

# Artificial Intelligence: Friend or a Foe?

Lígia Falcão Nogueira Miranda <sup>a</sup>

<sup>a</sup> Faculty of Internet System Technology, Federal Technological University of Paraná (UTFPR - Universidade Tecnológica Federal do Paraná), Guarapuava, Brazil, ligiafalcao@alunos.utfpr.edu.br.

**Abstract.** As artificial intelligence continues to advance, individuals have grown concerned about the potential threat to their jobs. Wonder if the developer job is being completely replaced for AI. They ponder whether this intelligence might supplant human capabilities, rendering them obsolete. With that in mind, this article is to illustrate how this tool can be exploited and whether its applications would ultimately position it as a foe or as a powerful ally and friend, assisting in the creation of more easy and efficient work, focusing on developer jobs.

**Keywords.** Artificial Intelligence (AI), developers, ChatGPT, Work Optimization, AI Benefits, AI Efficiency.

## 1. Introduction

Since Artificial Intelligence (AI) has gained headway in society and its usage has become more popular, many questions and concerns have arisen. Question like, will AI replace all jobs? Will we still need writers, artists, or developers when AI can perform these tasks independently?

Those doubts and fears began to arise in people's minds, as the main information channels began to massively address the same subject the same subject. A BBC technology reporter wrote an article entitled "AI Could Replace the Equivalent of 300 Million Jobs," authored by Christopher Vallance. In The Wall Street Journal, another article entitled "What Will AI Do to Your Job? Take a Look at What It's Already Doing to Coders" was written by Christopher Mims. Even the British Computer Society (BCS) published an article entitled "Will AI Replace Software Engineers?"

Beyond the fear of job loss, the real question we should be discussing is how we can make AI a friend rather than a foe. In this way, this technology will not "take our jobs" but enhance efficiency and productivity. We need to learn how to use AI tools effectively to quickly find solutions that would typically require hours of searching through multiple pages or watching extensive videos to comprehend issues with our code, such as spotting mistakes in camelCase or missing semicolons.

## 2. Methodology

### 2.1 What are the available AI

Artificial Intelligence (AI) represents a fascinating frontier in computer science and technology, seeking

for the development of systems and machines capable of performing tasks typically associated with human intelligence. This is often possible through the implementation of Machine Learning (ML), defined by Wired (2018) as "the process of refining computer predictions or task performance through the use of example data or real-world experience."

The technology demonstrate proficiency across a broad spectrum of cognitive abilities, encompassing problem-solving, learning, logical reasoning, understanding natural language, and recognizing patterns.

AI technology finds diverse applications across various domains. It powers virtual personal assistants like Siri and Alexa, enables self-driving cars, enhances recommendation systems, aids in medical diagnosis, and impacts numerous other areas. The rapid advancement of AI holds the potential to bring about revolutionary changes in numerous industries and aspects of our daily lives.

Peter Norvig and Stuart Russell classified into three primary levels of intelligence: Narrow AI, Artificial General Intelligence (AGI), and Artificial Superintelligence (ASI).

- **Common Usage**

In common usage, Narrow AI, often referred as Weak AI, is the most prevalent form. It is purpose-built for specific tasks, operating within a limited scope. While it outstands at well-defined tasks, it lacks the breadth of general intelligence.

Exemplification of Narrow AI applications include virtual personal assistants such as Siri or Alexa,

recommendation systems like those used by Netflix to suggest content based on your viewing history, and game-playing AI, such as chess programs that allow you to compete against a machine.

- **Developer Usage**

In the field of software development, concepts like AGI and ASI are garnering attention. Nevertheless, it's imperative to recognize that Narrow AI currently holds sway and is extensively applied across various domains. The pursuit of AGI and ASI remains an enduring and formidable challenge in the realm of AI research. And this kinda of usage will be the main focus of this article.

One practical application of Narrow AI is found in Integrated Development Environments (IDEs) like Visual Studio Code, IntelliJ IDEA and PyCharm. These IDEs employ concepts to enhance the coding experience. As developers write code, the AI analyzes the context and provides valuable suggestions for completing code snippets, suggesting function names, or recommending variable names. This functionality not only enhances productivity but also aids in error prevention and maintains code quality following coding conventions.

Another prominent example of Narrow AI made for developers is GitHub Copilot. Developed collaboratively by GitHub and OpenAI, GitHub Copilot is a tool that streamlines the coding process. It seamlessly integrates with code editors like Visual Studio Code, assisting by offer suggestions, auto-generating code snippets, and providing context-aware guidance as they write code. GitHub Copilot powered machine learning models and extensive datasets of code from open-source projects to make coding more efficient and effective, simplifying the resolution of programming challenges.

