Generative Artificial Intelligence (GenAI)

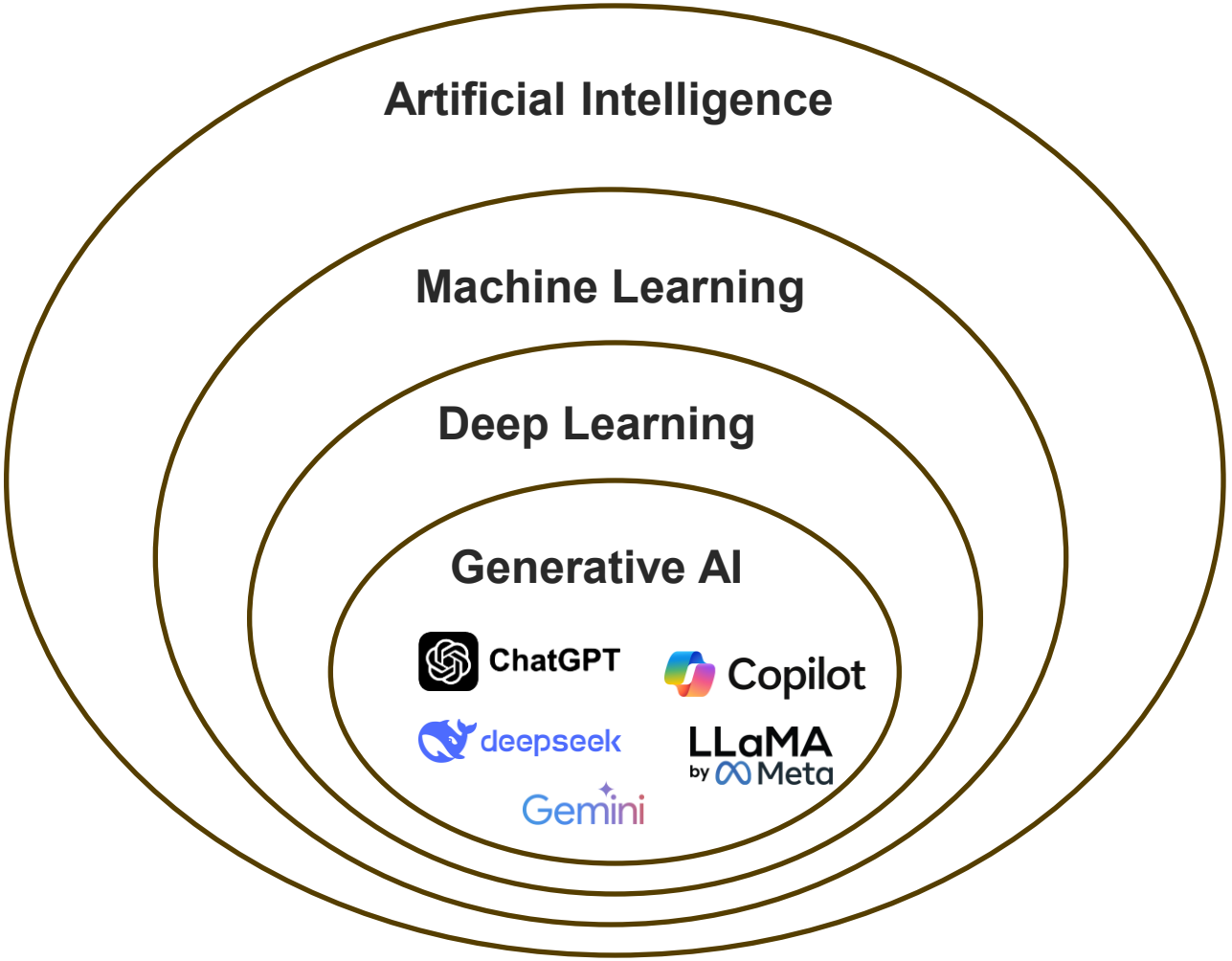
Eduardo Figueiredo

http://www.dcc.ufmg.br/~figueiredo

Generative AI

- GenAI uses models to generate text, images, audio, videos, software code or other forms of data
  - These models learn the underlying patterns and structures of their training data
- It is powered by foundation models
  - Including Large Language Models (LLMs), such as GPT

AI vs. ML vs. DL vs. GenAI



Transformer and Attention

- Transformer is a neural network to process sequences (like text)
  - It understands context and relationships between words
  - Instead of reading word-by-word, it looks at all words and decides which ones matter most
- Attention focuses on relevant words when predicting the next token

Predicting the Next Token

- Given a sequence of tokens, the model predicts the most likely next token

I love visiting _____

0.40: places

0.15: new

0.10: beautiful

Models do not always pick the top token.

I love visiting places _____

0.20: where

0.17: that

0.12: and

Models do not always pick the top token.

