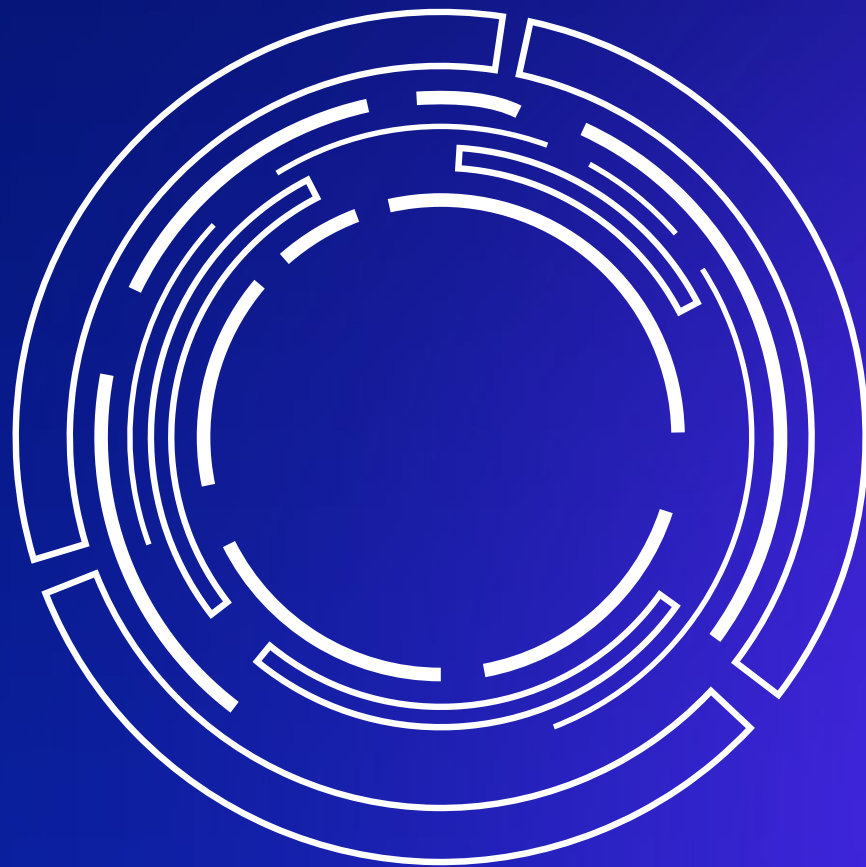


# Combining Constraint Programming Reasoning with Large Language Model Predictions

*Florian Régin, Elisabetta De Maria, and Alexandre Bonlarron*  
*Université Côte d'Azur, I3S, France*



Presented by Florian Régin



# INTRODUCTION

Goal of this presentation:

Showcase a new combination of Constraint Programming (CP) and Machine Learning (ML) for text generation under constraints.

# INTRODUCTION



LLM generates the domains  
of a CSP

# TEXT GENERATION UNDER CONSTRAINTS

Problem from Natural Language Processing (NLP).

- Generate sentences that have meaning and satisfy external constraints.

# TEXT GENERATION UNDER CONSTRAINTS

Problem from Natural Language Processing (NLP).

- Generate sentences that have meaning and satisfy external constraints.
- Constraints can be on the number of words/characters, the size of words, that the sentence contains/does not contain specific characters/words, etc...

# TEXT GENERATION UNDER CONSTRAINTS

Problem from Natural Language Processing (NLP).

- Generate sentences that have meaning and satisfy external constraints.
- Constraints can be on the number of words/characters, the size of words, that the sentence contains/does not contain specific characters/words, etc...

Problem came after the emergence of Chatbots like ChatGPT that do really well on the problem of Free sentence generation.

# TEXT GENERATION UNDER CONSTRAINTS

## ChatGPT: Meaning constraint

Summarize the next sentence in 2 words: "If thou would kindly cease movement of thine vocal strings, It would garner quite a pleasant continuation of mine day."