Like Copilot, ChatGPT also relies on OpenAI's GPT-3.5 architecture, classified as Narrow AI or Weak AI. GPT-3.5 is tailored for specific natural language comprehension and generation tasks, lacking the attributes associated with AGI (Artificial General Intelligence) or ASI (Artificial Superintelligence).

## 2.2 Reasons to use AI for developer research

The using of the technology for developer research can easily simplify the process of finding information, automate tasks, and gaining insights into various aspects of software development. Here are some steps on how to use AI for developer research:

- **Keyword Analysis and Trend Identification**

Utilizing AI-driven tools, such as keyword research applications, facilitates the identification of current trends and keywords within the developer community. This aids in making informed decisions regarding the specific areas of development to prioritize (Jones, 2012).

- **Content Generation**

Can be a valuable aid in producing developer-centric content, encompassing code snippets, documentation, and blog posts. Notably, tools like GPT-3 enhance the efficiency of generating developer resources (Marcus Du Sautoy, 2020).

- **Automated Documentation**

AI-driven documentation generation tools for the automatic creation and maintenance of developer documentation. This approach not only conserves time but also guarantees the documentation's ongoing accuracy and currency (Gentle & Fleming, 2017).

- **Code Generation and Refactoring**

Artificial intelligence plays a role in code snippet generation, code refactoring, and the identification of optimization opportunities. Specifically, tools like CodeGPT serve this purpose (Fowler, 2019).

- **Bug Prediction and Detection:**

Utilize machine learning models for the anticipation and detection of software bugs. These models possess the capability to scrutinize alterations in the codebase, historical bug records, and various contributing factors, thereby aiding in the early identification of potential issues during the development phase (Diamantopoulos & Symeonidis, 2020).

- **Code Search and Analysis:**

AI-based code search engines can help developers find relevant code snippets and libraries quickly. These tools often use natural language processing to understand and retrieve code based on descriptions and queries.

- **Collaboration and Code Review:**

Utilize collaboration and code review tools enhanced by AI capabilities, which can automate segments of the code review process, spotlight potential concerns, and propose enhancements.

- **Developer Community Insights**

AI has the capability to examine developer forums, social media platforms, and other online communities, offering guidance into the challenges faced by developers, emerging technologies, and the popularity of various tools.

- **Predictive Analytics for Resource Allocation**

Use to predict resource requirements for development projects. Machine learning models can analyze historical data to estimate the time, budget, and personnel needed for a project (SIEGEL,2016).

- **Chatbots and Virtual Assistants**

Integrate AI-powered chatbots or virtual assistants onto developer websites or platforms for real-time responses to common queries, offering guidance and immediate assistance (Shevat, 2017).

- **Natural Language Interfaces**

Create natural language interfaces enabling developers to interact with software and tools using everyday language. This enhancement can enhance accessibility and user-friendliness (Lane, 2022).

- **Security Analysis**

AI finds practical application in the realm of security research, encompassing activities such as vulnerability scanning, threat detection, and the analysis of security code to guarantee the software's security (Seitz, 2015).

- **Data Analytics and Visualization:**

AI-powered data analytics and visualization tools to extract insights from extensive datasets pertaining to developer activities, project advancement, and user behavior (Provost & Fawcett, 2013).

- **Continuous Integration and Deployment (CI/CD)**

Incorporate AI within CI/CD pipelines for the automation of testing, deployment, and monitoring procedures, enhancing both software quality and the speed of release (Humble & Farley, 2011).

- **Machine Learning Models for Development Tasks**

Train machine learning models for precise development undertakings, encompassing responsibilities like code completion, code categorization, or forecasting software metrics (Géron, 2019).

Bear in mind that although AI can elevate developer research and productivity, comprehending the constraints of AI tools and substantiating their outcomes in real-world scenarios is paramount. Furthermore, staying abreast of the latest AI advancements in the realm of software development can enable you to maximize the potential of these technologies for research endeavors.

### 2.3 How to use AI for developer research

For developer research can significantly enhance the efficiency and effectiveness of various aspects of software development. AI offers a wide range of capabilities that can be harnessed to make it easier the research, generate content, optimize coding processes, and improve overall project outcomes.

To embark on this journey, it's essential to begin with a clear and well-defined roadmap. Start by define your research goals and objectives. Take the time to consider what particular areas or challenges within developer research you aim to tackle through the

power of AI.

