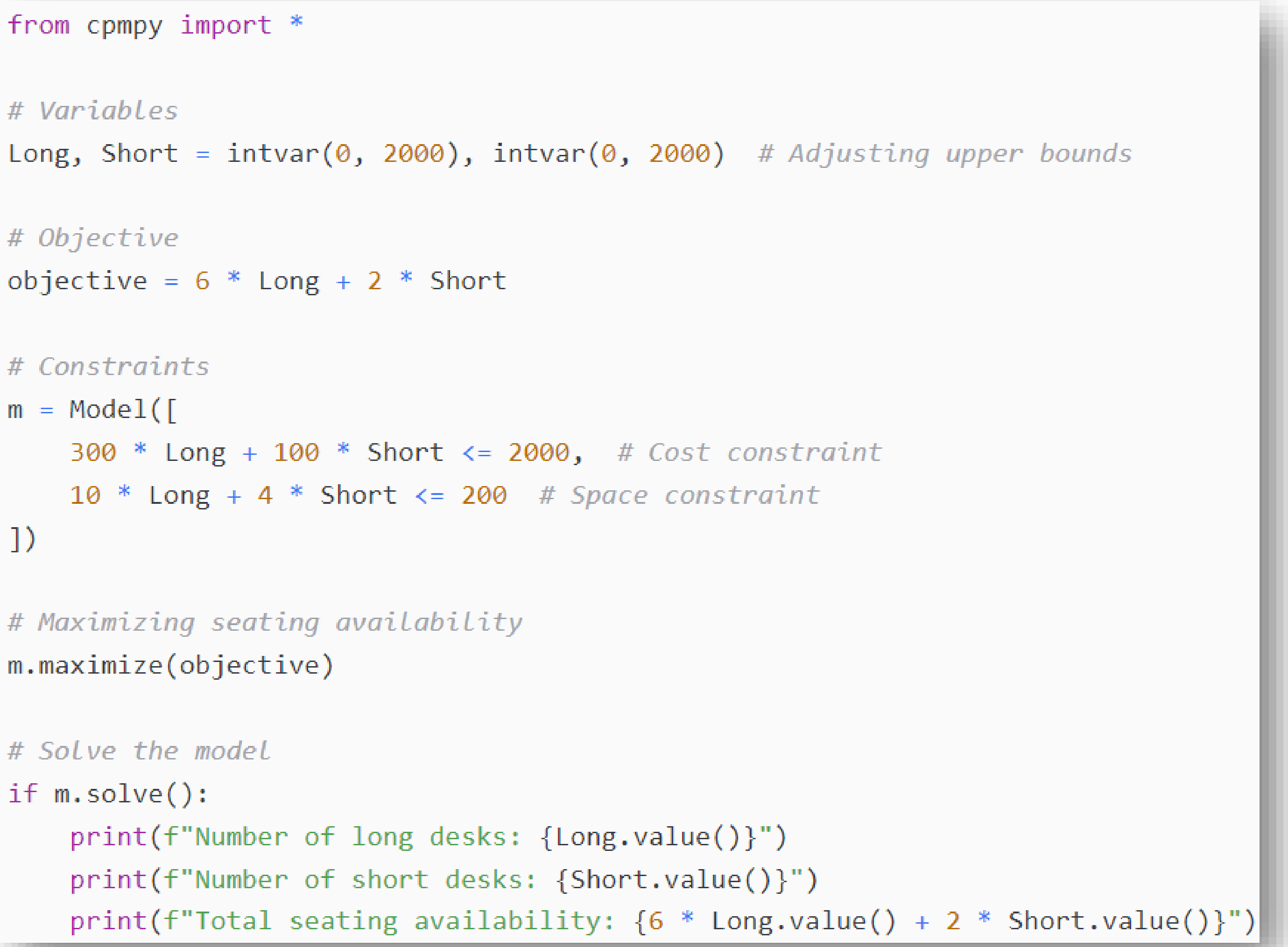
Objective: Maximize seating availability: 6 * Long + 2 * Short

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

ChatOpt: Step – II

Generate CPMpy code

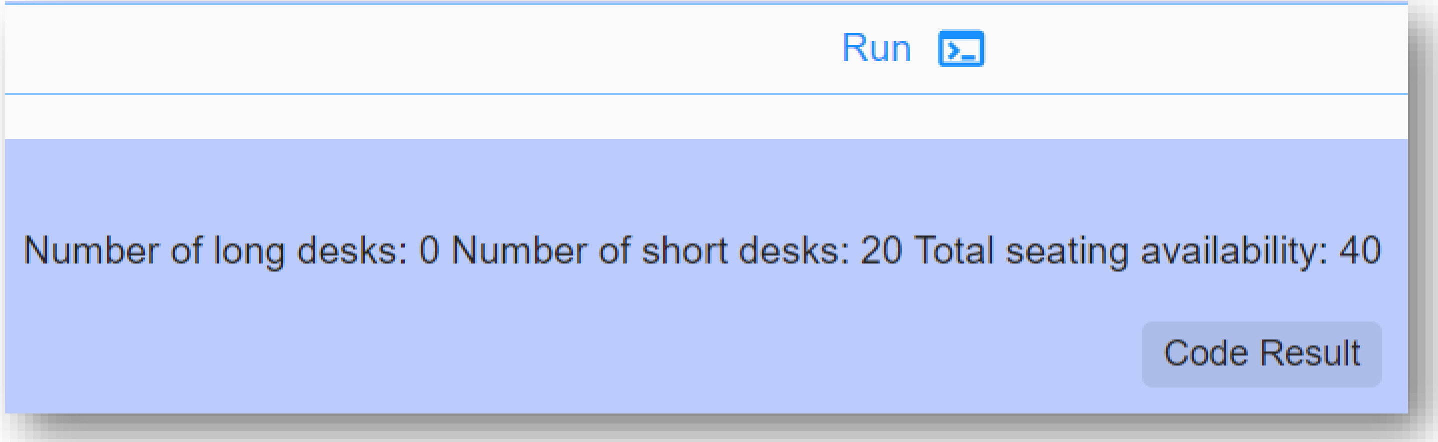


```
from cpmpy import *

# Variables
Long, Short = intvar(0, 2000), intvar(0, 2000)  # Adjusting upper bounds

# Objective
objective = 6 * Long + 2 * Short

# Constraints
m = Model([
    300 * Long + 100 * Short <= 2000,  # Cost constraint
    10 * Long + 4 * Short <= 200  # Space constraint
])

# Maximizing seating availability
m.maximize(objective)

# Solve the model
if m.solve():
    print(f"Number of long desks: {Long.value()}")
    print(f"Number of short desks: {Short.value()}")
    print(f"Total seating availability: {6 * Long.value() + 2 * Short.value()}")
```

https://cpmpy.readthedocs.io

ChatOpt: Step – III

Execute the code and get the solution



```
Number of long desks: 0 Number of short desks: 20 Total seating availability: 40
```

Experiments

Initial Results

❑ Initial results on 2 CP datasets:

- COPs: NL4Opt https://github.com/nl4opt/nl4opt-competition/tree/main/generation_data
- CSPs: Logic Grid Puzzles https://github.com/jelgun/LGPSolver/tree/master/data

❑ Evaluating accuracy on 3 different levels:

- Constraint-level
- Problem-level
- Solution-level

Experiments

Initial Results on NL4OPT

❑ Using gpt-3.5-turbo-1106 to generate pseudo-models

Ex. Selection	# Shots	Acc (Prob) %	Acc (Cons) %
Static	1	86.1	94.0
Similarity	1	84.7	94.3
Static	4	85.1	92.1
Similarity	4	91.7	96.8
MMR	4	92.0	96.5
MMR	8	92.7	97.3

Some observations:

- Adding in-context examples is efficient if they are relevant with the current problem
- No need to add more than 4

Experiments

Initial Results on LGP

❑ Using Mixtral-8x7B-v0.1 to generate CPMpy code

# Shots	Ex. Selection	Acc (Solution) %
1	Similarity	72.0
2	MMR	77.0
4	MMR	80.0
8	MMR	87.0

Some observations:

- Still some way to go to achieve higher accuracy
- Difficulty to model such problems due to the combinatorial nature

ChatOpt deep-dive

Ner4Opt deep-dive

What’s next?

Ner4Opt: Named Entity Recognition for Optimization Modelling, NeurIPS’22, CPAIOR’23
P. Dakle, S. Kadıoğlu, K. Uppuluri, R. Politi, P. Raghavan, S. Rallabandi, R. Srinivasamurthym

Ner4Opt: Named Entity Recognition for Optimization Modelling

Problem Definition and Optimization Entities

Given a sequence of tokens $s = \langle w_1, w_2, \ldots, w_n \rangle$, the goal of Ner4Opt is to output a list of tuples $\langle I_s, I_e, t \rangle$ each of which is a named entity specified in s. Here, $I_s \in [1, n]$ and $I_e \in [1, n]$ are the start and end indexes of a named entity while t is the entity type from a predefined category set of constructs related to optimization.