Be quiet.

# TEXT GENERATION UNDER CONSTRAINTS

## ChatGPT: Structural constraint

Create a sentence that contains 100 to 110 characters

Certainly! Here's a sentence with 105 characters:

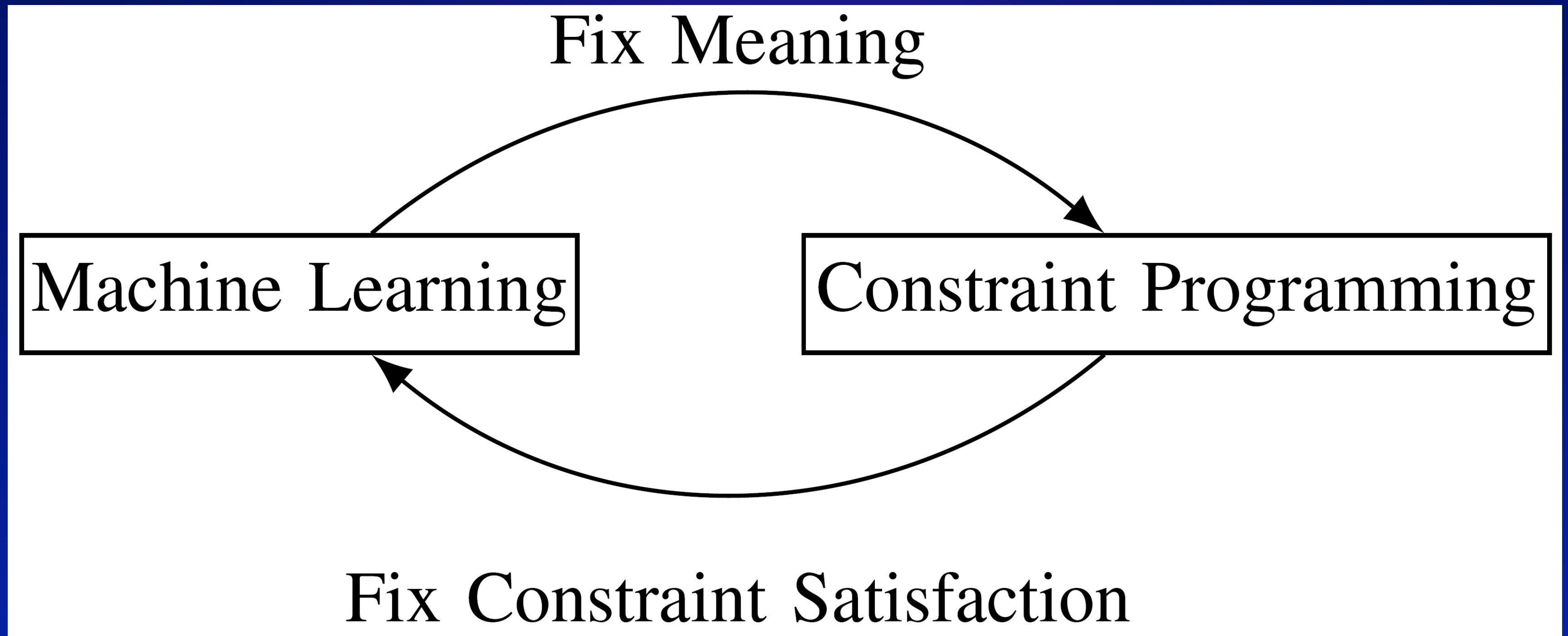
"The old lighthouse, weathered by time and sea, stood steadfast against the stormy night, guiding ships to safety."