By Identify these objectives upfront, you'll not only delimit a focused direction for your research but also ensure that the AI tools and methodologies you employ align precisely with the issues you intend to address. This initial step lays the foundation for a successful integration of AI into your developer research endeavors, enabling you to maximize its benefits and outcomes. Choose the most relevant AI tools and technologies that align with your research objectives to include natural language processing (NLP) models, ML algorithms or specialized developer-focused AI platforms.

Make possible in identifying current trends and keywords in the developer community that can help guide your research focus and provide insights into what developers are currently interested in. Also, AI-powered content generation tools can create developer-related content, such as code snippets, documentation, and blog posts, more efficiently. These tools, like GPT-3, can save time and help you giving a high-quality resource and can help you keep developer documentation up-to-date, which is crucial for project success and saving time on manual updates.

GitHub Copilot is an AI-powered code assistant integrated into the Visual Studio Code (VS Code) editor. It provides real-time code suggestions, auto-completions, and even generates code based on your comments or descriptions. It understands context, helps with syntax errors, and supports various programming languages. Developers can work more efficiently, explore code options, and enhance collaboration with this tool, though it's important to remember that it complements coding skills rather than replacing them.

Similar to Copilot, ChatGPT has been integrated into VS Code, allowing you to directly inquire about code issues. The chatbot reads your code as you write it, helping identify errors and even assisting with bug testing. This why the research gets more efficient without need to search in multiple different places.

### 2.4 How AI save coding time

Works as a time-saving powerhouse in the coding realm. It automates diverse tasks, from code generation to error detection, refactoring, and documentation. Testing processes are streamlined, code search is expedited, and predictive capabilities simplify coding patterns. This enhances efficiency, code quality, and liberates developers from repetitive tasks, allowing them to concentrate on high-level coding endeavors.

In essence, AI functions as a coding assistant, enabling more efficient and error-reduced coding. By handling routine and time-consuming coding tasks, AI empowers developers to channel their efforts into creative problem-solving and innovative development, ultimately resulting in significant time and effort savings in the coding process.

## 2.5 How to keep you code clean with AI

Help you maintain clean and well-organized code simply. It offers automated code formatting, assists in code reviews, generates documentation, suggests refactoring, and detects errors. This assures your code remains tidy and error-free, saving you time and effort in the process.

## 2.6 How be creative with AI

Promote creativity with AI involves harnessing its capabilities to inspire and elevate your creative endeavors. This encompasses generating novel ideas, creating content, collaborating with AI-driven insights, exploring new artistic frontiers, and refining your work with AI-provided feedback. By embracing AI as a creative partner, you can broaden your creative horizons and produce distinct and engaging creations across various domains.

Various AI tools are proficient at image generation. For example, DeepDream enhances images in dream-like ways using neural networks. GANs like BigGAN and StyleGAN create hyper-realistic visuals. DALL-E takes it a step further, generating images based on text descriptions. Additionally, AI Painter transforms hand-drawn sketches into digital paintings through AI algorithms. These tools span a wide spectrum of image creation possibilities, highlighting AI's versatility in the visual arts.

AI-generated images are indispensable supports for developers, whether they are working on website design as frontend developers or crafting elements for games. They help developers to bring their creative visions to life, ensuring that the graphical aspects of their projects align with their imagination and design goals.

Offer creative inspiration by tailoring ideas to your project's specific needs that provide you with multiple choices, making it easier to select the one that aligns best with your vision. Can help generate harmonious color palettes that streamline your decision-making process, ensuring they complement your system's design seamlessly.

## 2.7 How solve easy problems with AI

Solving simple problems with AI is swift and efficient. AI can swiftly analyze data, identify patterns, and offer solutions, saving time and minimizing errors. Its capabilities span from automating routine tasks to providing data-driven insights, making it a valuable tool for addressing straightforward challenges.

Upgrading systems or migrating code between languages and frameworks can be simplified with AI assistance.

## 3. Results and Discussions

In an effort to put into practice and illustrate the concepts discussed regarding the utilization of AI in code development, with the support of ChatGPT, I will begin by showcasing how to simplify the process

of swapping components between different frameworks and programming languages.