Predefined Optimization Entities

- $\boldsymbol{VAR}$: The variables of the problem – two advertising channels: **morning tv show** and **social media**
- $\boldsymbol{CONST\_DIR}$: The constraint direction – social media spots needs to be **at least** 30
- $\boldsymbol{LIMIT}$: Limits of constraints – plan at least **4** but no more than **7** morning show spots
- $\boldsymbol{OBJ\_NAME}$: The objective variable – maximize the **reach** of the campaign
- $\boldsymbol{OBJ\_DIR}$: The direction of optimization – **maximize** the reach of the campaign
- $\boldsymbol{PARAM}$: The parameters of the problem – costs the company $**1,000** to run advertisement spots

Ner vs. Ner4Opt

Challenges of Optimization Context

❑ NER for **information retrieval**, question answering, and machine translation

❑ **Multi-sentence word problem** with high-level of compositionality, ambiguity, variability

❑ Ner4Opt must be **domain agnostic** and generalize to new instances and applications

❑ **Extremely limited training data**. Even human annotation requires expertise.
Must operate on low-resource regime

Chinchor et. al.: Message Understanding-7 named entity task definition, MUC, 1998

Solving the Ner4Opt

Classical and Modern NLP and their Hybridization

Conditional Random Field

Augmentation and Fine-Tuning

Solution Components

Features – Models – Data Centric Approach

1 **Feature Extraction, Engineering, and Learning**

Classical and semantic models to extract features for tokens while leveraging optimization context

2 **Conditional Random Field Neural Networks**

Linear chain conditional random field or fully connected network as the modeling component

3 **Data Augmentation Fine Tuning LLMs**

Augment the data set and fine-tune pre-trained large-language models

Conditional Random Field

Brief Introduction

Given an input sequence of tokens **x$_i$** and a set of feature extraction functions **f$_j$** at each token position, a **conditional random field** models a conditional probability distribution of labels **y$_i$** that can be assigned to appropriate segments in x.

$$D = [(x_1, y_1), (x_2, y_2), (x_3, y_3), \ldots., (x_d, y_d)] \quad i.i.d\ training\ examples \qquad (1)$$

CRF

$$score(y|x) = \sum_{j=1}^{m} \sum_{i=1}^{n} w_j f_j (x, i, y_i, y_{i-1}) \qquad (2)$$

$$p(y|x) \ = \ \frac{\exp^{score(y|x)}}{\sum_{y'} \exp^{score(y'|x)}} \qquad (3)$$

$$L(w, D) \ = \ -\sum_{k=1}^{d} log \left[p(y^k|x^k) \right] \qquad (4)$$

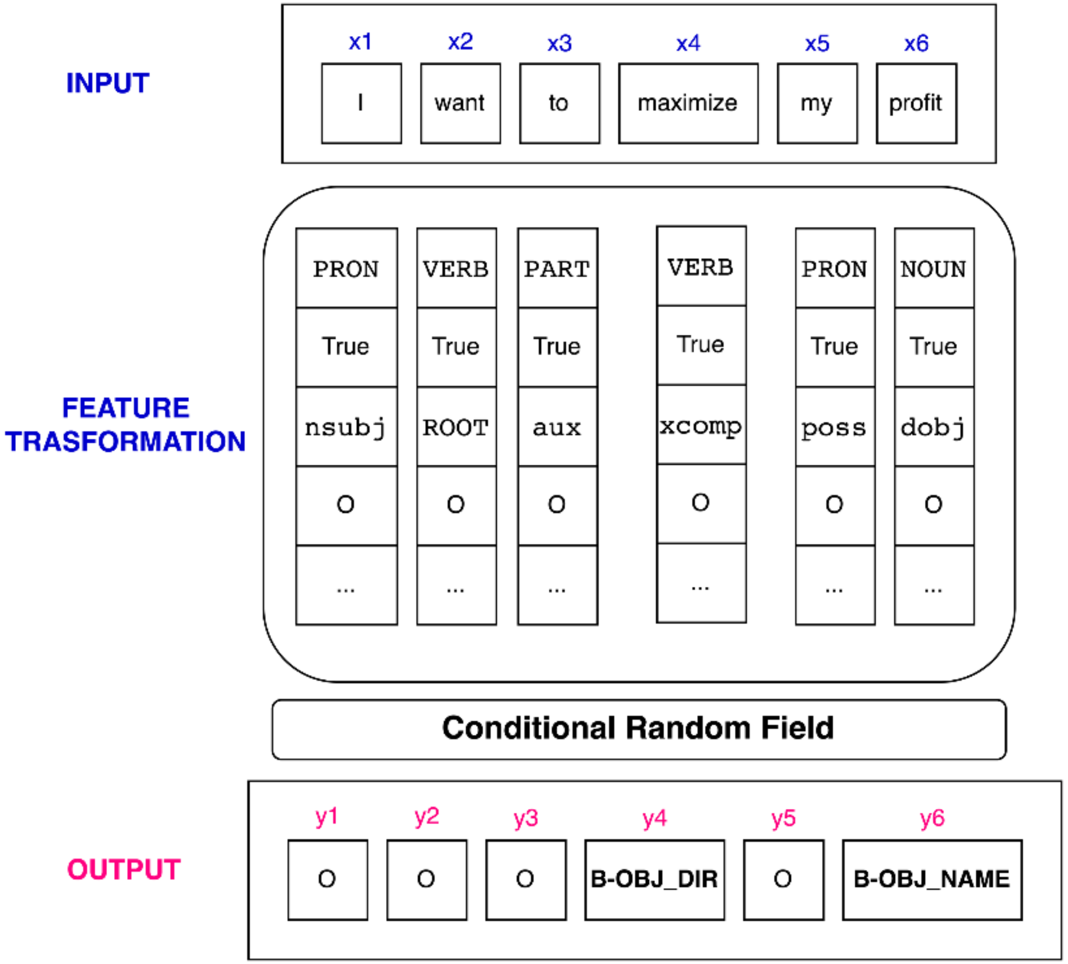
$$w^* \ = \ \arg\min_{w} \ L(w, D) \ + \ C \ \frac{1}{2} ||w||^2 \qquad (5)$$

Here, w is the weight vector and C is the regularization parameter.

Lafferty, J.D et. al. Conditional random fields: Probabilistic models for segmenting and labeling sequence data, ICML 2001

Classical NLP: CRF applied to Ner4Opt

Input → Tokens → Feature Extraction → CRF → OBIE Tags



	x1	x2	x3	x4	x5	x6
INPUT	I	want	to	maximize	my	profit
FEATURE TRASFORMATION	PRON	VERB	PART	VERB	PRON	NOUN
	True	True	True	True	True	True
	nsubj	ROOT	aux	xcomp	poss	dobj
	O	O	O	O	O	O
	...	...	...	...	...	...

Conditional Random Field

	y1	y2	y3	y4	y5	y6
OUTPUT	O	O	O	B-OBJ_DIR	O	B-OBJ_NAME

❑ In NLP, feature extraction function explores linguistic properties of a token or a group of tokens

❑ **Grammatical features**: part-of-speech (pos) tagging, dependency parsing, etc.

❑ **Morphological features**: prefix, suffix and word shape, capitalized, numeric, etc.

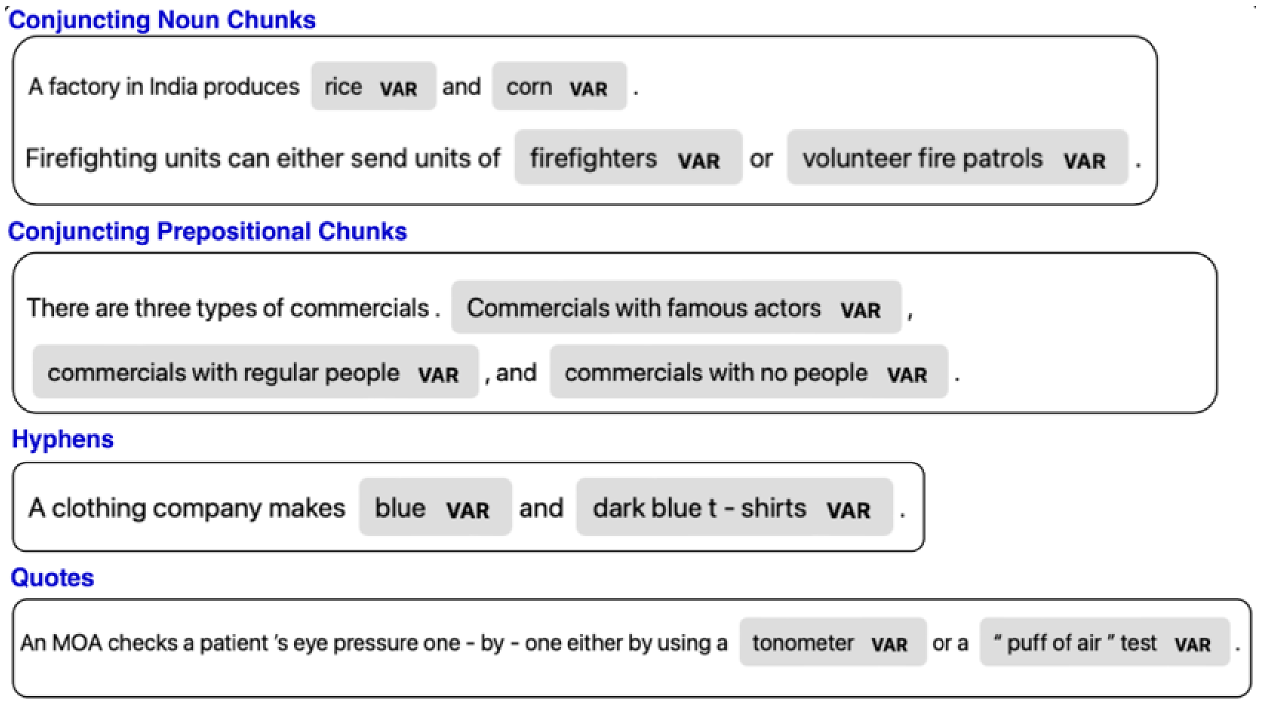
Ratinov, L., Roth, D.: Design challenges and misconceptions in NER, CoNLL, 2009

Feature Engineering for Optimization

Gazetteer and Syntactic features

- ❑ **Vocabulary features:** gazetteer features serve as lookup tables. Especially useful when the entity class has frequent keywords. maximize and minimize OBJ_DIR, at least and at most CONST_DIR

- ❑ **Syntactic features:** In linguistics, a conjunct is a group of tokens joined together by conjunction or punctuation. VAR and OBJ_NAME entities are associated with unique syntactical properties in the form of conjuncts, noun phrases and propositional phrases, etc.



Conjuncting Noun Chunks

A factory in India produces rice VAR and corn VAR .

Firefighting units can either send units of firefighters VAR or volunteer fire patrols VAR .

Conjuncting Prepositional Chunks

There are three types of commercials . Commercials with famous actors VAR ,

commercials with regular people VAR , and commercials with no people VAR .