Wrong answer:

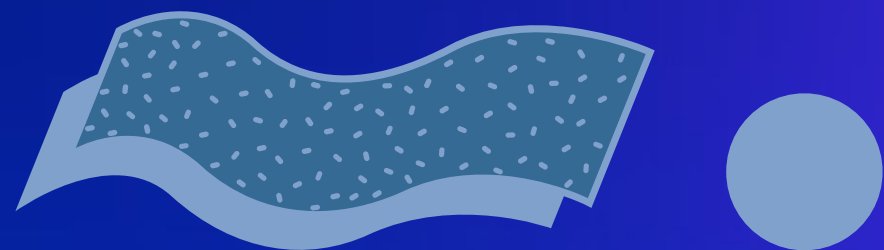
The sentence is 113 characters

# COMBINING CONSTRAINT PROGRAMMING AND MACHINE LEARNING

## Text Generation Under Constraints



# PLAN



01

Generative CP (GenCP)

02

Combining GenCP and  
ML (GenCPML)

03

Experimental Results

04

Other Results

# CONSTRAINT PROGRAMMING (CP)<sup>11</sup>

Paradigm for solving combinatorial problems.

Problem

**Model:**

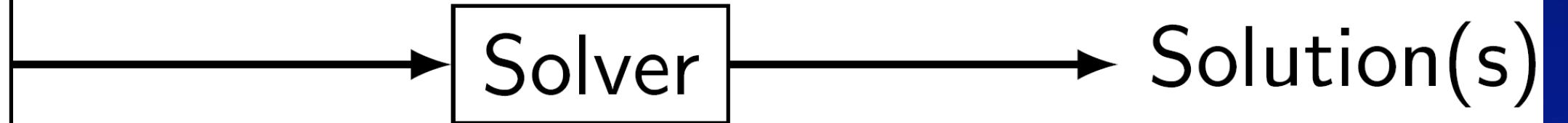
Variables  $\mathcal{X} = \{x_1, \dots, x_n\}$

Domains  $\mathcal{D} = \{d_1, \dots, d_n\}$

Constraints  $\mathcal{C} = \{c_1, \dots, c_m\}$

Solver

Solution(s)



# CONSTRAINT PROGRAMMING (CP)<sup>12</sup>

## Text Generation Under Constraint

- Variables represent words.
- 2024 Oxford English Dictionary contains  $\approx 171,476$  words.
- Huge number of combinations.
- Can cause memory problems.

# GENERATIVE CONSTRAINT PROGRAMMING

## Formal Definition

Problem

**Model:**

Variables  $\mathcal{X} = \emptyset$

Domains  $\mathcal{D} = \emptyset$

Constraints  $\mathcal{C} = \emptyset$

**Functions:**

Generative Functions

Boolean Functions

Solver

Solution(s)

# GENERATIVE CONSTRAINT PROGRAMMING

## Core Ideas

- Variables are generated one after the other.

# GENERATIVE CONSTRAINT PROGRAMMING

## Core Ideas

- Variables are generated one after the other.
- Inner workings:
  1. A variable is generated.

# GENERATIVE CONSTRAINT PROGRAMMING

## Core Ideas

- Variables are generated one after the other.
- Inner workings:
  1. A variable is generated.
  2. The domain of this variable is generated. (here by an LLM)

# GENERATIVE CONSTRAINT PROGRAMMING

## Core Ideas

- Variables are generated one after the other.
- Inner workings:
  1. A variable is generated.
  2. The domain of this variable is generated. (here by an LLM)
  3. Constraints associated with this variable are generated.

# GENERATIVE CONSTRAINT PROGRAMMING

## Core Ideas

- Variables are generated one after the other.
- Inner workings:
  1. A variable is generated.
  2. The domain of this variable is generated. (here by an LLM)
  3. Constraints associated with this variable are generated.
  4. Constraint propagation.

# GENERATIVE CONSTRAINT PROGRAMMING

## Core Ideas

- Variables are generated one after the other.
- Inner workings:
  1. A variable is generated.
  2. The domain of this variable is generated. (here by an LLM)
  3. Constraints associated with this variable are generated.
  4. Constraint propagation.
  5. If any problem occurs: backtrack.