As you can observe in **Figure 1**, we have created an input example using the Angular framework. However, our intention is to migrate it to the React framework. We've shared the code with ChatGPT and asked for guidance on how to achieve this transition.

```
import { Component, Input } from '@angular/core';

@Component({
  selector: 'app-input',
  template: '<input [value]="value" (input)="onInput($event)" />',
})
export class InputComponent {
  @Input() value: string;

  onInput(event: any) {
    this.value = event.target.value;
  }
}
```

**Fig. 1** – Angular Input Component.

In **Figure 2**, they provide us with the outcome: how to implement the same input in React as we did in Angular. Furthermore, they offer an explanation of how this works.

```
import React, { useState } from 'react';

function InputComponent(props) {
  const [value, setValue] = useState(props.value);

  const handleInputChange = (event) => {
    setValue(event.target.value);
  };

  return (
    <input value={value} onChange={handleInputChange} />
  );
}

export default InputComponent;
```

**Fig. 2** – Transformed into React Input Component by ChatGPT.

The chat reply to us that “This Angular input component takes an @Input property called value and binds it to the input field's value. It also listens for input changes and updates the value property accordingly. In this React version, we use the **useState** hook to manage the component's state. We initialize the value state with *the props.value value*. We also define a **handleInputChange** function to update the value state when the input field value changes. Finally, we render the input element, binding its value to the value state and attaching the **onChange** event handler to keep the state updated.” (OpenAI, 2023).

Without the right knowledge, make this transformation can consume a significant amount of time as you search for solutions. However, by simply asking, you can receive an answer in a matter of seconds. Besides, if the initial solution doesn't work, you can easily request alternatives by providing the

error encountered. This approach allows you to comprehend common tasks, such as creating an input, more quickly and allocate your efforts toward achieving the specific form or result you desire.

Ruiyun Xu, Yue Feng, Hailiang Chen (2023) did an experiment with 10 people and evaluate each participant's answer to each task using a scoring system based on a total of 10 points. Each task in our experiment has objective answers. After three tasks, they got these results in **Figure 3**.

Measure Group Mean	ChatGPT (48 participants)	Google Search (47 participants)	Difference (ChatGPT - Google)	F-statistic
<b>Panel A. Search Efficiency</b>				
<b>Self-reported Task Time (min)</b>				
Total time for three tasks	11.35	18.75	-7.40	26.88***
Time spent on task 1	1.83	3.37	-1.54	18.09***
Time spent on task 2	2.40	3.61	-1.20	7.22***
Time spent on task 3	7.12	11.78	-4.66	22.86***
<b>Time Spent on Search Tool (min)</b>				
Total time for three tasks	5.79	14.95	-9.15	34.81***
Time spent on task 1	0.34	2.42	-2.08	22.11***
Time spent on task 2	0.52	2.78	-2.26	40.39***
Time spent on task 3	4.93	9.81	-4.88	14.06***
<b>Panel B. Search Efforts</b>				
Total # of queries on three tasks	7.36	8.13	0.77	1.30
# of queries (task 1)	1.55	2.13	-0.58	7.13***
# of queries (task 2)	1.30	1.65	-0.35	3.39*
# of queries (task 3)	4.51	4.35	0.16	0.09
Average query length for three tasks	37.54	12.05	25.49	27.59***
Query length (task 1)	13.50	9.90	3.60	12.84***
Query length (task 2)	18.43	6.11	12.32	156.63***
Query length (task 3)	80.72	19.82	60.90	18.74***
<b>Panel C. Search Performance (Full Score: 10)</b>				
Average performance score on three tasks	8.55	8.77	-0.22	0.83
Performance score on task 1	10.00	8.19	1.81	19.46***
Performance score on task 2	9.81	9.74	0.07	0.14
<i>If answers pointing to the destinations</i>	5.00	8.88	-3.88	201.68***
Performance score on task 3	5.83	8.37	-2.54	24.23***
<b>Panel D. User Experience</b>				
Perceived information quality	5.90	4.62	1.27	15.85***
Technology trust	5.38	5.30	0.07	0.46
Perceived ease of use	6.00	5.57	0.43	3.80*
Perceived usefulness	6.19	5.30	0.89	10.10***
Perceived enjoyment	5.87	4.74	1.12	14.19***
Satisfaction	6.06	5.27	0.79	9.32***

**Fig. 3** – Comparisons of Search Performance, Behavior, and User Experience.

**Note:** Analysis of variance (ANOVA) is employed to test the difference between the two groups. Significant level: \*\*\*  $p < 0.01$ ; \*\*  $p < 0.05$ ; \*  $p < 0.1$ .

