GenAI Unpacked: BEYOND BASICS

Developer World

Good Vibes. Good People. Good Skills.

Zur DWX

Dr. Damir Dobric

Lead Software Architect daenet GmbH / ACP Digital

Microsoft Regional Director,

Most Valuable Professional: AI

ML vs. GenAI

What is different and why it is important?

- Traditional ML
  - Lot of Data
  - Long Training
- GenAI
  - Pretrained
  - GPT=General Pretrained Transformer

AI Services

daenet - an ACP Digital company

Open AI

https://platform.openai.com/docs/models

Azure AI Foundry

https://oai.azure.com/portal/****/models
https://ai.azure.com/

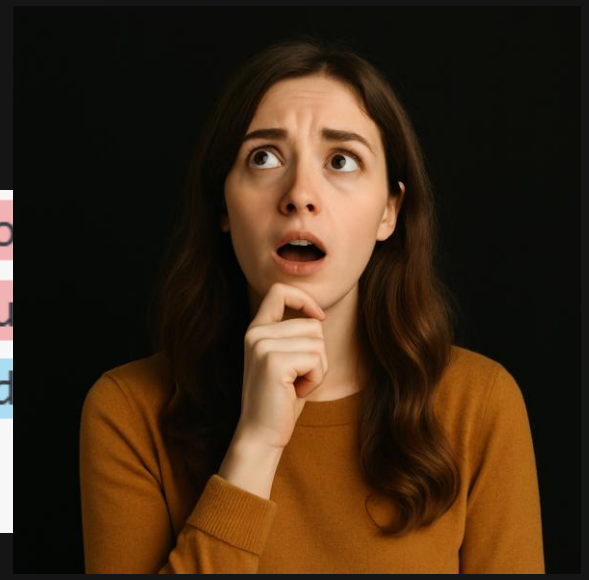
Working with TOKENS

daenet - an ACP Digital company

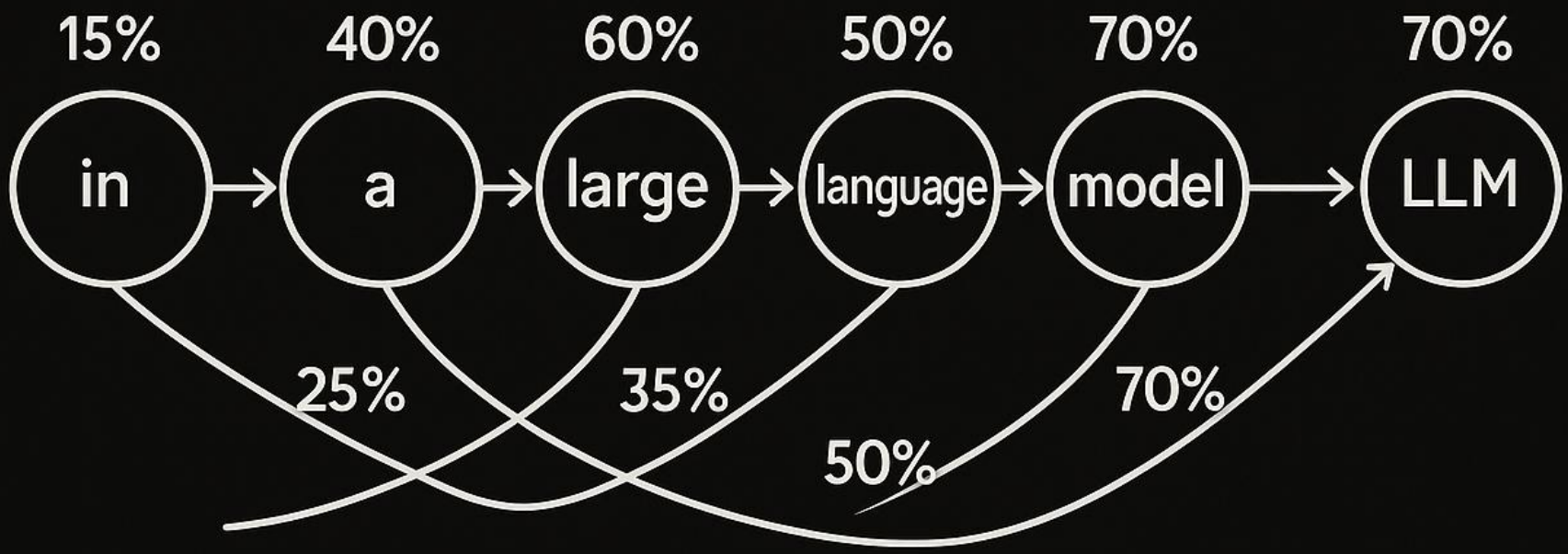
What are Tokens?

- 1 token ~= 4 chars in English
- 1 token ~= ¾ words
- 100 tokens ~= 75 words
- Byte Pare Encoding (Gage,1994): Wikipedia
- What are tokens and how to count them?
- Token Pricing: Pricing - OpenAI API

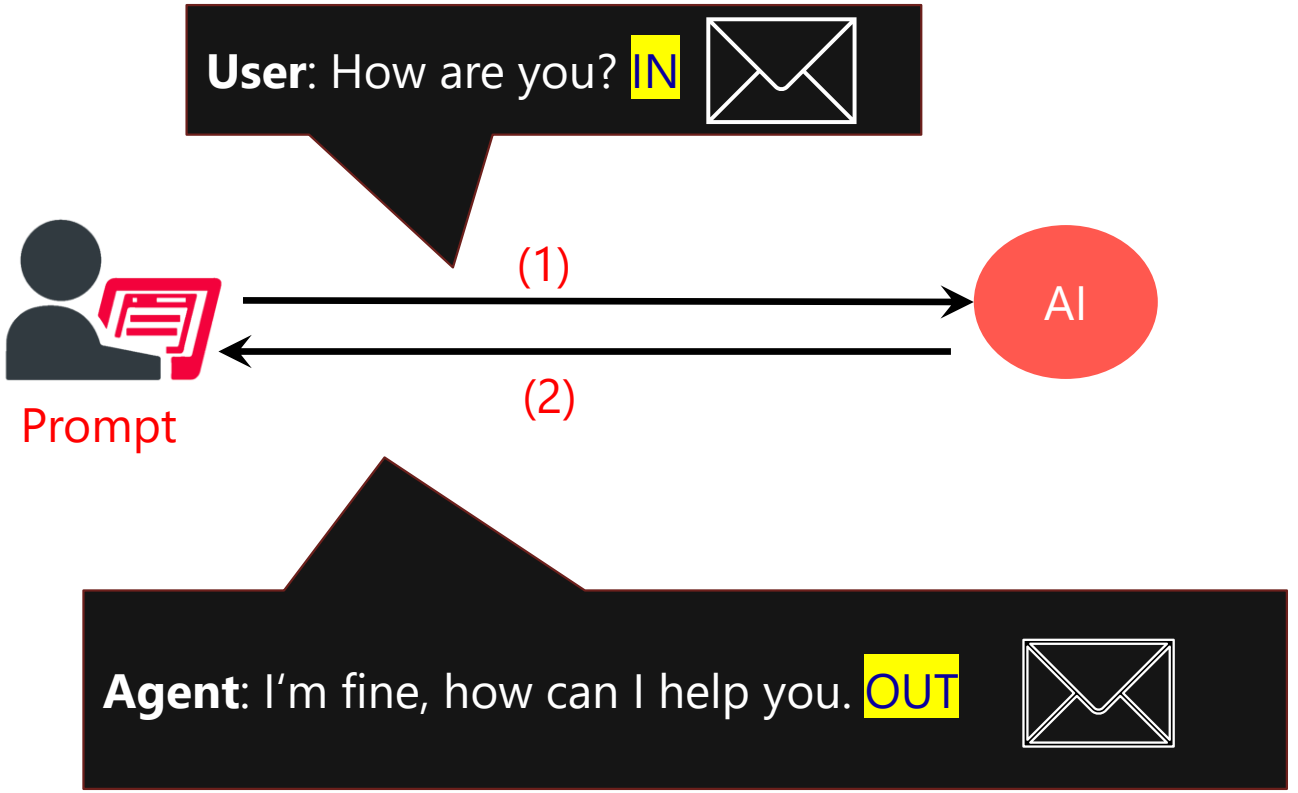
```
OpenAI's large language models process text using tokens, which are co
 sequences of characters found in a set of text. The models learn to u
 the statistical relationships between these tokens, and excel at prod
 next token in a sequence of tokens.
```