# GENERATIVE CONSTRAINT PROGRAMMING

## Core Ideas

- Variables are generated one after the other.
- Inner workings:
  1. A variable is generated.
  2. The domain of this variable is generated. (here by an LLM)
  3. Constraints associated with this variable are generated.
  4. Constraint propagation.
  5. If any problem occurs: backtrack.
  6. Call the boolean function to check if a solution is found or to cause a backtrack.

# GENERATIVE CONSTRAINT PROGRAMMING

## Core Ideas

- Variables are generated one after the other.
- Inner workings:
  1. A variable is generated.
  2. The domain of this variable is generated. (here by an LLM)
  3. Constraints associated with this variable are generated.
  4. Constraint propagation.
  5. If any problem occurs: backtrack.
  6. Call the boolean function to check if a solution is found or to cause a backtrack.
  7. Go back to 1.

# Text Generation Under Constraints

|       | Domain Size | Structural Constraints | Meaning   | Constraint Satisfaction |
|-------|-------------|------------------------|-----------|-------------------------|
| CP    | Struggles   | Strives                | Struggles | Strives                 |
| GenCP | Strives     | Strives                | Struggles | Strives                 |

# Text Generation Under Constraints

|       | Domain Size | Structural Constraints | Meaning   | Constraint Satisfaction |
|-------|-------------|------------------------|-----------|-------------------------|
| CP    | Struggles   | Strives                | Struggles | Strives                 |
| GenCP | Strives     | Strives                | Struggles | Strives                 |
| ML    |             | Struggles              | Strives   | Struggles               |

# Text Generation Under Constraints

|         | Domain Size | Structural Constraints | Meaning   | Constraint Satisfaction |
|---------|-------------|------------------------|-----------|-------------------------|
| CP      | Struggles   | Strives                | Struggles | Strives                 |
| GenCP   | Strives     | Strives                | Struggles | Strives                 |
| ML      |             | Struggles              | Strives   | Struggles               |
| GenCPML | Strives     | Strives                | Strives   | Strives                 |

# GenCPML

- A variable represents a word.
- Begin with a word/sentence (LLM provides better results).
- An LLM generates the domain of the variables.
  - The domains are of size  $k$ .

# GenCPML

- A variable represents a word.
- Begin with a word/sentence (LLM provides better results).
- An LLM generates the domain of the variables.
  - The domains are of size  $k$ .
- Inner workings:
  1. Give the current sentence formed by the variables to the LLM.

# GenCPML

- A variable represents a word.
- Begin with a word/sentence (LLM provides better results).
- An LLM generates the domain of the variables.
  - The domains are of size  $k$ .
- Inner workings:
  1. Give the current sentence formed by the variables to the LLM.
  2. The LLM answers 1000 tokens.

# GenCPML

- A variable represents a word.
- Begin with a word/sentence (LLM provides better results).
- An LLM generates the domain of the variables.
  - The domains are of size  $k$ .
- Inner workings:
  1. Give the current sentence formed by the variables to the LLM.
  2. The LLM answers 1000 tokens.
  3. Remove invalid tokens (symbols, etc.).

# GenCPML

- A variable represents a word.
- Begin with a word/sentence (LLM provides better results).
- An LLM generates the domain of the variables.
  - The domains are of size  $k$ .
- Inner workings:
  1. Give the current sentence formed by the variables to the LLM.
  2. The LLM answers 1000 tokens.
  3. Remove invalid tokens (symbols, etc.).
  4. Remove the tokens that do not satisfy the constraints.

# GenCPML

- A variable represents a word.
- Begin with a word/sentence (LLM provides better results).
- An LLM generates the domain of the variables.
  - The domains are of size  $k$ .
- Inner workings:
  1. Give the current sentence formed by the variables to the LLM.
  2. The LLM answers 1000 tokens.
  3. Remove invalid tokens (symbols, etc.).
  4. Remove the tokens that do not satisfy the constraints.
  5. Return the  $k$  token with the highest probability.