Hyphens

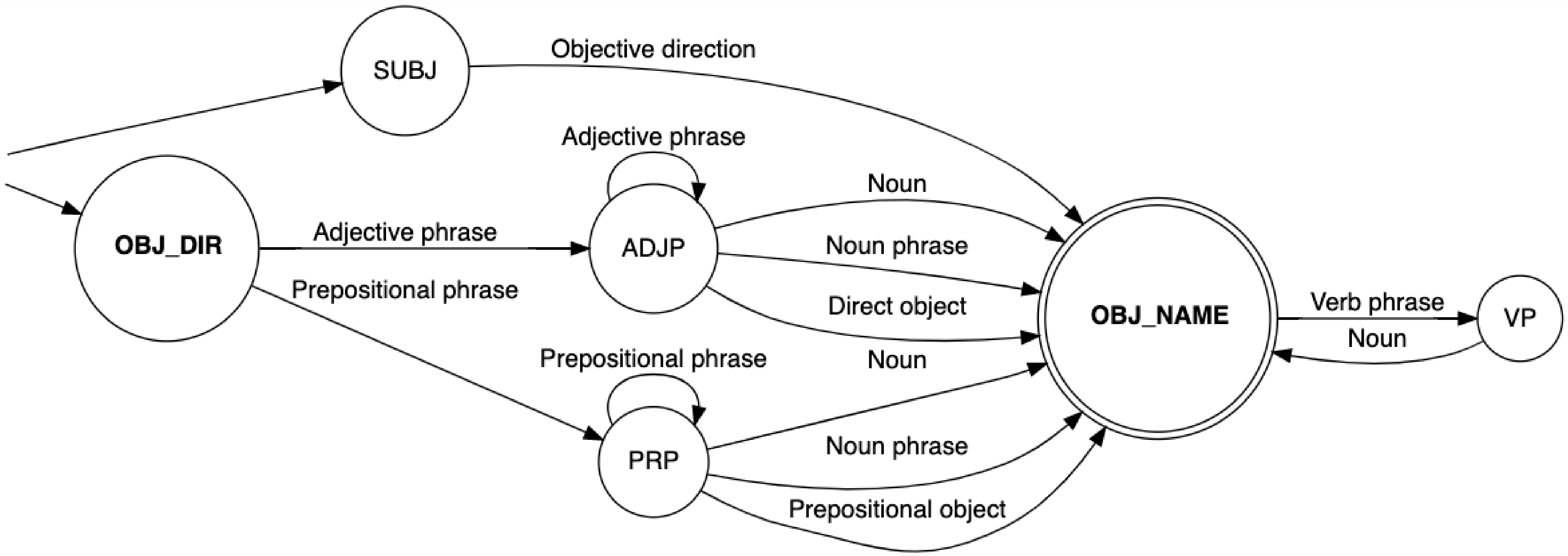
A clothing company makes blue VAR and dark blue t - shirts VAR .

Quotes

An MOA checks a patient 's eye pressure one - by - one either by using a tonometer VAR or a " puff of air " test VAR .

Regular Automaton for Name Extraction

Extracting the Objective Name



profit SUBJ to be maximized OBJ_DIR

maximize OBJ_DIR the total monthly ADJP profit NOUN

- ❑ **Contextual features:** extract left and right context of window size w
- ❑ **Constituent parsing**, word-frequency etc.

Modern NLP

Feature Engineering to Feature Learning

- ❑ So far, only considered classical methods based on feature extraction and manual feature engineering. This helps us establish a **baseline performance**.

- ❑ The challenger to this baseline is motivated by the **recent advances in NLP**, offering advantages over traditional techniques.

- ❑ Specifically, **deep neural networks** alleviate the need for manual feature extraction.

- ❑ Not only saves a significant amount of but offers more **robust behavior.**

- ❑ Moreover, the **nonlinearity in the activation functions** enables learning complex features and dependencies from the labeled training data.

Modern NLP

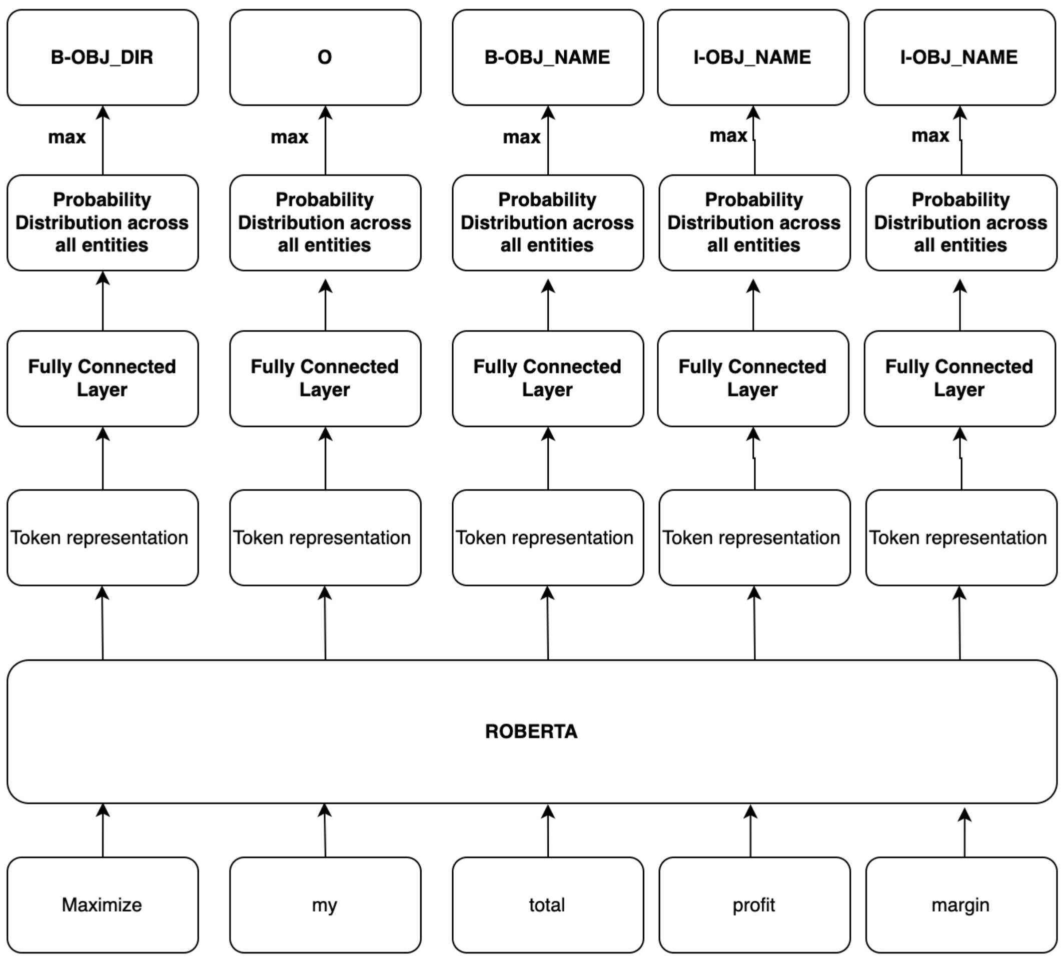
Feature Engineering to Feature Learning

- ❑ In practice, Ner4Opt problems require modeling **long-range text dependencies**.
- ❑ When operating on the long-range, **recurrent architectures** are known to struggle with vanishing and exploding gradients.
- ❑ As a remedy, most recent works rely on the **Transformers architecture** that solve the long-range problem by replacing the recurrent component with the attention mechanism.
- ❑ There are many variants of this architecture, and here, we consider distinct flavors based on **RoBERTa** to generate the feature embeddings.

Vaswani et. al.: Attention is all you need, NeurIPS 2017
Liu et. al.: Roberta: A robustly optimized bert pretraining approach, 2019

Formulate Ner4Opt as Token Classification

Use BERT-style models as encoders



Maximize my total profit margin

- ❑ Token classification problem with encoders
- ❑ Roberta embeddings with 1024 dimensions
- ❑ A fully-connected layer of size 1024 learns to map token level embeddings into named-entity-labels
- ❑ Followed by softmax activation function to output dimension of 1 x 13
- ❑ Minimize training loss with cross-entropy loss

Fine-Tuning with Optimization Corpora

Improving LLMs for domain-specific Ner4Opt

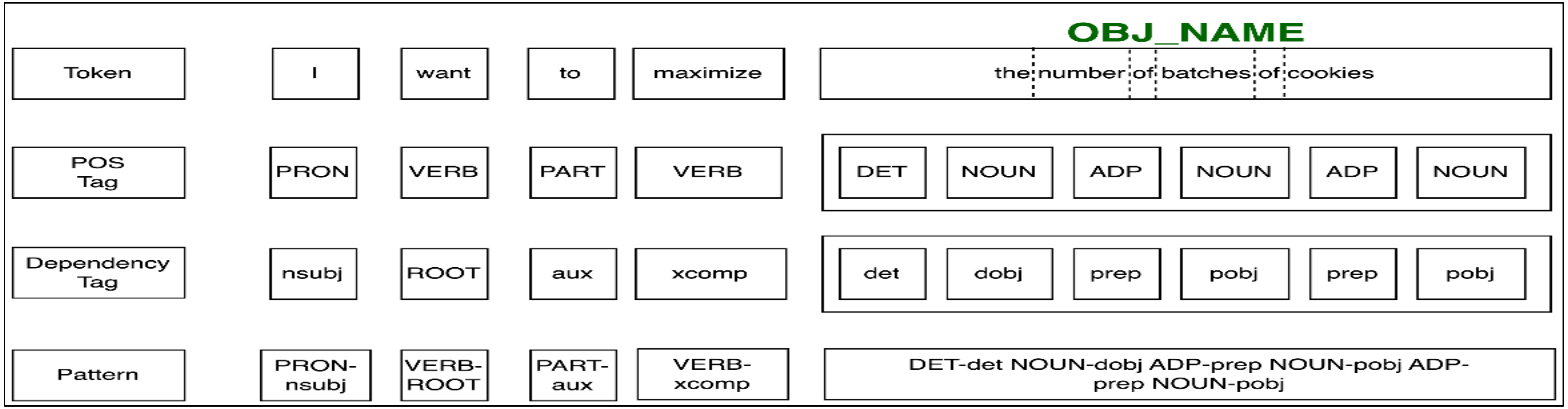
- ❑ LLMs, such as BERT, RoBERTa, GPT, are pretrained on **non-domain specific text** for good downstream performance on language-oriented tasks
- ❑ For domain specific tasks, performance can be improved using **domain specific corpora** to fine-tune pre-trained models
- ❑ Convex optimization, linear programming, game theory books, course notes on optimization from Open Optimization Platform
- ❑ Our work is among the first to fine-tune with optimization corpora using **Masked Language Modelling** with 15% words are random, replace 80% with MAST token, 10% with random, and the remaining 10% with the original word

Howard J., Ruder, S.: Universal language model fine-tuning for text classification, 2018