Transformer follows token probabilities



Token Costs


User: How are you? IN
(1)
AI
(2)
Prompt
Agent: I'm fine, how can I help you. OUT

DEMO

- Tokenizer
- Creating tokens in C#

daenet

daenet - an ACP Digital company

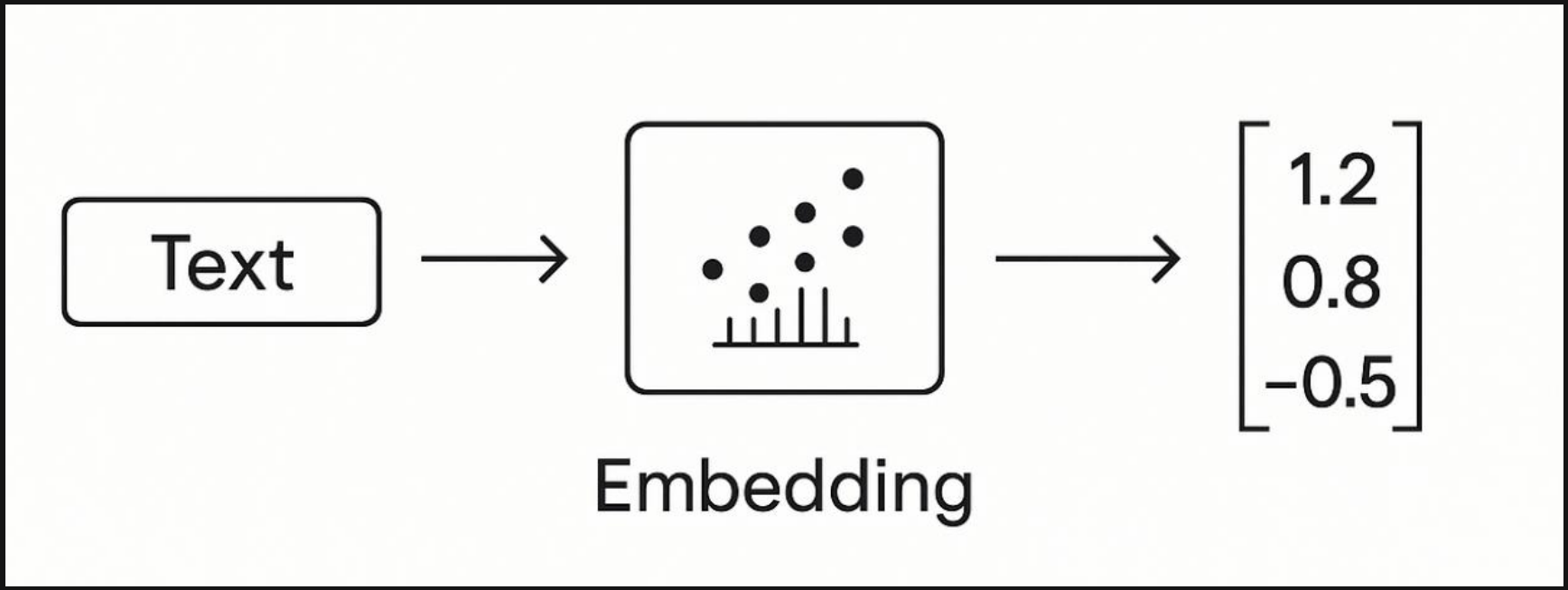
Working with EMBEDDINGS

daenet

daenet - an ACP Digital company

Embeddings

- Making a vector from text



Similarity Between Multidimensional Vectors

- Dot Product

$$A \cdot B = a_1 \cdot b_1 + a_2 \cdot b_2 + \ldots + a_n \cdot b_n$$

- The Norm

$$\|\mathbf{A}\| = \sqrt{a_1^2 + a_2^2 + \ldots + a_n^2}$$

- Cosine Similarity

$$\mathbf{A} \cdot \mathbf{B} = \|\mathbf{A}\| \|\mathbf{B}\| \cos(\theta)$$

Embeddings Use-Cases?

- Semantic Search
- Classification
- Clustering
- Recommendations
- . . .

DEMO

- Consuming Embedding Model with HTTP/Post
- Creating Embeddings in C#
- Classification Demo

