

# Generative AI from natural language to code, with precision

Q2 2023

## Can we get from natural language to code?

### Why do we need it?

Can you get from natural language to code? Just a few years ago, people would have filed this question under the “too difficult” category. However, recent advances in technology have brought new techniques to the table, and the solutions are getting better and better.

In this article, we explore some of the latest technology and ask - is it possible for a person with a limited programming background, to write text in natural language and obtain code that is usable out-of-the-box?

The first main question is - why do we want this technology to get from natural language to code? The answer is that this brings about many benefits that can be exploited.

Firstly, being able to go from language to code lowers the barriers of entry for people to be able to generate code for their projects. Instead of needing to learn the ins and outs of a programming language, one only must learn the required syntax in natural language, which makes programming more accessible to those otherwise unfamiliar with technology.

Next, this increases the reusability of code between projects. Once code has been written and associated with the right phrases, the user only must enter a sufficient and roughly the same language to reuse the code, instead of re-typing it out or copying it from a different source. This ensures that future projects can piggyback off existing ones more efficiently and lessen the time needed to create future projects.

Lastly, with the shortage of IT staff available globally, this tool is useful in allowing simpler use cases to be generated from natural language, freeing up talent to focus on the more challenging use cases which require more human intervention.

## The answer engine

### Wolfram Alpha

The maths-and-science-inclined university student would remember Wolfram Alpha, a tool that has probably saved them at various parts of their studying career. Officially launched in 2009, it is an early example of a tool that can parse natural language to deliver something technical to the user, albeit not a fully workable programming code. Wolfram Alpha can plot graphs, solve equations and do arithmetic - for instance, we can ask it to solve the following quintic equation:



Find the solution to the equation  $x^5 + 3x + 1 = 0$

NATURAL LANGUAGE MATH INPUT EXTENDED KEYBOARD EXAMPLES UPLOAD RANDOM

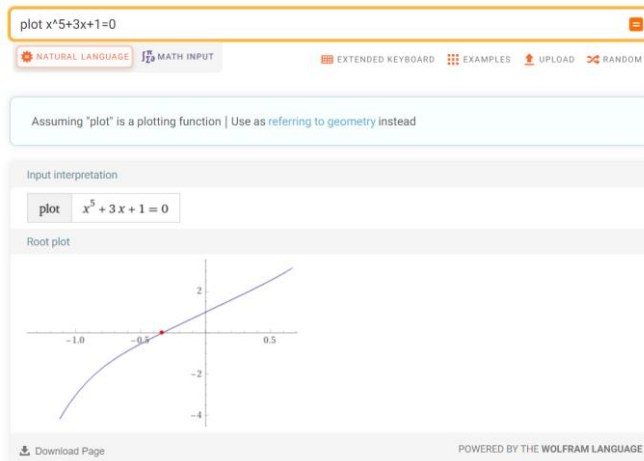
Input interpretation

solve  $x^5 + 3x + 1 = 0$

Results Exact forms More digits

$x \approx -0.839072 - 0.943852i$   
 $x \approx -0.839072 + 0.943852i$   
 $x \approx -0.331989$   
 $x \approx 1.00507 - 0.937259i$   
 $x \approx 1.00507 + 0.937259i$

Being able to parse natural language, it can identify what the user needs despite the “mathematical” aspects of the statement being the same. We can, for instance, ask it to plot the above on a graph instead:



plot  $x^5 + 3x + 1 = 0$

NATURAL LANGUAGE MATH INPUT EXTENDED KEYBOARD EXAMPLES UPLOAD RANDOM

Assuming "plot" is a plotting function | Use as referring to geometry instead

Input interpretation

plot  $x^5 + 3x + 1 = 0$

Root plot

Download Page POWERED BY THE WOLFRAM LANGUAGE

## The current rave - the linguistic engine

### ChatGPT

If you have been following the news in the last few months, you would have learnt of ChatGPT by OpenAI - an artificial intelligence chatbot that can respond to a variety of user topics. Naturally, people have tried using ChatGPT to see if it can help the end user write code. Let's try something simple - can we get ChatGPT to generate a simple Python code to add two numbers together?

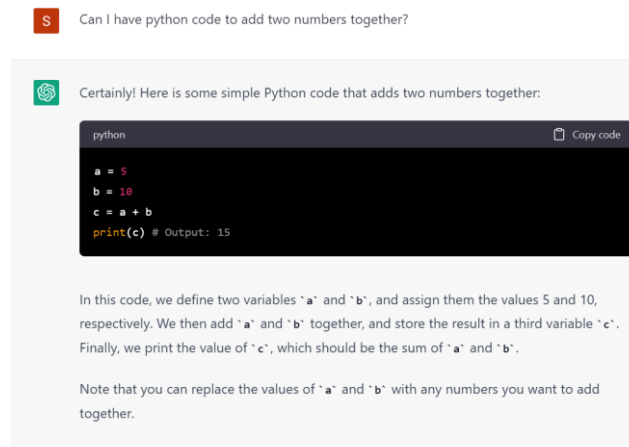


Figure 1 Python code to add two numbers together, along with a simple explanation.