# GenCPML

assign  
« The » to X1

|     |  |  |  |
|-----|--|--|--|
| X1  |  |  |  |
| The |  |  |  |
|     |  |  |  |
|     |  |  |  |
|     |  |  |  |

# GenCPML

ask the LLM  
1000 tokens

remove invalid  
tokens

keep  $k$   
that satisfy  
the constraints

assign  
« boy » to X2

| X1  | X2     |  |  |
|-----|--------|--|--|
| The | boy    |  |  |
|     | man    |  |  |
|     | fridge |  |  |
|     | ship   |  |  |

# GenCPML

ask the LLM  
1000 tokens

remove invalid  
tokens

keep  $k$   
that satisfy  
the constraints

assign  
« and » to X3

| X1  | X2     | X3    |  |
|-----|--------|-------|--|
| The | boy    | and   |  |
|     | man    | is    |  |
|     | fridge | wants |  |
|     | ship   | play  |  |

# GenCPML

ask the LLM  
1000 tokens

remove invalid  
tokens

keep  $k$   
that satisfy  
the constraints

assign  
« the » to X4

| X1  | X2     | X3    | X4   |
|-----|--------|-------|------|
| The | boy    | and   | the  |
|     | man    | is    | his  |
|     | fridge | wants | that |
|     | ship   | play  | this |

# GenCPML

send LLM ->  
« The boy and  
the »  
does not answer a  
dot (« . ») in 1000  
tokens.

Not a solution

| X1  | X2     | X3    | X4   |
|-----|--------|-------|------|
| The | boy    | and   | the  |
|     | man    | is    | his  |
|     | fridge | wants | that |
|     | ship   | play  | this |

# GenCPML

The LLM never answers a dot (« . ») in 1000 tokens with any of the values of x4.

| X1  | X2     | X3    | X4   |
|-----|--------|-------|------|
| The | boy    | and   | the  |
|     | man    | is    | his  |
|     | fridge | wants | that |
|     | ship   | play  | this |

# GenCPML

backtrack  
assign  
« is » to X3

| X1  | X2     | X3    |  |
|-----|--------|-------|--|
| The | boy    | and   |  |
|     | man    | is    |  |
|     | fridge | wants |  |
|     | ship   | play  |  |

# GenCPML

ask the LLM  
1000 tokens

remove invalid  
tokens

keep  $k$   
that satisfy  
the constraints

assign  
« happy » to X4

| X1  | X2     | X3    | X4     |
|-----|--------|-------|--------|
| The | boy    | and   | happy  |
|     | man    | is    | grumpy |
|     | fridge | wants | going  |
|     | ship   | play  | eating |

# GenCPML

send LLM ->  
« The boy is  
happy »  
answer a dot (« . »)  
in 1000 tokens.

A solution is found.

| X1  | X2     | X3    | X4     |
|-----|--------|-------|--------|
| The | boy    | and   | happy  |
|     | man    | is    | grumpy |
|     | fridge | wants | going  |
|     | ship   | play  | eating |

# Experimental Results

## Technical Remarks

### Comparison between Beam Search and GenCPML:

- Beam Search (Post & Villar, 2018)
  - Well-known NLP technique, a greedy algorithm.
- Compared on Text generation under constraints benchmarks.

# Experimental Results

## Technical Remarks

### Comparison between Beam Search and GenCPML:

- Beam Search (Post & Villar, 2018)
  - Well-known NLP technique, a greedy algorithm.
- Compared on Text generation under constraints benchmarks.
- Comparison 1: time to find the same number of solutions.

# Experimental Results

## Technical Remarks

### Comparison between Beam Search and GenCPML:

- Beam Search (Post & Villar, 2018)
  - Well-known NLP technique, a greedy algorithm.
- Compared on Text generation under constraints benchmarks.
- Comparison 1: time to find the same number of solutions.
- Comparison 2: number of solutions found given the same time.