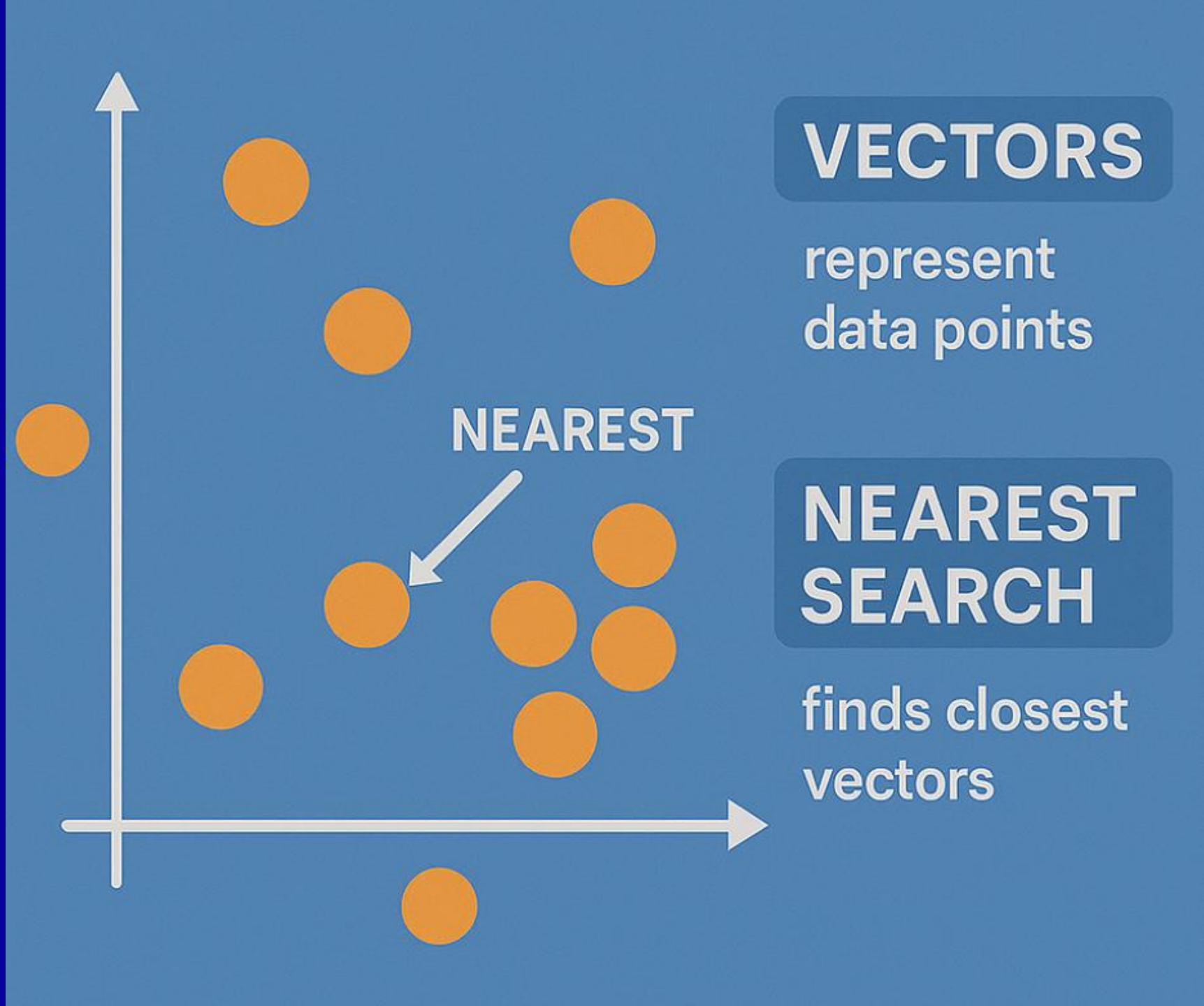
daenet

daenet - an ACP Digital company

VECTOR DATABASES

daenet - an ACP Digital company

Embeddings in GraphDB



Vector DBs

- SqlServer 2025 Native Vector Search
- Quadrant
- CosmosDB
- …

SqlServer 2025 Native Vector Search

- SqlServer 2025 Native Vector Search
  - In Azure
  - On-Prem
- SQL Server Native Vector Search for .NET Developers

```
CREATE TABLE test.Vectors
(
    [Id] INT IDENTITY(1,1) NOT NULL,
    [Text] NVARCHAR(MAX) NULL,
    [VectorShort] VECTOR(3)  NULL,
    [Vector] VECTOR(1536)  NULL
);
```

DEMO

- SQL Server Native Vector Search

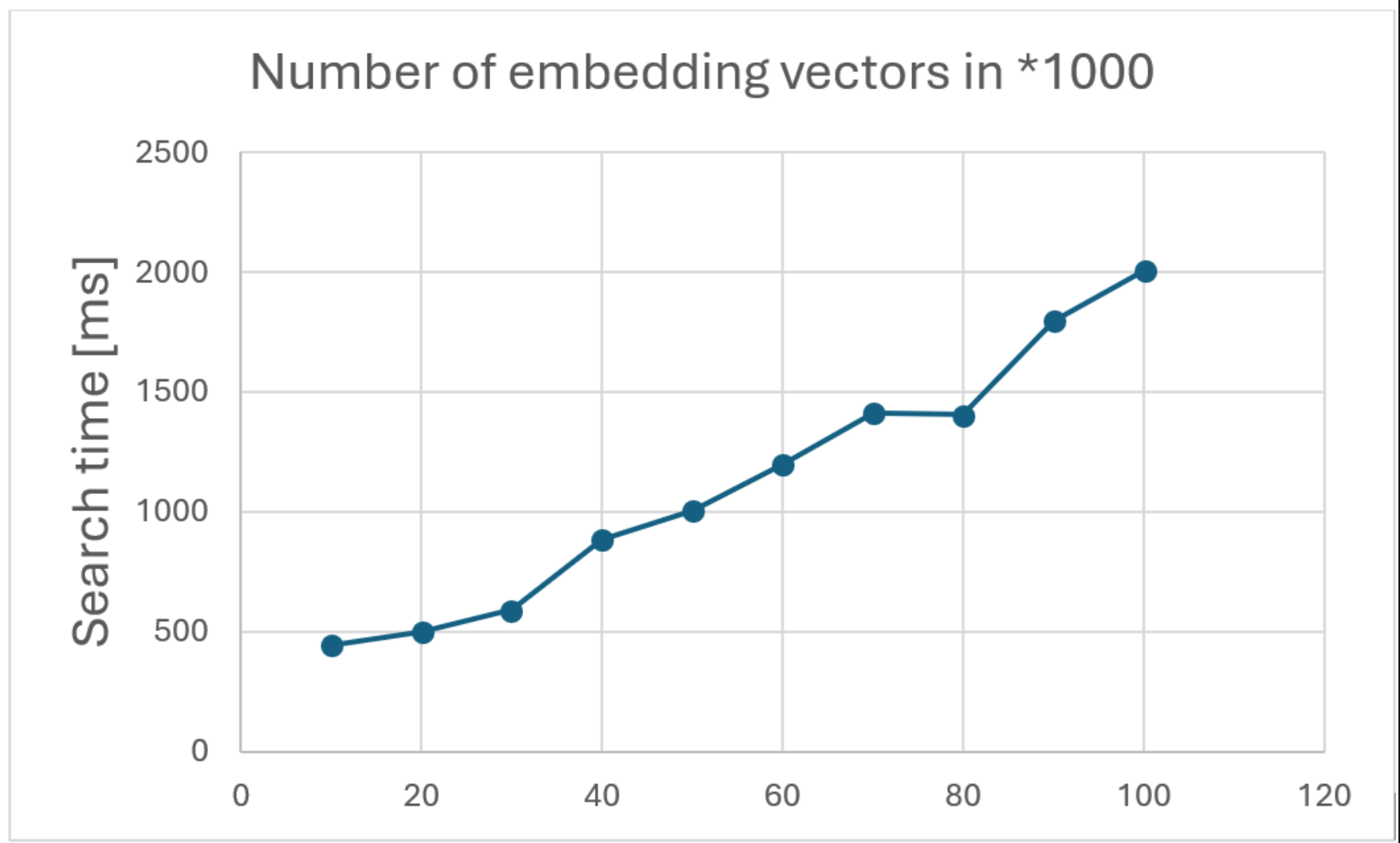
  SQL Server Native Vector

```
DECLARE @v1 VECTOR(2) = '[1,1]';
DECLARE @v2 VECTOR(2) = '[-1,-1]';

SELECT
    VECTOR_DISTANCE('euclidean', @v1, @v2) AS euclidean,
    VECTOR_DISTANCE('cosine', @v1, @v2) AS cosine,
    VECTOR_DISTANCE('dot', @v1, @v2) AS negative_dot_product;
```

daenet - an ACP Digital company

Keep in Mind!!