Data Augmentation

Up-Sampling Infrequent Patterns

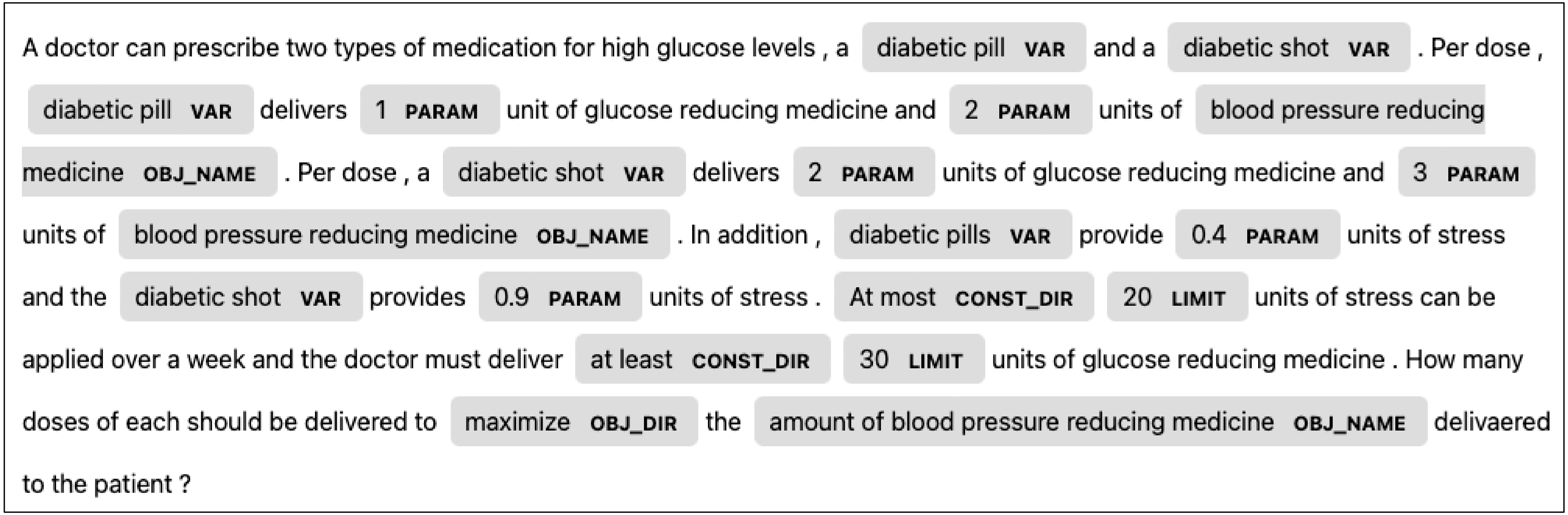
- ❑ Distribution of classes is balanced. However, **lexical features** exhibit popular traits with infrequent features
- ❑ **Example:** objective is maximize/minimize but sometimes as adjective, cost to be minimal
- ❑ Challenge is to **find infrequent feature** without manual inspection: Combine POS+DEP Tags



					OBJ_NAME					
Token	I	want	to	maximize	the	number	of	batches	of	cookies
POS Tag	PRON	VERB	PART	VERB	DET	NOUN	ADP	NOUN	ADP	NOUN
Dependency Tag	nsubj	ROOT	aux	xcomp	det	dobj	prep	pobj	prep	pobj
Pattern	PRON-nsubj	VERB-ROOT	PART-aux	VERB-xcomp	DET-det NOUN-dobj ADP-prep NOUN-pobj ADP-prep NOUN-pobj					

Dealing with Disambiguation

Is it a variable or objective variable?



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 ?

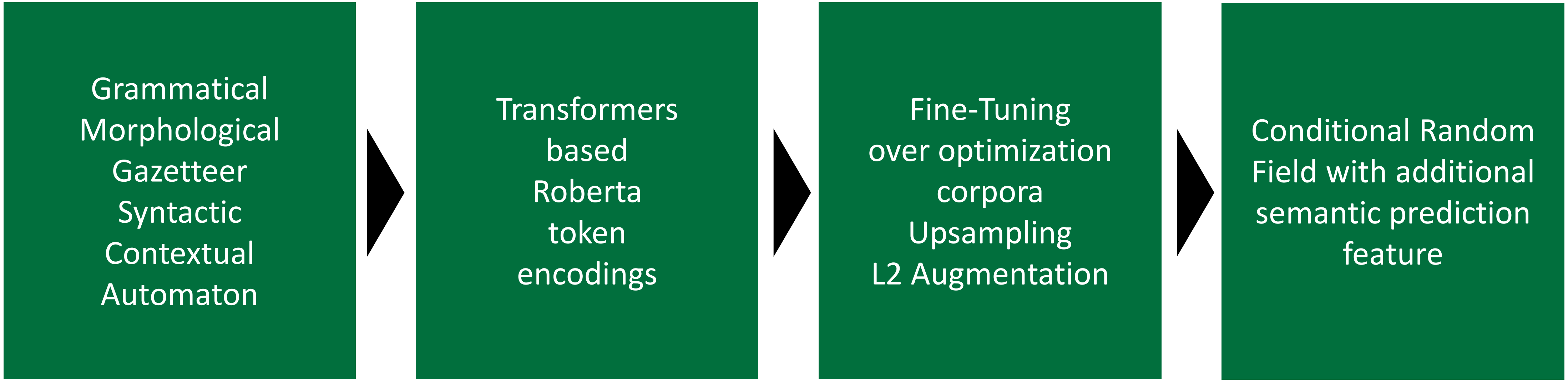
Apply L2 Augmentation

Hybrid Modeling

Feature Engineering + Feature Learning

Feature engineering might be brittle but helps build apriori information

Feature learning brings semantic representations but struggles with long-range dependency



Grammatical
Morphological
Gazetteer
Syntactic
Contextual
Automaton

Transformers based Roberta token encodings

Fine-Tuning over optimization corpora
Upsampling
L2 Augmentation

Conditional Random Field with additional semantic prediction feature

Numerical Results

Effectiveness of the Ner4Opt Solution

Post-mortem and ChatGPT

Experiments

Research Questions

1. What is the baseline classical performance and does feature engineering help?

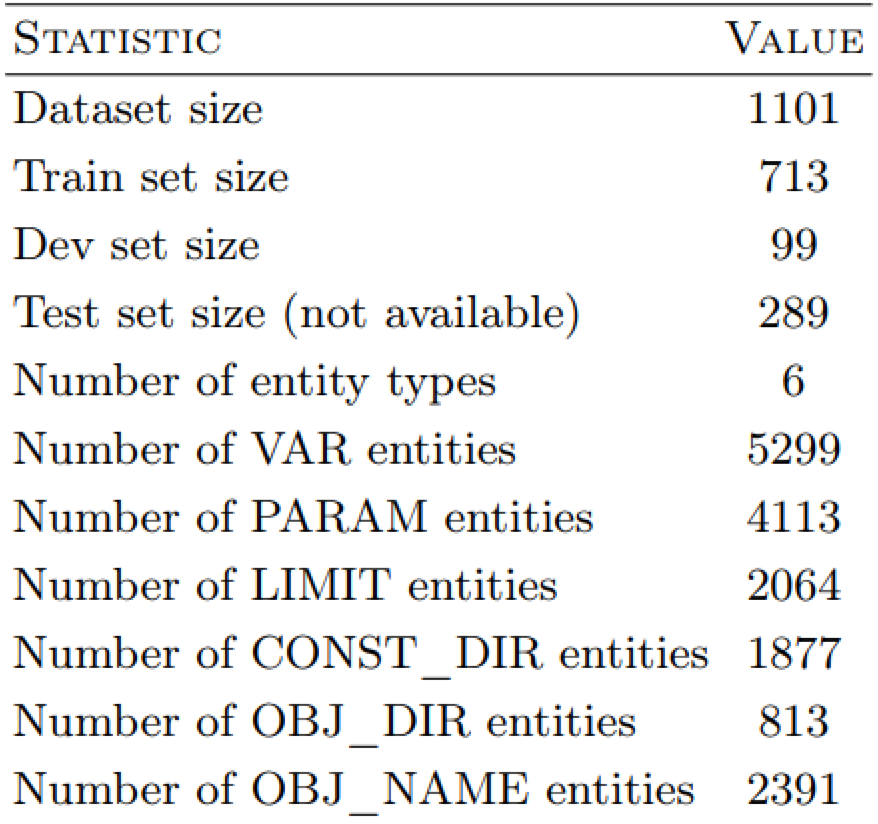
2. How do modern NLP perform, do we improve over the state-of-the-art?

3. Does the hybrid model perform better than its counterparts in isolation?

4. Where does Ner4Opt fail and how about ChatGPT?

Experiments

Data & Experimental Setup



STATISTIC	VALUE
Dataset size	1101
Train set size	713
Dev set size	99
Test set size (not available)	289
Number of entity types	6
Number of VAR entities	5299
Number of PARAM entities	4113
Number of LIMIT entities	2064
Number of CONST_DIR entities	1877
Number of OBJ_DIR entities	813
Number of OBJ_NAME entities	2391

□ **Optimization word problems** released as part of NeurIPS’22 NL4Opt Workshop. 1101 optimization instances with annotated entities. 15 annotators

□ **Source Domain:** advertising, investment, sales

□ **Target Domain:** production, science, transportation

Training dataset only comes from Source domain

Test and Dev set comes from Source and Target

□ **Libraries**: HuggingFace transformers, Simple transformers, SpaCy, sklearn-crf

□ Limited hyperparameter tuning to avoid over-fitting

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

Experiments

Comparisons

Classical
Classical+

Classical based on grammatical and morphological features, plus with hand-crafted gazetteer, syntactic, and contextual features.

XLM-RB
XLM-RL

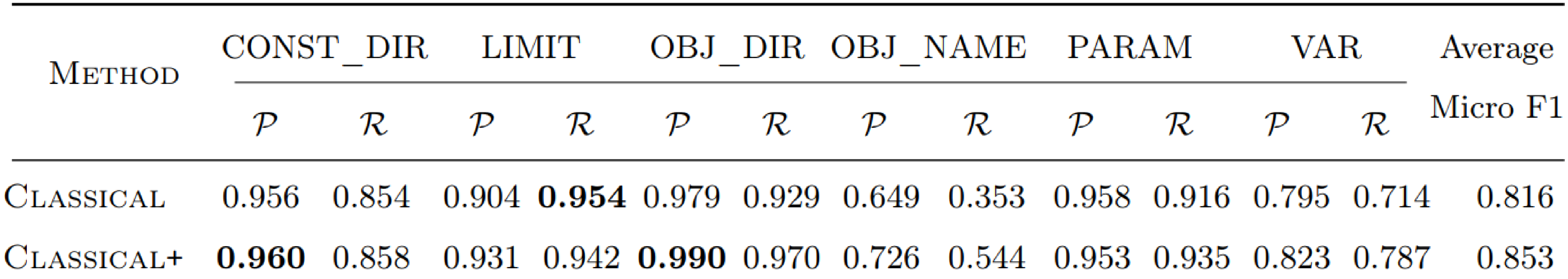
The state-of-the-art method based on XLM-Roberta Base and its Large variant

XLM-RL+
Hybrid

Our optimization fined tuned XML-RL+ and
Hybrid method with feature engineering and learning

Experiments

Q1: What is baseline classical performance and does feature engineering help?