And it can! However, ChatGPT fails spectacularly if you give it anything just a bit more complicated than this, so the role of the programmer is safe, at least for now.

Nonetheless, this represents a step forward, and it shows that the building blocks are there, albeit not precise yet right now.

Trying to  
connect  
the dots  
between  
language  
to code

### GitHub Copilot

Another tool that has come out in recent years is GitHub Copilot, which is both able to autocomplete code that the user types in, as well as understand text prompts inputted by the user, and is able to suggest code that the user can then accept or reject.

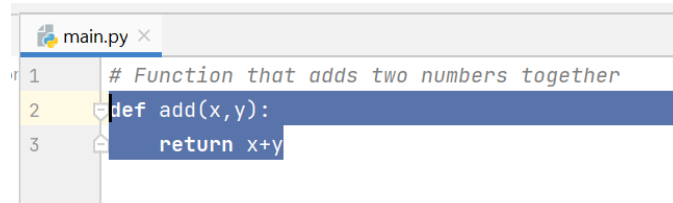


Figure 2 - Sample of auto-generated code using GitHub Copilot

For the above example, we can similarly write a prompt to GitHub Copilot - that we want a function that can add two numbers together - and Copilot is able to decipher the name and content of the function.

Alternatively, the user can begin typing the code they desire, and Copilot will attempt to automatically fill in the rest of the code - the user is then able to either accept or reject the suggestion. Copilot requires a bit more technical background - after all, you are writing the first bits of code - but in return is also able to provide a more precise solution.

## Language to code with precision

Can generative AI be precise? The answer is yes - natural language can be turned into code, albeit the development right now is still in its early stages. We have seen an early example of Wolfram Alpha, which can perform simple calculations from natural language requirements. Following that, more recent examples where tools such as ChatGPT and GitHub Copilot have been introduced that are beginning to be able to generate simple code from written requirements. This is only bound to get better as models develop further, with tools being able to understand and parse more complicated requirements and generate better code for the user.

Besides these tools, other companies like Google and Microsoft have developed Bard and Bing Chat respectively, their answers to ChatGPT. Over in China, other similar products are in the works from companies such as Baidu, Alibaba and Tencent. This increased competition in the market is sure to accelerate growth and bring tools like these to the forefront - with companies inevitably filling the hole in the market that is currently there.

Over here at AgileAlgo, we believe that generative AI can help organisations be more efficient and effective in their systems work. Generative AI can support machine learning of written requirements to code. Deeptech can significantly improve the software engineering process.

We have created a Virtual System Implementer (VSI) Platform that can generate AI use cases in a matter of minutes. We have started this process and continue to machine learn the language to code use cases. Our platform is built using various AI techniques including Natural Language Processing for the written requirements and Graph Neural

Networks to infer which requirement is associated to which codes. Our AI create other AI use cases.

We have digitised the agile approach for epics and user story writing to create the precision required for code generation. We will learn as we progress.

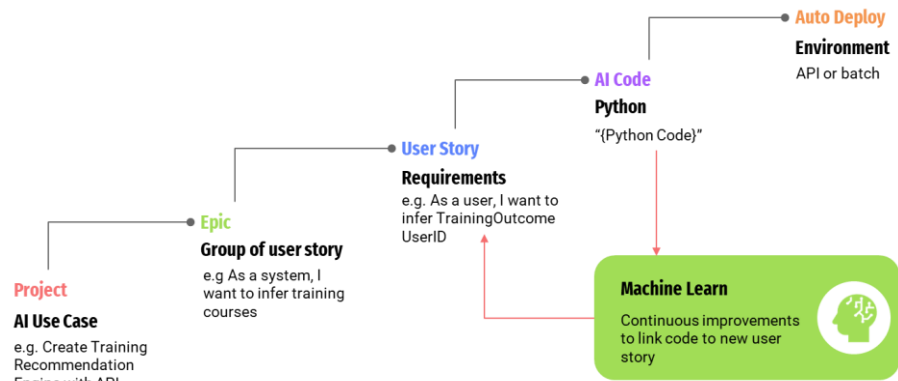


Figure 3- Our digitised agile approach

Currently, we support 213 different user story types that can be combined to create various AI cases. We support Graph Inference, Big Data Analytics Algorithms, Computer Vision, Natural Language Processing algorithms, and combinations of data engineering use cases.

## What's next for you?

AgileAlgo is established in helping companies deal with the lack of Data Science and IT resources as we focus on applying deep tech in the space of Artificial Intelligence and Software Engineering. We solve issues via an automated written requirements-to-code approach. This means that we use written requirements, which all projects should have, and interpret them automatically to get to the required codes. We seek to deliver faster and at a fraction of the cost to get to your use cases.

We are an enabler and accelerator to your business growth. We use AI to create AI. We can create an Inference Engine in 5 minutes (we are happy to demonstrate that!). We have developed a platform that makes it simple for companies to use. Through training and self-service options, users are able to help themselves, hence keeping the cost of such development low.

Sign Up for a FREE account

Talk to us!

Visit our website