Number of embedding vectors in *1000
Search time [ms]
2500
2000
1500
1000
500
0
0
20
40
60
80
100
120

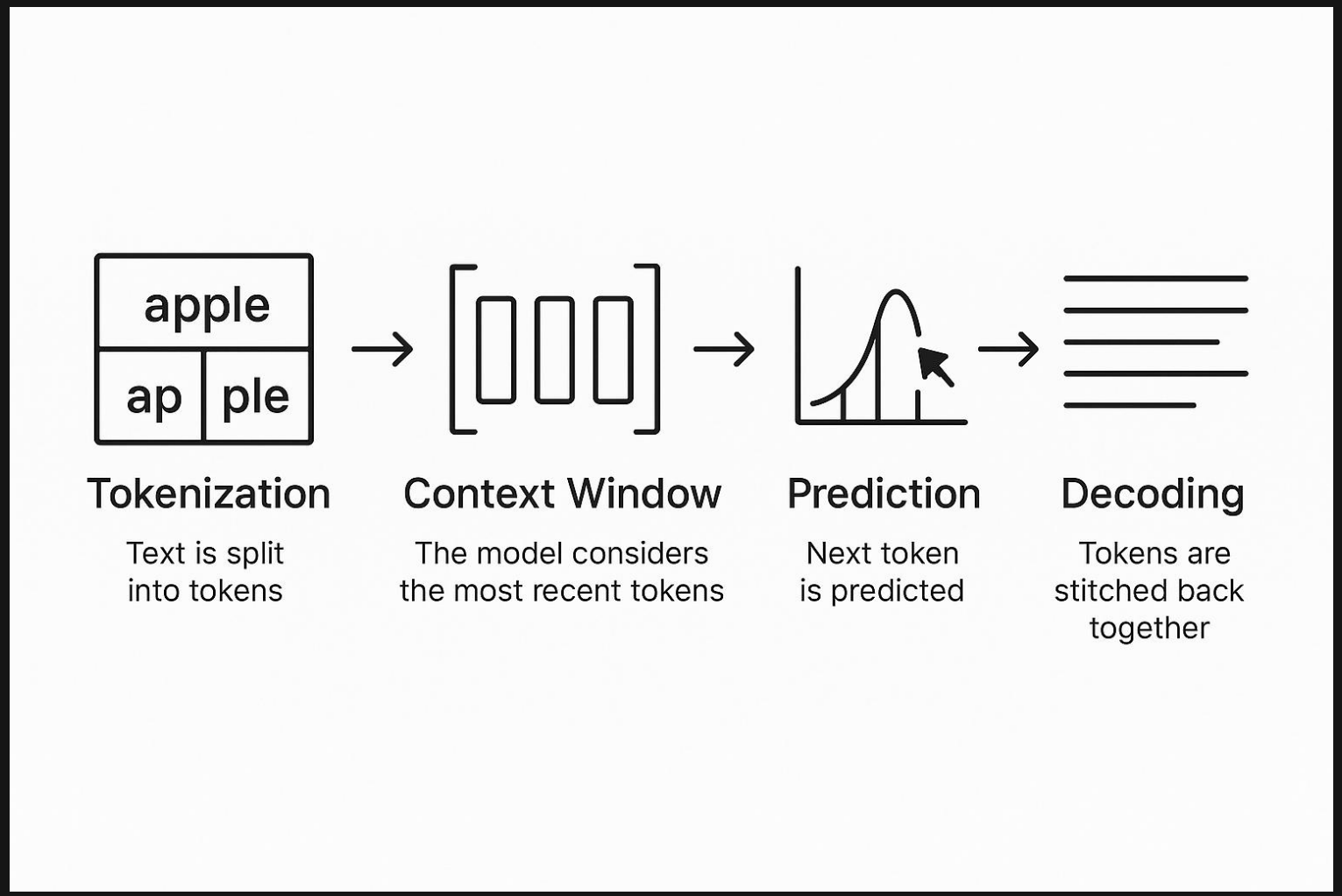
COMPLETION MODELS

How does all this work?

daenet - an ACP Digital company

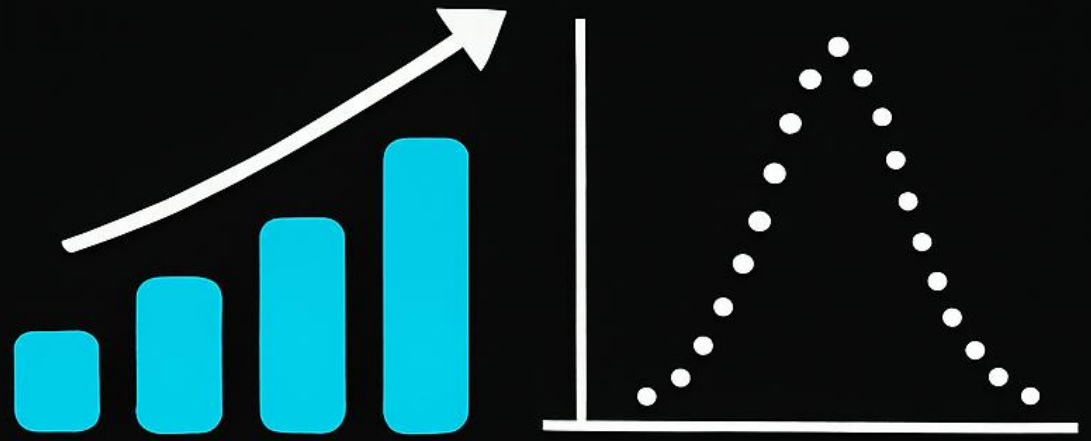
Completion Models

Text Completion Flow

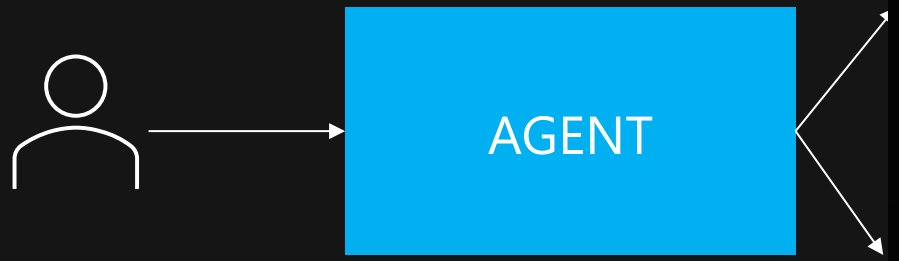


DEMO

- C# Application
- Using OpenAI nuget package
- Executing ChatCompletions
- Presenting Probabilities