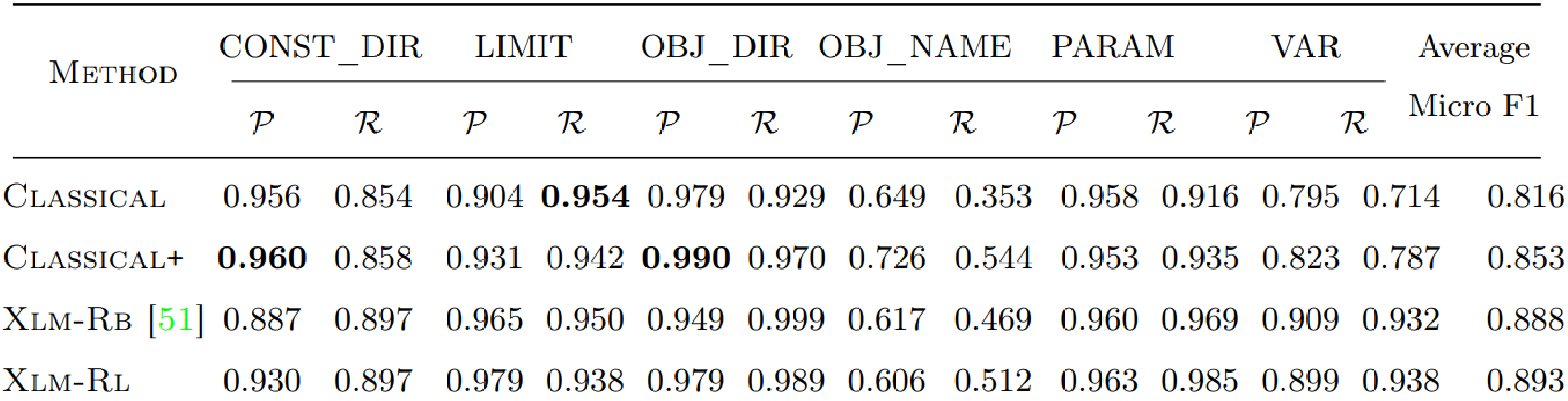
Method	CONST_DIR		LIMIT		OBJ_DIR		OBJ_NAME		PARAM		VAR		Average
	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{P}$	$\mathcal{R}$	Micro F1
Classical	0.956	0.854	0.904	**0.954**	0.979	0.929	0.649	0.353	0.958	0.916	0.795	0.714	0.816
Classical+	**0.960**	0.858	0.931	0.942	**0.990**	0.970	0.726	0.544	0.953	0.935	0.823	0.787	0.853

$$F1 = \frac{2 * P * R}{P + R}$$

- **Classical+ jumps from 0.81 to 0.85** by hand-crafted gazetteer, syntactic and contextual features
- Feature engineering focus on CONST_DIR and OBJ_DIR which improves
- Classical reports 0.90+ P and 0.85+ R except OBJ_NAME and VAR (ambiguity and long range)

Experiments

Q2: What is the performance of Modern NLP?

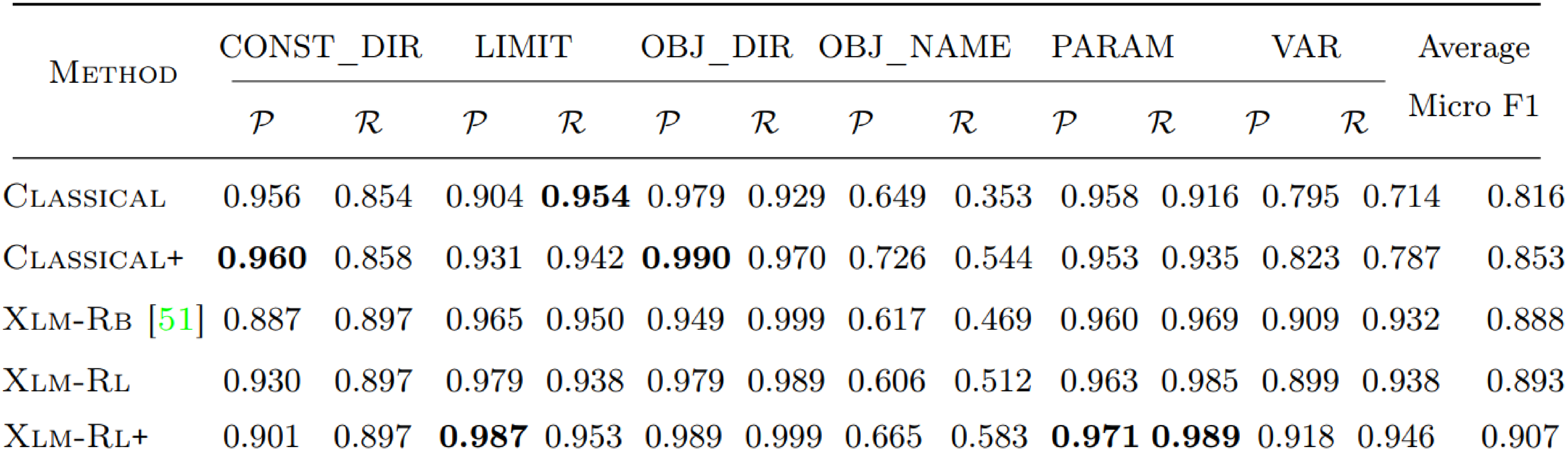


Method	CONST_DIR		LIMIT		OBJ_DIR		OBJ_NAME		PARAM		VAR		Average
	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{P}$	$\mathcal{R}$	Micro F1
Classical	0.956	0.854	0.904	**0.954**	0.979	0.929	0.649	0.353	0.958	0.916	0.795	0.714	0.816
Classical+	**0.960**	0.858	0.931	0.942	**0.990**	0.970	0.726	0.544	0.953	0.935	0.823	0.787	0.853
Xlm-Rb [51]	0.887	0.897	0.965	0.950	0.949	0.999	0.617	0.469	0.960	0.969	0.909	0.932	0.888
Xlm-Rl	0.930	0.897	0.979	0.938	0.979	0.989	0.606	0.512	0.963	0.985	0.899	0.938	0.893

- **Modern NLP improves over the Classical from 0.81 to 0.88**
- Slight gains when switching to larger models
- Multilingual training of XLM is not beneficial for Ner4Opt (compared to RoBERTa)

Experiments

Q3: What the impact of optimization fine-tuning?

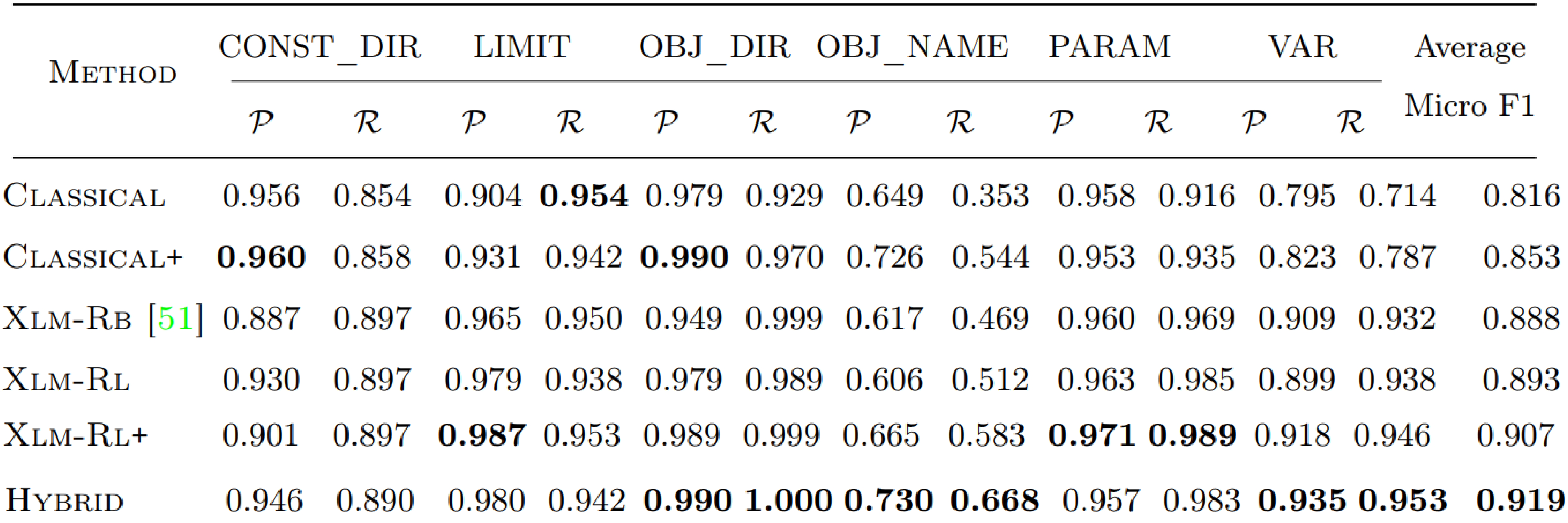


Method	CONST_DIR		LIMIT		OBJ_DIR		OBJ_NAME		PARAM		VAR		Average Micro F1
	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{P}$	$\mathcal{R}$	
Classical	0.956	0.854	0.904	**0.954**	0.979	0.929	0.649	0.353	0.958	0.916	0.795	0.714	0.816
Classical+	**0.960**	0.858	0.931	0.942	**0.990**	0.970	0.726	0.544	0.953	0.935	0.823	0.787	0.853
Xlm-Rb [51]	0.887	0.897	0.965	0.950	0.949	0.999	0.617	0.469	0.960	0.969	0.909	0.932	0.888
Xlm-Rl	0.930	0.897	0.979	0.938	0.979	0.989	0.606	0.512	0.963	0.985	0.899	0.938	0.893
Xlm-Rl+	0.901	0.897	**0.987**	0.953	0.989	0.999	0.665	0.583	**0.971**	**0.989**	0.918	0.946	0.907

- **Our XLM-RL+ improves with optimization fine-tuning**
- Encouraging result with only a few textbooks over large training corpora
- While higher average score, modern NLP does not improve P/R in every class

Experiments

Q3: What is the performance of Hybrid solutions?