In conclusion, their study found that ChatGPT offers significant improvements in work efficiency, allowing users to complete tasks more quickly and potentially creating a better user experience. However, it's important to recognize that ChatGPT doesn't consistently outperform traditional search engines. While ChatGPT excels in providing answers to straightforward questions and offering general solutions, this convenience may unintentionally discourage users from conducting further exploration and identifying potential misinformation in its responses. The survey data they gathered also indicates that people generally perceive ChatGPT-generated information as higher in quality and more accessible compared to Google Search, and they express a similar level of trust in both sources (Xu, Feng & Chen, 2023).

## 4. Conclusion

By the end of the day, it's evident that AI serves as a valuable tool, higher efficiency and speed of tasks for individuals. However, it's important to emphasize that AI can't entirely replace human roles or even solve all the problems. The technology, while powerful, to reached an advanced level that

guarantee complete trust in it to replace human jobs.

Indeed, AI it's a helpful friend and powerful tool for simple or routine tasks. Allows individuals to allocate more time and attention to challenging tasks. However, similar to research on Google, one must possess a level of knowledge to interpret and utilize the information that is given. Knowing how to ask the right questions and use the tool is crucial. Otherwise, the tool may not yield meaningful results.

Besides, creating and maintaining AI systems can be compare to managing any other complex system, where without adequate knowledge and skills, the system may accumulate errors and issues that you'll not know how to fix them. In the end, while AI is a important tool, it's more efficient in the hands off individuals who understand the capabilities and limitations.

## 5. References

- [1] Vallance, C. *AI could replace equivalent of 300 million jobs*. [Internet] 2022. [updated 2022 Mar; cited 2023 Set 10] Available from: <https://www.bbc.com/news/technology-65102150>
- [2] Mims, C. *What Will AI Do to Your Job? Take a Look at What It's Already Doing to Coders*. [Internet] 2023. [updated 2023 Jun; cited 2023 Set 10] Available from: <https://www.wsj.com/articles/ai-jobs-replace-tech-workers-8f3dc92>
- [3] BSC. *Will AI replace software engineers?* [Internet] 2022. [updated 2022 Aug; cited 2023 Set 10] Available from: <https://www.bcs.org/articles-opinion-and-research/will-ai-replace-software-engineers>
- [4] Wired, *The WIRED Guide to Artificial Intelligence*. [Internet] 2018. [updated 2022 Aug; cited 2023 Set 10] Available from: <https://www.wired.com/story/guide-artificial-intelligence/>
- [5] Russell, S., & Norvig, P. *Inteligência Artificial: Uma Abordagem Moderna*, Pearson, 2013.
- [6] Jones, R. *Keyword Intelligence: keyword research for search, social, and beyond*. Indianapolis: J. Wiley & Sons, 2012.
- [7] Marcus Du Sautoy. *CREATIVITY CODE: Art and Innovation in the Age of Ai*. S.L., Belknap Harvard, 2020.
- [8] Gentle, A., FLEMING, D. *HOLCOMB, K. Docs like code: write, review, test, merge, build, deploy, repeat*. Austin, Tx: Just Write Click, 2017.
- [9] Fowler, M. *Refactoring: improving the design of existing code*. Boston: Addison-Wesley, 2019.

- [10] Diamantopoulos, T., Symeonidis, A. L. *Mining software engineering data for software reuse*. Cham: Springer, 2020.
- [11] Siegel, E. *Predictive analytics: the power to predict who will click, buy, lie, or die*. Hoboken, New Jersey: Wiley, 2016.
- [12] Shevat, A. *Designing Bots*. [s.l.] "O'Reilly Media, Inc.", 2017.
- [13] Lane, H. *Natural Language Processing in Action*. S.L.: O'reilly Media, 2022.
- [14] Seitz, Justin. *Black Hat Python: Python Programming for Hackers and Pentesters*. San Francisco, No Starch Press, 2015.
- [15] Provost, F., Fawcett, T. *Data Science for Business: What You Need to Know About Data Mining an Data-Anal*. Beijing Etc., O'reilly, 2013.
- [16] Humble, J., Farley, D. *Continuous Delivery*. Addison-Wesley, 2011.
- [17] Géron, A. *Hands-on machine learning with Scikit-Learn and TensorFlow concepts, tools, and techniques to build intelligent systems*. 2. ed. [s.l.] O'Reilly Media, Inc., 2019.
- [18] OpenAI. ChatGPT. [Internet] 2023. [Accessed September 10, 2023] Available from: <https://www.openai.com/chatgpt/>
- [19] Xu, R., Feng, Y., Chen, H. "*ChatGPT vs. Google: A Comparative Study of Search Performance and User Experience*", 2023. Cornell University. arXiv preprint arXiv:2307.01135.