Extending Models



Knowledge Tools => RAG

Bing Search

File Search

Azure AI Search

MS Fabric

Action Tools => Function Calling

Function Calling

Code Interpreter

OpenAPI Defined Tools

Azure Functions

RAG

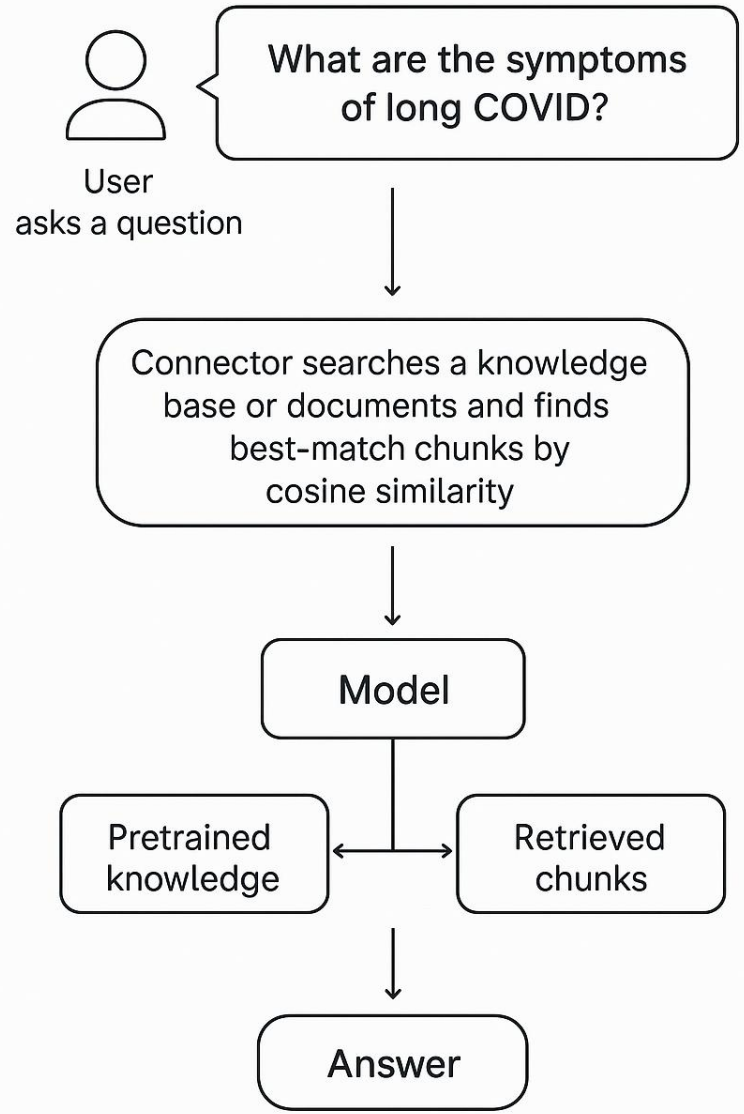
Retrieval Augmented Generation

daenet - an ACP Digital company

RAG
Retrieval-Augmented Generation

- Extending the model knowledge
- Combines information retrieval with text generation

https://arxiv.org/abs/2005.11401



DEMO (c#)

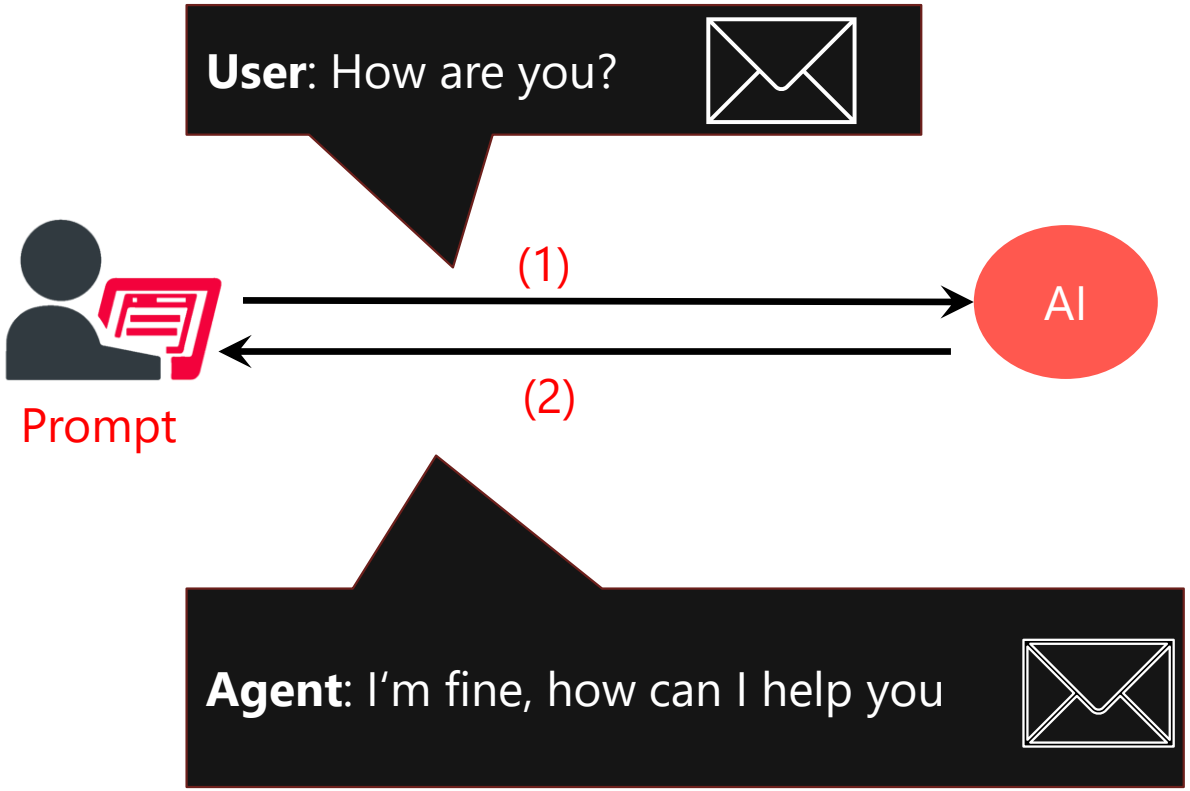
- Creating chunks
- Creating embeddings
- InMemory Vector DB
- Calculating Similarity

daenet - an ACP Digital company

Function Calling

daenet - an ACP Digital company

Chat Model Interaction



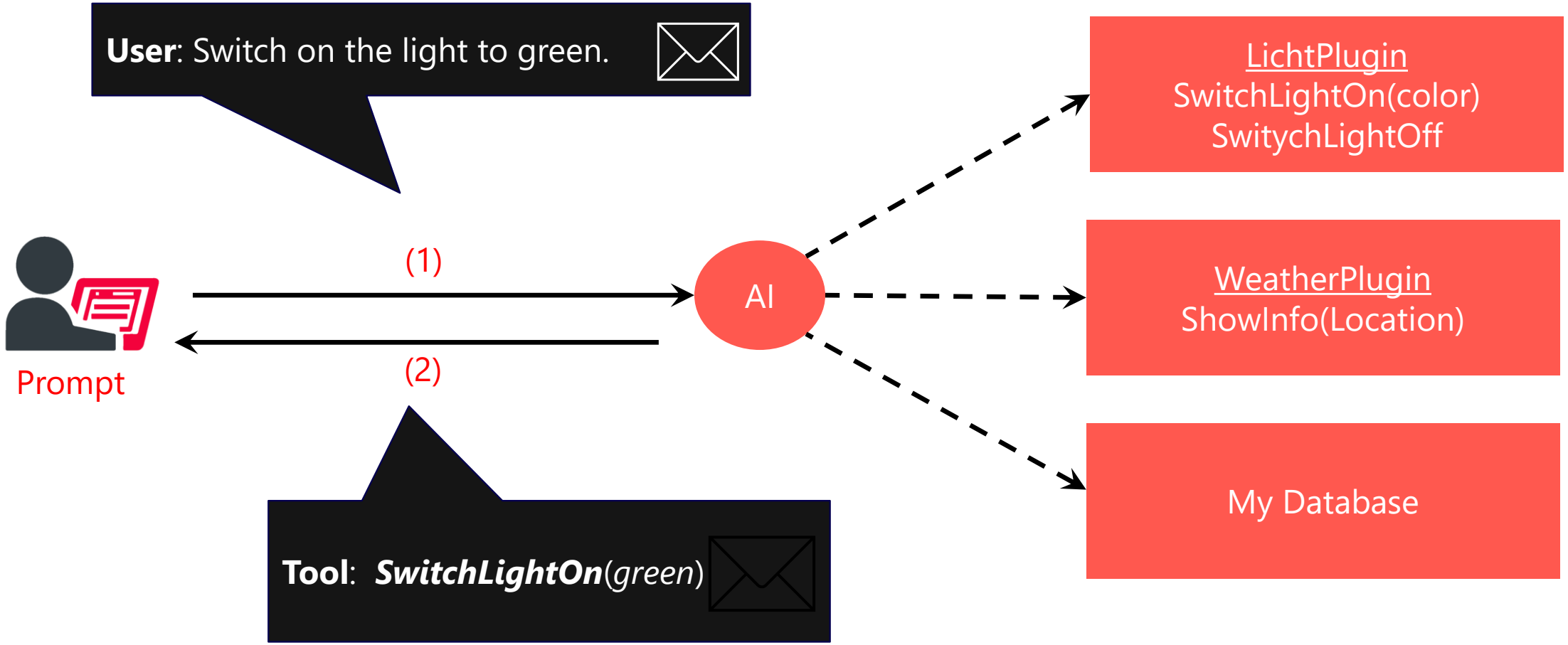
Function Calling

daenet

daenet - an ACP Digital company

```
{
    "type": "function",
    "function": {
        "name": "get_weather",
        "description": "Retrieves current weather for the given location.",
        "strict": true,
        "parameters": {
            "type": "object",
            "properties": {
                "location": {
                    "type": "string",
                    "description": "City and country e.g. Bogotá, Colombia"
                },
                "units": {
                    "type": ["string", "null"],
                    "enum": ["celsius", "fahrenheit"],
                    "description": "Units the temperature will be returned in."
                }
            },
            "required": ["location", "units"],
            "additionalProperties": false
        }
    }
}
```