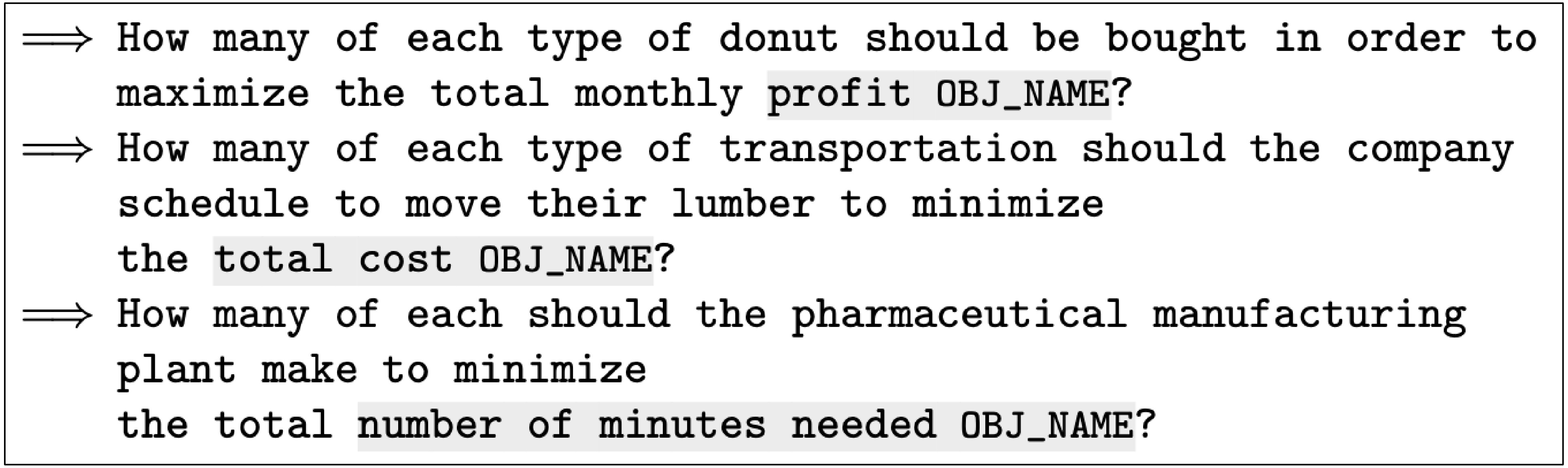


Method	CONST_DIR		LIMIT		OBJ_DIR		OBJ_NAME		PARAM		VAR		Average
	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{P}$	$\mathcal{R}$	Micro F1
Classical	0.956	0.854	0.904	**0.954**	0.979	0.929	0.649	0.353	0.958	0.916	0.795	0.714	0.816
Classical+	**0.960**	0.858	0.931	0.942	**0.990**	0.970	0.726	0.544	0.953	0.935	0.823	0.787	0.853
Xlm-Rb [51]	0.887	0.897	0.965	0.950	0.949	0.999	0.617	0.469	0.960	0.969	0.909	0.932	0.888
Xlm-Rl	0.930	0.897	0.979	0.938	0.979	0.989	0.606	0.512	0.963	0.985	0.899	0.938	0.893
Xlm-Rl+	0.901	0.897	**0.987**	0.953	0.989	0.999	0.665	0.583	**0.971**	**0.989**	0.918	0.946	0.907
Hybrid	0.946	0.890	0.980	0.942	**0.990**	**1.000**	**0.730**	**0.668**	0.957	0.983	**0.935**	**0.953**	**0.919**

- **Our Hybrid achieves the best performance 0.919**
- Best performance in most / hardest classes

Post-Mortem

Q4: Where does Ner4Opt solution fails?



$\Longrightarrow$ How many of each type of donut should be bought in order to maximize the total monthly profit OBJ_NAME?

$\Longrightarrow$ How many of each type of transportation should the company schedule to move their lumber to minimize
the total cost OBJ_NAME?

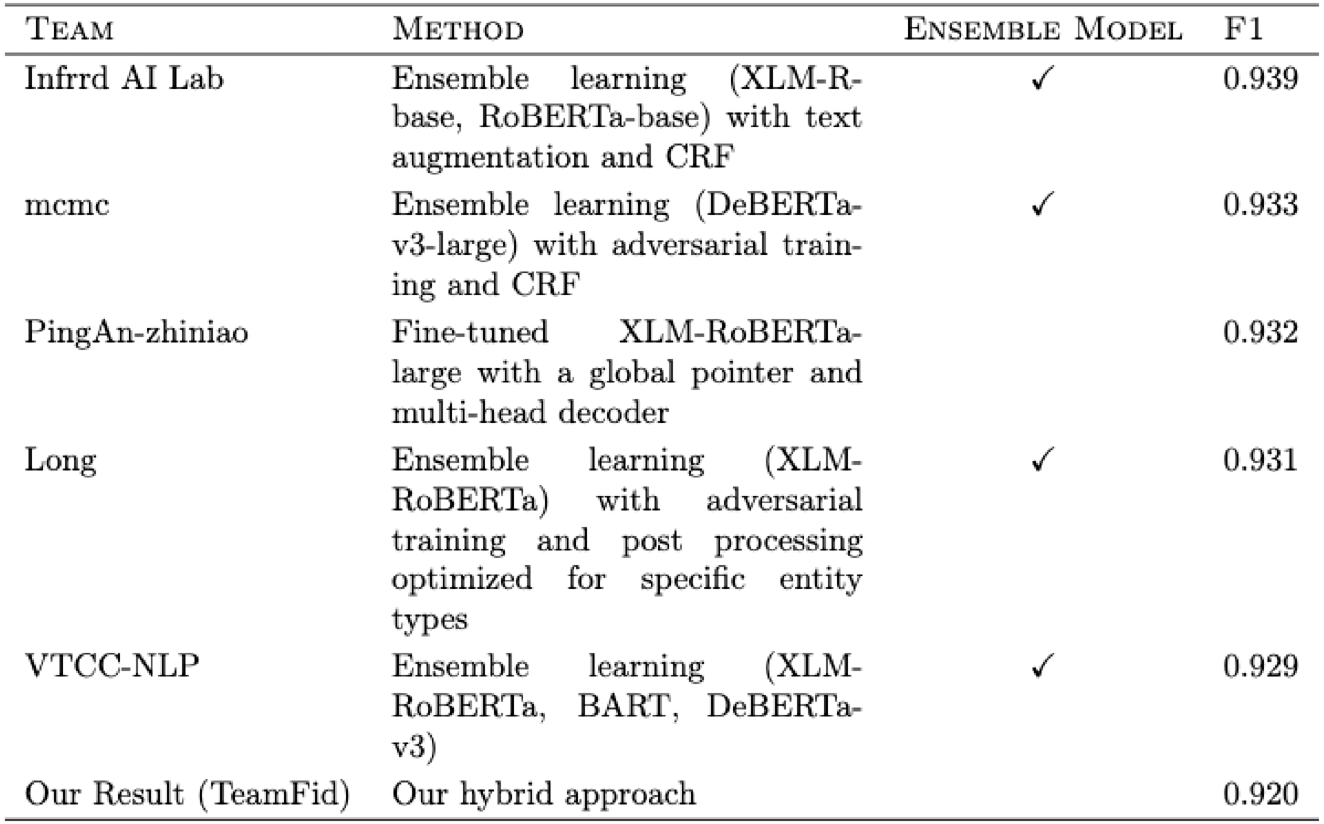
$\Longrightarrow$ How many of each should the pharmaceutical manufacturing plant make to minimize
the total number of minutes needed OBJ_NAME?

- Conflicting token span in annotation entities between training and dev sets
- Similar inconsistencies for all classes. Even human annotators cannot agree
- Aleatoric uncertainty stemming from data, difficult to address

Kadioglu et. al. Modeling uncertainty to improve personalized recommendations via Bayesian deep learning, JDSA 2021

Ner4Opt as part of NL4OPT

NL4OPT Competition results (not exhaustive)