# Experimental Results

## Results

### GenCPML VS Beam Search:

- GenCPML is faster than Beam Search while having 100% satisfaction.
- GenCPML finds around 10 times more solutions than Beam Search.

# Experimental Results

## Results

### GenCPML VS Beam Search:

- GenCPML is faster than Beam Search while having 100% satisfaction.
- GenCPML finds around 10 times more solutions than Beam Search.
- Sometimes, Beam Search answers “wrong output”, outputs that do not satisfy all the constraints.

# Experimental Results

## Results

### GenCPML VS Beam Search:

- GenCPML is faster than Beam Search while having 100% satisfaction.
- GenCPML finds around 10 times more solutions than Beam Search.
- Sometimes, Beam Search answers “wrong output”, outputs that do not satisfy all the constraints.
- On average, Beam Search has 50% constraint satisfaction. Depending on the query Beam Search has 0% satisfaction or 100% satisfaction.

# Experimental Results

## Chosen Queries

Text generation under constraints, not free text generation.

COLLIE: Systematic Construction of Constrained Text Generation Tasks,

Yao et al, ICLR 2024

| Query number | words count | character count           | lexical constraints                                           |
|--------------|-------------|---------------------------|---------------------------------------------------------------|
| 1            |             | = 82                      |                                                               |
| 2            | = 10        |                           | $X_3 = \text{soft}, X_7 = \text{beach}, X_{10} = \text{math}$ |
| 3            | $\geq 20$   | $\forall i,  X_i  \leq 6$ |                                                               |
| 4            |             |                           | soft, beach, math                                             |

# Experimental Results

## LLM Constraint Satisfaction

LLM results. Asked the queries 10 times.

#s/#f is the number of generated sentences that satisfy/do not satisfy the query.

%sat is the percentage of sentences that satisfy the queries.

| Query number | GPT-4 |    |      | GPT-4.0 |    |      | Mistral Next |    |      | Claude3.5 Sonnet |    |      | Gemini |    |      |
|--------------|-------|----|------|---------|----|------|--------------|----|------|------------------|----|------|--------|----|------|
|              | #s    | #f | %sat | #s      | #f | %sat | #s           | #f | %sat | #s               | #f | %sat | #s     | #f | %sat |
| 1            | 1     | 9  | 10%  | 0       | 10 | 0%   | 0            | 10 | 0%   | 1                | 9  | 10%  | 0      | 10 | 0%   |
| 2            | 0     | 10 | 0%   | 0       | 10 | 0%   | 0            | 10 | 0%   | 0                | 10 | 0%   | 0      | 10 | 0%   |
| 3            | 1     | 9  | 10%  | 5       | 5  | 50%  | 0            | 10 | 0%   | 9                | 1  | 90%  | 1      | 9  | 10%  |
| 4            | 9     | 1  | 90%  | 9       | 1  | 90%  | 10           | 0  | 100% | 10               | 0  | 100% | 1      | 9  | 10%  |

# Other Results

## 10 words between 72 and 74 characters

- The following article was originally published in Business Review North. (10 words, 72 characters).
- The following are resources available for students seeking financial aid. (10 words, 73 characters).

## 10-15 words, does not contain the characters “e”, “a” and “t”

- I know you will find your own words of wisdom. (10 words).
- I work in my own shop in my own building. (10 words).

## 20 words, a word contains maximum 5 characters

- The first time you see a photo of your baby in the arms of his or her birth mom is an image that will stay with you (27 words).