Function Calling

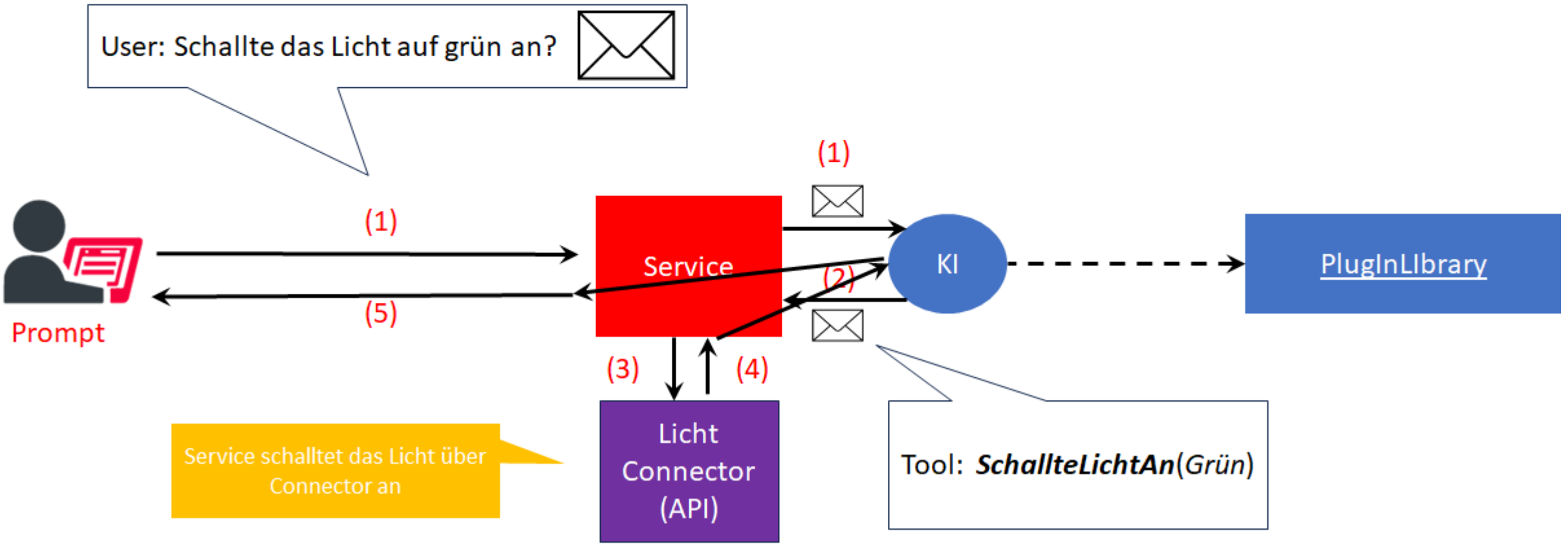


Function Calling

daenet

Model wird zwei Mal angewendet:

1. Um Prompt auf Funktion zu mappen. Keine Daten an Model – (1)
2. Um die User Message zu generieren (Die Daten, die von Connector zur Verfügung gestellt werden(4). Z.B. Licht ist an.



DEMO (c#)

- Function Calling with
- SemanticKernel Plugin

daenet

daenet - an ACP Digital company

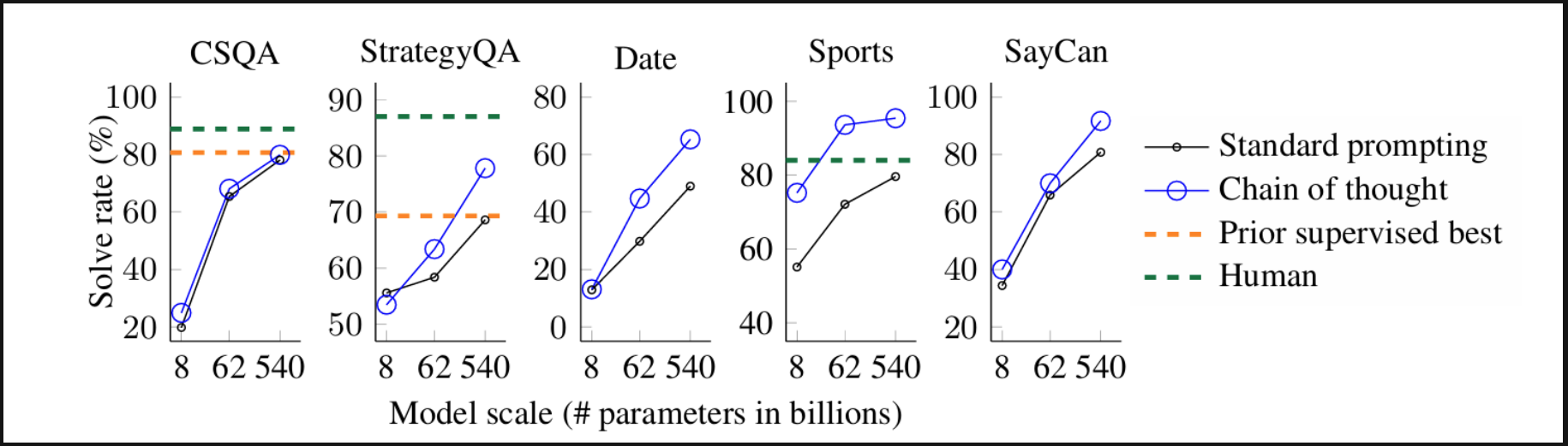
A Scientific look under the hub

daenet

daenet - an ACP Digital company

Chain of Tought

- A prompting technique used to improve reasoning
- Models think step by step before giving an answer.