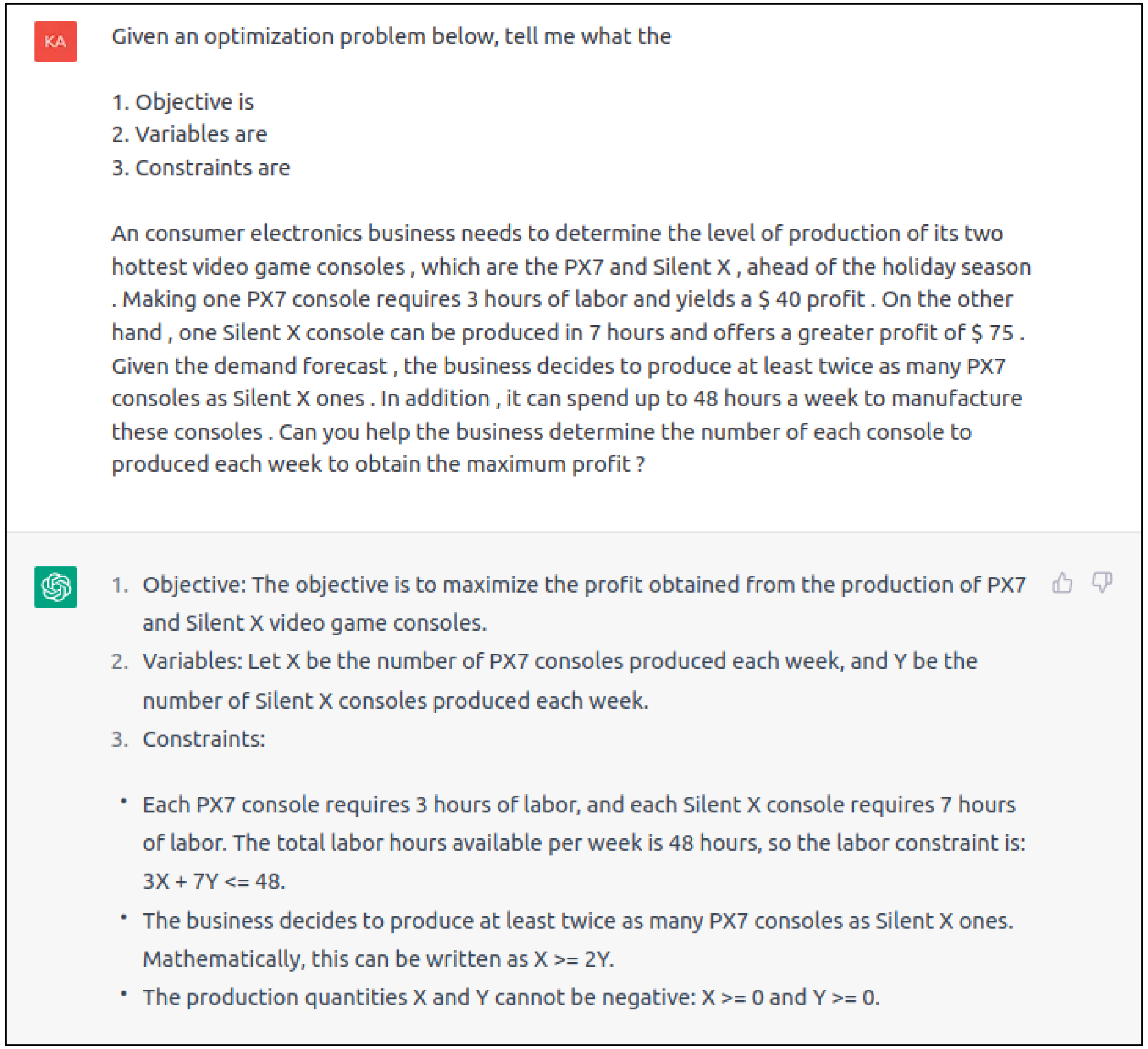


Team	Method	Ensemble Model	F1
Infrrd AI Lab	Ensemble learning (XLM-R-base, RoBERTa-base) with text augmentation and CRF	✓	0.939
mcmc	Ensemble learning (DeBERTa-v3-large) with adversarial training and CRF	✓	0.933
PingAn-zhiniao	Fine-tuned XLM-RoBERTa-large with a global pointer and multi-head decoder		0.932
Long	Ensemble learning (XLM-RoBERTa) with adversarial training and post processing optimized for specific entity types	✓	0.931
VTCC-NLP	Ensemble learning (XLM-RoBERTa, BART, DeBERTa-v3)	✓	0.929
Our Result (TeamFid)	Our hybrid approach		0.920

- ❑ Combinations of **CRF** and **LLM**
- ❑ LLMs **RoBERTa** and **DeBERTa**
- ❑ Top performance via **Ensembles**
- ❑ Consider **inference time** (interactivity)

Post-Mortem

Q4: Why not just use ChatGPT?



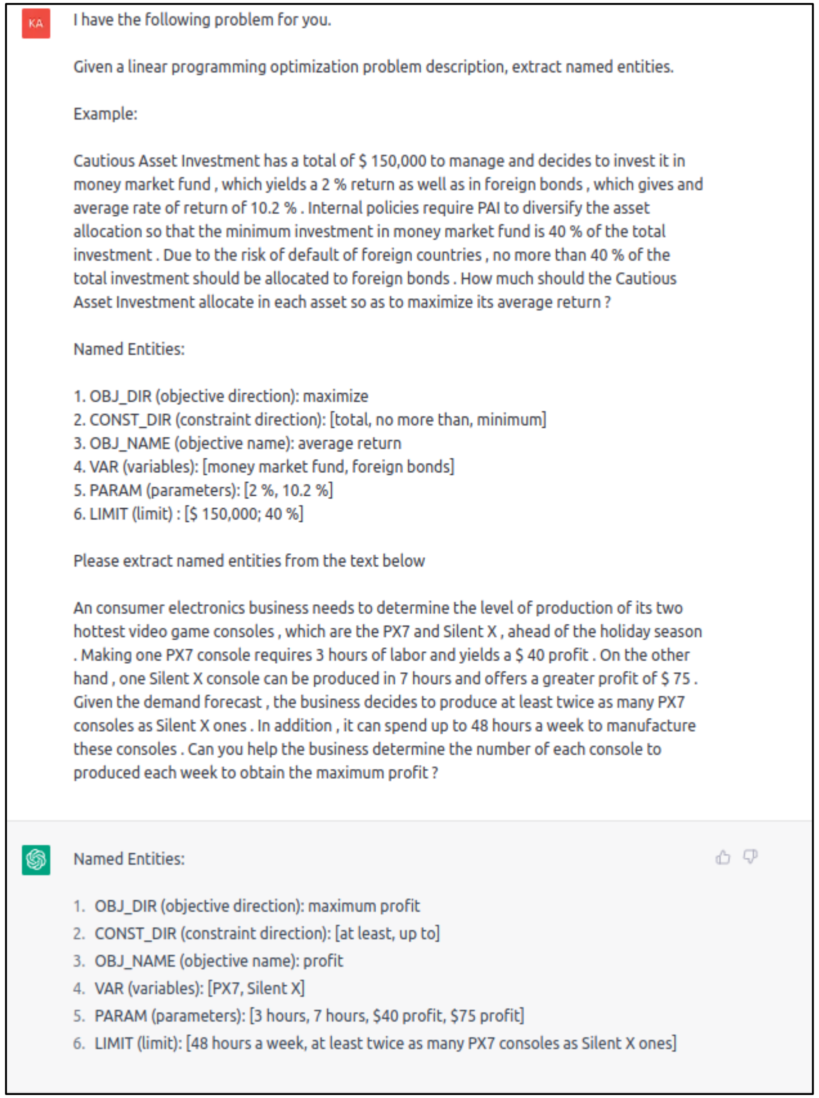
Given an optimization problem below, tell me what the

1. Objective is
2. Variables are
3. Constraints are

An consumer electronics business needs to determine the level of production of its two hottest video game consoles , which are the PX7 and Silent X , ahead of the holiday season . Making one PX7 console requires 3 hours of labor and yields a $ 40 profit . On the other hand , one Silent X console can be produced in 7 hours and offers a greater profit of $ 75 . Given the demand forecast , the business decides to produce at least twice as many PX7 consoles as Silent X ones . In addition , it can spend up to 48 hours a week to manufacture these consoles . Can you help the business determine the number of each console to produced each week to obtain the maximum profit ?

1. Objective: The objective is to maximize the profit obtained from the production of PX7 and Silent X video game consoles.
2. Variables: Let X be the number of PX7 consoles produced each week, and Y be the number of Silent X consoles produced each week.
3. Constraints:

- Each PX7 console requires 3 hours of labor, and each Silent X console requires 7 hours of labor. The total labor hours available per week is 48 hours, so the labor constraint is: 3X + 7Y <= 48.
- The business decides to produce at least twice as many PX7 consoles as Silent X ones. Mathematically, this can be written as X >= 2Y.
- The production quantities X and Y cannot be negative: X >= 0 and Y >= 0.



I have the following problem for you.

Given a linear programming optimization problem description, extract named entities.

Example:

Cautious Asset Investment has a total of $ 150,000 to manage and decides to invest it in money market fund , which yields a 2 % return as well as in foreign bonds , which gives and average rate of return of 10.2 % . Internal policies require PAI to diversify the asset allocation so that the minimum investment in money market fund is 40 % of the total investment . Due to the risk of default of foreign countries , no more than 40 % of the total investment should be allocated to foreign bonds . How much should the Cautious Asset Investment allocate in each asset so as to maximize its average return ?

Named Entities:

1. OBJ_DIR (objective direction): maximize
2. CONST_DIR (constraint direction): [total, no more than, minimum]
3. OBJ_NAME (objective name): average return
4. VAR (variables): [money market fund, foreign bonds]
5. PARAM (parameters): [2 %, 10.2 %]
6. LIMIT (limit) : [$ 150,000; 40 %]

Please extract named entities from the text below

An consumer electronics business needs to determine the level of production of its two hottest video game consoles , which are the PX7 and Silent X , ahead of the holiday season . Making one PX7 console requires 3 hours of labor and yields a $ 40 profit . On the other hand , one Silent X console can be produced in 7 hours and offers a greater profit of $ 75 . Given the demand forecast , the business decides to produce at least twice as many PX7 consoles as Silent X ones . In addition , it can spend up to 48 hours a week to manufacture these consoles . Can you help the business determine the number of each console to produced each week to obtain the maximum profit ?

Named Entities:

1. OBJ_DIR (objective direction): maximum profit
2. CONST_DIR (constraint direction): [at least, up to]
3. OBJ_NAME (objective name): profit
4. VAR (variables): [PX7, Silent X]
5. PARAM (parameters): [3 hours, 7 hours, $40 profit, $75 profit]
6. LIMIT (limit): [48 hours a week, at least twice as many PX7 consoles as Silent X ones]

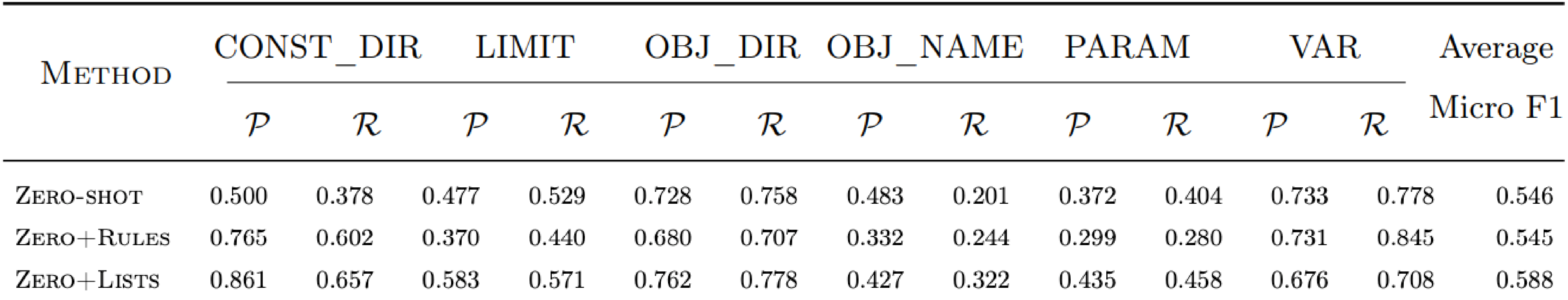
Ner4Opt with GPT4.0

Prompting Strategies

- ❑ **Zero-Shot:** baseline that specifies types of entities to extract and output format
  - ○ **Zero-Shot + Rules:** Incorporates soft rules for each entity type
    Example: Objective Direction is generally a verb (e.g., maximize, minimize).
    Sometimes Objective Direction is also an adjective (e.g., I want the cost to be minimal).
  - ○ **Zero-Shot + Lists:** Adapts Zero-Shot + Rules prompt in a more structured format
    Rules are captured as a numbered list instead of a paragraph
- ❑ **Few-Shot-k:** Extends zero-shot strategies by incorporating actual examples of the optimization problems and their corresponding results