# Other Results: Metavar

## Some solutions of Query 1 (82 characters):

- The following is a list of all the posts in this blog that have to do with eBooks.
- {The; following; is; a; list; of; all; the; posts; in; this; blog; that; have; to; do; with; e;Book;s;.}
- The following is a list of all the posts in this blog that have to do with eating.
- {The; following; is; a; list; of; all; the; posts; in; this; blog; that; have; to; do; with; e;ating;.}
- The following is a list of all the posts in this blog that have to do with my dad.
- {The; following; is; a; list; of; all; the; posts; in; this; blog; that; have; to; do; with; my; d;ad;.}

# Other Results: Restart

## Some solutions of Query 1 (82 characters) with restart at the second word:

- The following is a list of all the posts in this blog that have to do with eBooks.
- {The; following; is; a; list; of; all; the; posts; in; this; blog; that; have; to; do; with; e;Book;s;.}
- The newest addition to the family of the world's most popular and versatile darts.
- {The; new;est; addition; to; the; family; of; the; world;'s; most; popular; and; vers;atile; d;arts;.}
- The first time you see a picture of yourself, it can be a shocking moment to take.
- {The; first; time; you; see; a; picture; of; yourself;;; it; can; be; a; shock;ing; moment; to; take;.}

# Other Results: Metavar & Restart

| Query | k  | restart( $n$ ) | #sols | callLLM |
|-------|----|----------------|-------|---------|
| 1     | 50 | 0              | 28    | 247     |
|       |    | 2              | 5     | 306     |

Old: 2 solutions in 20 minutes.

Now: 28 solutions in 10 minutes without restart and 5 in 10 minutes with restart(2).

# Other Results: Paragraph

3 sentences, at least 15 words each (a period counts as a word):

- The first time you see a picture of yourself, it can be a shock. You may not like what you see and feel that your appearance is unattractive. This feeling can lead to low self esteem and even depression in some cases.
- {The; first; time; you; see; a; picture; of; yourself;; it; can; be; a; shock;. ; You; may; not; like; what; you; see; and; feel; that; your; appearance; is; un;att;ract;ive;. ; This; feeling; can; lead; to; low; self; este;em; and; even; de;pression; in; some; cases; .}

# Other Results: Better meaning

Using the LLM OpenAI babbage-002 instead of LLaMa Q4 7B.

## Pre-prompt:

- “Amidst the crimson glow of a setting sun, a lone warrior, clad in battle-worn silver, stood atop the ancient ruins, his blade gleaming with the promise of legend.”

## 82 characters:

- The warrior stood silent and immovable for a long while, his body rigid and gaunt.
- {; The; warrior; stood; silent; and; imm;ovable; for; a; long; while;; his; body; rigid; and; ga;unt;}.

# Other Results: Better meaning

Using the LLM OpenAI babbage-002 instead of LLaMa Q4 7B.

## Pre-prompt:

- “Amidst the crimson glow of a setting sun, a lone warrior, clad in battle-worn silver, stood atop the ancient ruins, his blade gleaming with the promise of legend.”

## 3 sentences, sentence 1 begin by “Dragons”, 2 by “Kingdoms” and 3 by “Barbarians”:

- Dragons, giants, elves and otherworldly foes lay scattered around the hilltop. Kingdoms were crumbling at his feet. Barbarians roamed the land.
- {; Dragons;; giants;; elves; and; other;world;ly; foes; lay; scattered; around; the; hill;top;. Kingdoms; were; crumbling; at; his; feet;. Barbarians; ro;amed; the; land;.

# CONCLUSION

- **Generative Constraint Programming with Machine Learning:**
  - Use an LLM to generate the domains of a CSP.
  - Can manage Structural AND Semantic constraints.
    - Can solve problems that ML alone and CP alone struggle with.
  - Avoid CP combinatorial explosion.
  - Fix ML constraint satisfaction (does not answer “false” solutions).
  - Has better results than a well-known Natural Language Processing (NLP) technique: Beam Search.

# Personnal

- I'm looking for a post-doctorate position.
- My email: [florian.regin06@gmail.com](mailto:florian.regin06@gmail.com)
- Linkedin link: [www.linkedin.com/in/florian-regin-b15073352](https://www.linkedin.com/in/florian-regin-b15073352)