Predicting the Next Tokens

I love visiting _____

new places, exploring different cultures.

places that make me feel inspired and alive.

beautiful places and creating unforgettable memories.

places where I can relax, explore, and learn something new.

Prompt and Outputs

I love visiting

new places, exploring different cultures.

places that make me feel inspired and alive.

beautiful places and creating unforgettable memories.

places where I can relax, explore, and learn something new.

Prompt

GenAI Outputs

Beyond Text

- Since models work with numerical representations (vectors), the same idea is applicable to other inputs and outputs
- Examples of GenAI models
  - Text: ChatGPT, Claude
  - Image: DALL-E, Midjourney
  - Music: Suno AI, MusicLM
  - Video: OpenAI Sora, Runway

A Java Example

Prompt: Java implementation of factorial

Output:

```
public class FactorialExample {

  public static int factorial(int n) {
    if (n == 0) return 1; // base case
    return n * factorial(n - 1); // recursive call
  }
}
```

- Note that
  - The model does not “understand recursion”
  - It predicts patterns seen in training data

GPT and Attention

- GPT: Generative Pre-trained Transformer
  - Generative AI
  - Trained on a massive dataset (entire web)
  - Understands context and relationships between words
- Transformer relies on attention to focus on relevant words

An Example of Attention

- Process the entire input as a whole
- Focus on relevant words

Input: *“Jane, who lives in London and works as a software engineer, met her best friend Sarah for dinner”.*

Prompt: Where does Jane live? What does Jane do for a living?

An Example of Attention

- Process the entire input as a whole
- Focus on relevant words

Input: *"Jane, who lives in London and works as a software engineer, met her best friend Sarah for dinner".*

Prompt: Where does Jane live? What does Jane do for a living?

Output: Jane lives in London and works as a software engineer.

Impact for Software Engineers

AI-Driven Development Is Here: Should You Worry?

Neil A. Ernst and Gabriele Bavota

By leveraging massive language models and the plethora of openly available source code, AIDEs promise to automate many of the obvious, routine tasks in programming.

IEEE Software, 2022

Why Should We (Not) Worry?

Morgan Willis for AWS
Posted on 3 de mar. • Edited on 4 de mar.

28 7 5 6 7

Is Software Engineering Cooked? Not Yet. But Maybe.

#ai #developers #coding #agents

"Software engineering is solved." This is all I see lately when scrolling LinkedIn, X, or Reddit.

Code Is Easy for AI

Software Is Verifiable

Automation

“I think humans will be in the loop for longer than we believe.”

Source: Dev.to

LabSoft Research on GenAI

Evaluating the Effectiveness of LLMs in Fixing Maintainability Issues in Real-World Projects

Henrique Nunes
Federal University of Minas Gerais
Belo Horizonte, Brazil
henrique.mg.bh@gmail.com

Eduardo Figueiredo
Federal University of Minas Gerais
Belo Horizonte, Brazil
figueiredo@dcc.ufmg.br

Larissa Rocha
State University of Bahia
Alagoinhas, Brazil
larissabastos@uneb.br

Sarah Nadi
New York University Abu Dhabi
Abu Dhabi, United Arab Emirates
sarah.nadi@nyu.edu

Fischer Ferreira
Federal University of Itajubá
Itajubá, Brazil
fischer.ferreira@unifei.edu.br

SANER, 2025

MaRV: A Manually Validated Refactoring Dataset

Henrique Nunes
Federal University of Minas Gerais
Belo Horizonte, Brazil
henrique.mg.bh@gmail.com

Tushar Sharma
Dalhousie University
Halifax, Canada
tushar@dal.ca

Eduardo Figueiredo
Federal University of Minas Gerais
Belo Horizonte, Brazil
figueiredo@dcc.ufmg.br

FORGE, 2025

Improving JavaScript Test Quality with Large Language Models: Lessons from Test Smell Refactoring

Gabriel Amaral
UEFS
gabrielamaralsousa@gmail.com

Henrique Nunes Gomes
UFMG
henrique.mg.bh@gmail.com

Eduardo Figueiredo
UFMG
figueiredo@dcc.ufmg.br

Carla Bezerra
UFC
carlailane@ufc.br

Larissa Rocha
UNEB/PGCC-UEFS
larissabastos@uneb.br

SBES, 2025

Bad Smell Detection using Google Gemini

Larisse Amorim, Ivandeclei Mendes da Costa, Leticia Alves, and Eduardo Figueiredo

{larisseamorim, ivandeclei, leticiasma}@ufmg.br, figueiredo@dcc.ufmg.br
Software Engineering Laboratory (LabSoft), Federal University of Minas Gerais (UFMG), Brazil

COMPSAC, 2025

Bibliography

- Neil Ernst and Gabriele Bavota. **AI-Driven Development Is Here: Should You Worry?** IEEE Software, 2023.

- Ashish Vaswani *et al*. **Attention is all you need**. Advances in Neural Information Processing Systems (NIPS) 30, 2017.