Ner4Opt with GPT4.0

Numerical Results: Zero Shot

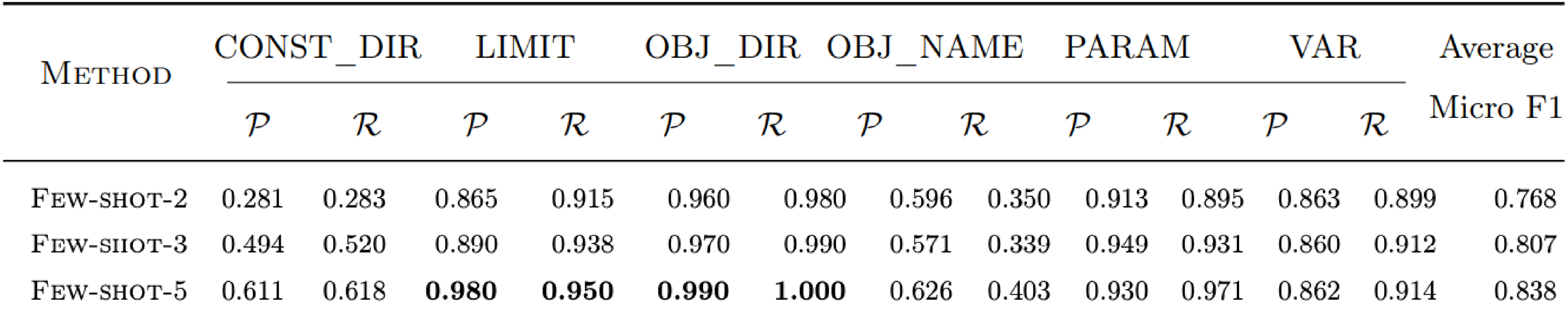


METHOD	CONST_DIR		LIMIT		OBJ_DIR		OBJ_NAME		PARAM		VAR		Average
	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{P}$	$\mathcal{R}$	Micro F1
ZERO-SHOT	0.500	0.378	0.477	0.529	0.728	0.758	0.483	0.201	0.372	0.404	0.733	0.778	0.546
ZERO+RULES	0.765	0.602	0.370	0.440	0.680	0.707	0.332	0.244	0.299	0.280	0.731	0.845	0.545
ZERO+LISTS	0.861	0.657	0.583	0.571	0.762	0.778	0.427	0.322	0.435	0.458	0.676	0.708	0.588

- Vanilla **Zero-Shot** achieves an average micro score F1 of **0.546**, severely under-performing even our Classical baseline (0.816)
- Underperform on **difficult entities** such as OBJ_NAME and VAR, but also on **easier entities** such as, LIMIT and PARAM
- Introducing **Rules & Lists** only **slightly help** the score to **0.588**
- Out-of-the-box zero-shot LLMs cannot yet address Ner4Opt

Ner4Opt with GPT4.0

Numerical Results: In-context Learning

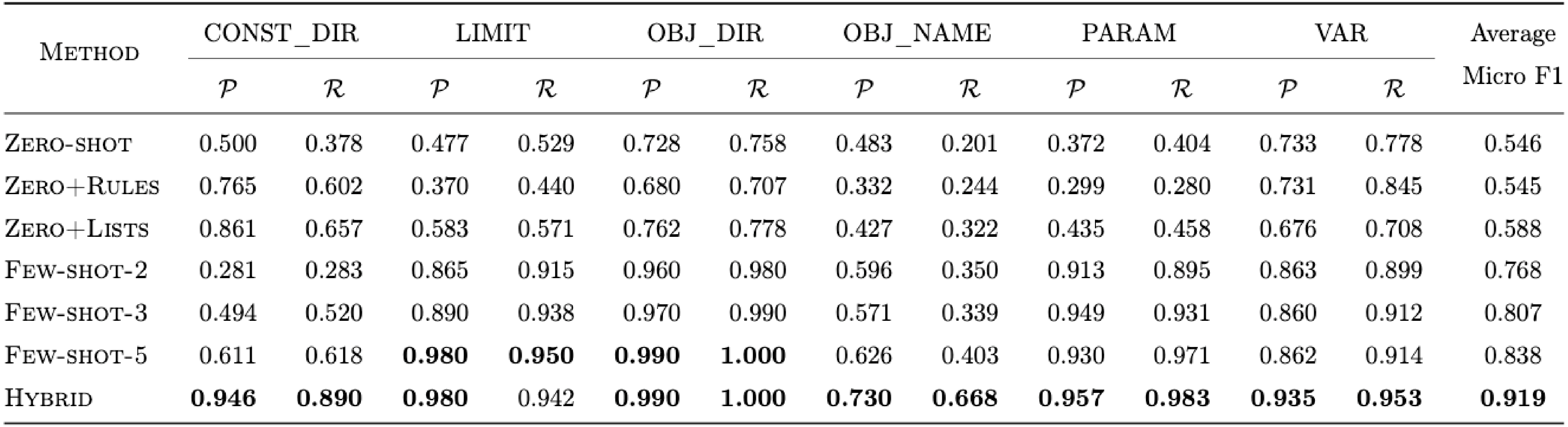


Method	CONST_DIR		LIMIT		OBJ_DIR		OBJ_NAME		PARAM		VAR		Average
	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{P}$	$\mathcal{R}$	Micro F1
Few-shot-2	0.281	0.283	0.865	0.915	0.960	0.980	0.596	0.350	0.913	0.895	0.863	0.899	0.768
Few-shot-3	0.494	0.520	0.890	0.938	0.970	0.990	0.571	0.339	0.949	0.931	0.860	0.912	0.807
Few-shot-5	0.611	0.618	**0.980**	**0.950**	**0.990**	**1.000**	0.626	0.403	0.930	0.971	0.862	0.914	0.838

- Considerable performance improvement with in-context learning up to 0.838
- Caveat: Need diversity in examples

Ner4Opt with GPT4.0

Numerical Results: In-context Learning



Method	CONST_DIR		LIMIT		OBJ_DIR		OBJ_NAME		PARAM		VAR		Average Micro F1
	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{P}$	$\mathcal{R}$	
Zero-shot	0.500	0.378	0.477	0.529	0.728	0.758	0.483	0.201	0.372	0.404	0.733	0.778	0.546
Zero+Rules	0.765	0.602	0.370	0.440	0.680	0.707	0.332	0.244	0.299	0.280	0.731	0.845	0.545
Zero+Lists	0.861	0.657	0.583	0.571	0.762	0.778	0.427	0.322	0.435	0.458	0.676	0.708	0.588
Few-shot-2	0.281	0.283	0.865	0.915	0.960	0.980	0.596	0.350	0.913	0.895	0.863	0.899	0.768
Few-shot-3	0.494	0.520	0.890	0.938	0.970	0.990	0.571	0.339	0.949	0.931	0.860	0.912	0.807
Few-shot-5	0.611	0.618	**0.980**	**0.950**	**0.990**	**1.000**	0.626	0.403	0.930	0.971	0.862	0.914	0.838
Hybrid	**0.946**	**0.890**	**0.980**	0.942	**0.990**	**1.000**	**0.730**	**0.668**	**0.957**	**0.983**	**0.935**	**0.953**	**0.919**

- LLM performance falls short of our best **Hybrid Model** (0.919)
- This again highlights the **inherent complexity** of Ner4Opt
- **Disclaimer:** general-purpose LLM vs. fully supervised approach

Post-Mortem

Where does GPT4.0 fail?

❑ In-context learning requires **careful selection with unique examples** for inclusion in the prompt

❑ Selection has an **adverse affect** on unseen patterns

- ○ If examples showcase objective with nouns, the model struggles to correctly identify objective names in new examples where the span includes a verb phrase qualifying the noun

❑ Incorrectly identifies conjuncting person names as **VARs** (ex: John and Abraham)

❑ Over predicts **OBJ_NAME** and **CONST_DIR** spans by including neighboring tokens

- ○ In the phrase *“.. find the minimum cost spicy paste that can be made.. “*
  GPT-4 predicts "*cost spicy paste*" as OBJ_NAME
  Ground truth only tags *"cost"*
- ○ In the sentence *“.. A train can carry at most 500 passengers.. “*
  GPT-4 predicts *"can carry at most"* as the CONST_DIR entity
  Ground truth only tags *"at most*

Concluding Remarks & Future Directions

LLMs for Optimization: Toward Automated Modelling Assistants

- ❑ Rich literature for integrating ML + Opt but only recent work in **LLM + Opt**
- ❑ Assistants are immediately relevant for Opt but also challenging tasks for NLP (counter-intuitive)
- ❑ Important to study **building blocks** in depth. **Generalization** to new domains is possible
- ❑ **HCI questions** when non-technical users are empowered with Opt over text or audio
- ❑ **Call-to-Action:** How to break the low annotated data regime to realize LLM-style success

References

Research & Open-Source Software

❑ [PTHG@CP'23] Holy Grail 2.0: From Natural Language to Constraint Models	
❑ [NeurIPS'22, CPAIOR'23] Ner4Opt	https://github.com/skadio/ner4opt
❑ Ner4Opt Demo	https://huggingface.co/spaces/skadio/Ner4Opt
❑ ChatOpt Demo	https://chatopt.cs.kuleuven.be
❑ [NeurIPS'22] NL4Opt Challenge	https://nl4opt.github.io
❑ Logic Grid Puzzles	https://github.com/jelgun/LGPSolver
❑ CPMpy: CP and Modeling in Python	https://cpmpy.readthedocs.io

pip install ner4opt

skadio.github.io