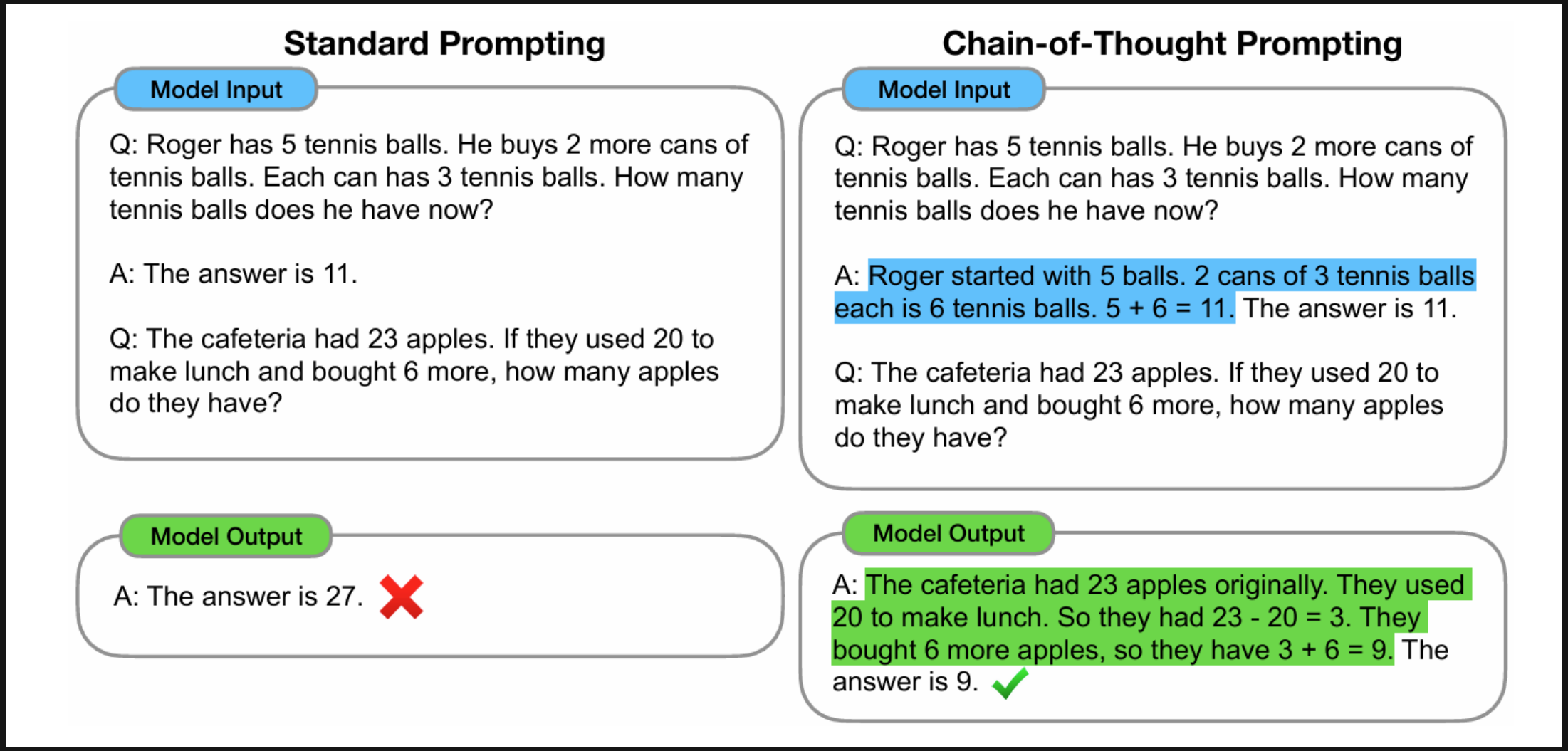


Google Research, Brain Team
https://arxiv.org/abs/2201.11903

The Chain of Tought (CoT)

daenet

daenet - an ACP Digital company



Reasoning

- Common Models are
  - Outcome-Supervised Reward Models (ORM).
  - They use final answers
- Reasoning Models
  - Receive Feedback on Each Step in Chain-of-Tought process
  - Process Reasoning Models (PRM)

https://cdn.openai.com/improving-mathematical-reasoning-with-process-supervision/Lets_Verify_Step_by_Step.pdf

Human Supervised Learning

The denominator of a fraction is 7 less than 3 times the numerator. If the fraction is equivalent to $2/5$, what is the numerator of the fraction? (Answer: $\boxed{14}$)

Let's call the numerator x.

So the denominator is 3x-7.

We know that x/(3x-7) = 2/5.

So 5x = 2(3x-7).

5x = 6x - 14.

So x = 7.

Figure 1: A screenshot of the interface used to collect feedback for each step in a solution.

Let

$$x^8 + 3x^4 - 4 = p_1(x)p_2(x)\cdots p_k(x),$$

where each non-constant polynomial $p_i(x)$ is monic with integer coefficients, and cannot be factored further over the integers. Compute $p_1(1) + p_2(1) + \cdots + p_k(1)$.

I notice that the given polynomial has even degree and only even powers of x, so I can try to make a substitution to simplify it.

Let $y = x^4$, then the polynomial becomes $y^2 + 3y - 4$, which is a quadratic equation.

I can factor this quadratic equation as $(y + 4)(y - 1)$, so the original polynomial is $(x^4 + 4)(x^4 - 1)$.

Now I need to factor each of these factors further, if possible, over the integers.

For the first factor, $x^4 + 4$, I recall the Sophie Germain identity, which says that $a^4 + 4b^4 = (a^2 + 2b^2 + 2ab)(a^2 + 2b^2 - 2ab)$.

Applying this identity with $a = x$ and $b = 1$, I get $x^4 + 4 = (x^2 + 2x + 2)(x^2 - 2x + 2)$.

For the second factor, $x^4 - 1$, I can use the difference of squares formula, which says that $a^2 - b^2 = (a + b)(a - b)$.

Applying this formula with $a = x^2$ and $b = 1$, I get $x^4 - 1 = (x^2 + 1)(x^2 - 1)$.

Then I can apply the difference of squares formula again with $a = x$ and $b = 1$, to get $x^2 - 1 = (x + 1)(x - 1)$.

So the original polynomial is $(x^4 + 4)(x^4 - 1) = (x^2 + 2x + 2)(x^2 - 2x + 2)(x^2 + 1)(x + 1)(x - 1)$.

These are all the irreducible factors over the integers, so $p_1(x) = x^2 + 2x + 2$, $p_2(x) = x^2 - 2x + 2$, $p_3(x) = x^2 + 1$, $p_4(x) = x + 1$, and $p_5(x) = x - 1$.

To find the sum of their values at $x = 1$, I just need to plug in $x = 1$ into each factor and add them up.

I get $p_1(1) = 1^2 + 2 \cdot 1 + 2 = 5$, $p_2(1) = 1^2 - 2 \cdot 1 + 2 = 1$, $p_3(1) = 1^2 + 1 = 2$, $p_4(1) = 1 + 1 = 2$, and $p_5(1) = 1 - 1 = 0$.

So the final answer is $5 + 1 + 2 + 2 + 0 = 10$.

Answer: 10

Let

$$x^8 + 3x^4 - 4 = p_1(x)p_2(x)\cdots p_k(x),$$

where each non-constant polynomial $p_i(x)$ is monic with integer coefficients, and cannot be factored further over the integers. Compute $p_1(1) + p_2(1) + \cdots + p_k(1)$.

This problem is about factoring a polynomial over the integers, which can be a tricky task.

One possible strategy is to look for rational roots of the polynomial, using the rational root theorem.

However, that would involve checking a lot of possible factors of -4, and there is no guarantee that the polynomial has any rational roots at all.

Another possible strategy is to use some algebraic tricks to simplify the polynomial or reduce its degree.

I notice that the polynomial has only even powers of x, so I can try to substitute $y = x^2$ and see what happens.

Then the polynomial becomes $y^4 + 3y^2 - 4$, which is a quadratic equation in y^2.

I can try to factor this quadratic equation by looking for two numbers that multiply to -4 and add to 3.

Those numbers are 4 and -1, so I can write $y^4 + 3y^2 - 4 = (y^2 + 4)(y^2 - 1)$.

Now I can substitute back $x^2 = y$ and get $x^8 + 3x^4 - 4 = (x^4 + 4)(x^4 - 1)$.

This looks promising, because I have reduced the degree of the factors from 8 to 4.

I can try to factor each of these factors further, using the difference of squares formula.

I get $x^8 + 3x^4 - 4 = (x^2 + 2)(x^2 - 2)(x^2 + 1)(x^2 - 1)$.

I can apply the difference of squares formula again to the last factor and get $x^8 + 3x^4 - 4 = (x^2 + 2)(x^2 - 2)(x^2 + 1)(x + 1)(x - 1)$.

Now I have factored the polynomial completely into monic linear and quadratic factors with integer coefficients.

These are the $p_i(x)$'s that the problem is asking for.

To find the sum of their values at $x = 1$, I just need to plug in $x = 1$ into each factor and add them up.

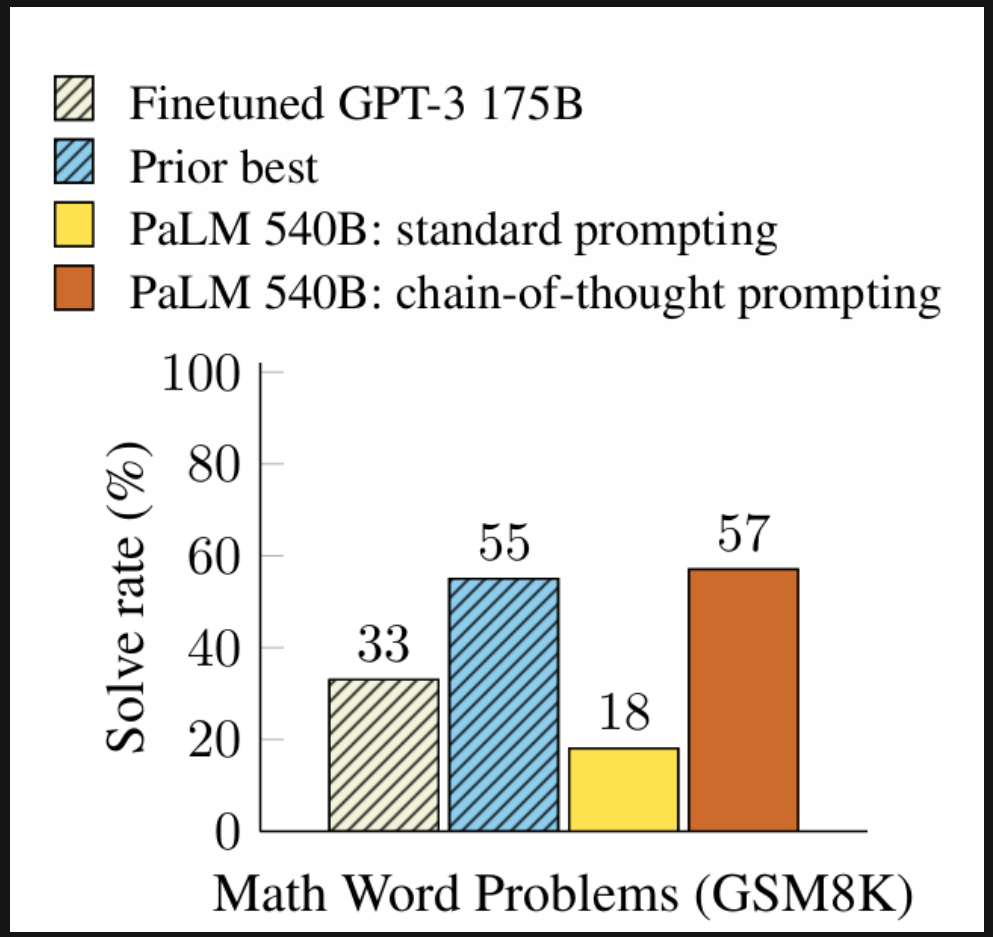
I get $p_1(1) + p_2(1) + \cdots + p_k(1) = (1^2 + 2)(1^2 - 2)(1^2 + 1)(1 + 1)(1 - 1)$.

Simplifying, I get $p_1(1) + p_2(1) + \cdots + p_k(1) = (3)(-1)(2)(2)(0)$.

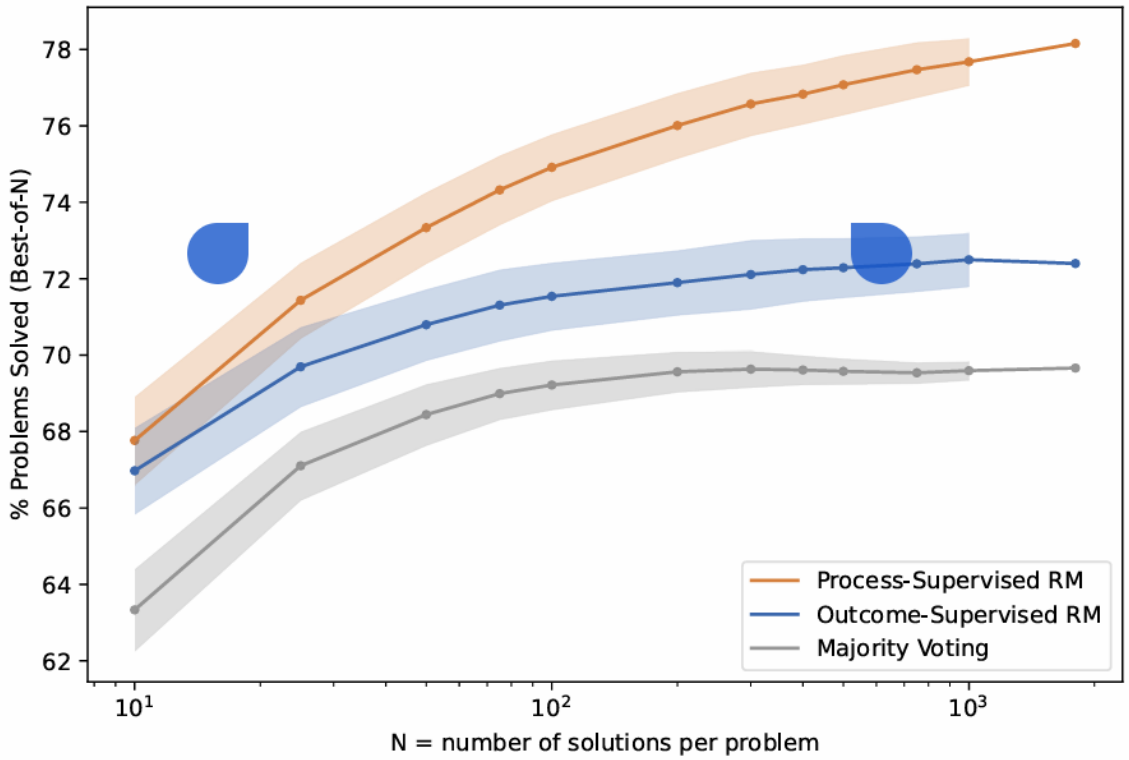
Multiplying, I get $p_1(1) + p_2(1) + \cdots + p_k(1) = 0$.

Answer: 0

Comparison: CoT vs. PRM (Process Reasoning Models)



	ORM	PRM	Majority Voting
% Solved (Best-of-1860)	72.4	**78.2**	69.6



Hendrycks, C. Burns, S. Kadavath, A. Arora, S. Basart, E. Tang, D. Song, and J. Steinhardt. Measuring mathematical problem solving with the math dataset. arXiv preprint arXiv:2103.03874, 2021

Model Performance

LiveBench