

We keep hearing how AI and Large Language Models will take all of our jobs. Back in 1980, when I first knew I wanted to be a computer programmer, I was told that the computers would program themselves soon, and I would be out of a job. It hasn't happened yet, but the latest technology is certainly changing. It is certainly an interesting time to be working in software. So what exactly does writing Delphi code with a Large Language Model assistant look like?

## What is a Large Language Model?

I'm only going to provide the basics, since this is changing so fast. I'll also provide some historical background.

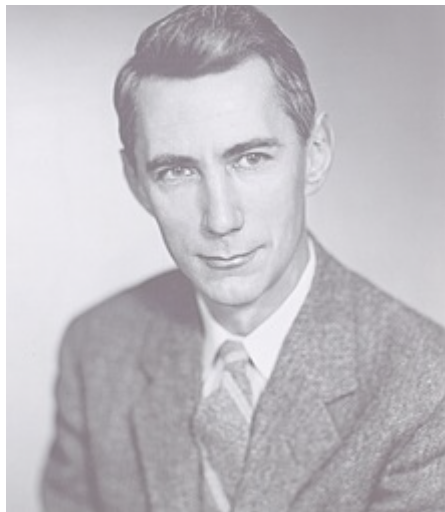
If you are completely new to LLM and GPT, know I may use those terms interchangeably, but technically a [Generative Pre-trained Transformer \(GPT\)](#) is a type of [Large Language Model \(LLM\)](#). These fall under the category of [Generative Artificial Intelligence](#) built out of [Artificial Neural Networks](#) that are trained on [Natural Language Processing](#) tasks. [Follow those Wikipedia links for more recently updated information.]

The idea of an artificial neural network has been around a long time, but it was only recently, thanks to the massive parallel nature of modern GPUs that they have become practical. Essentially they are a mathematical model based on the neural network in a biological brain that is trained on massive amounts of input data. This results in the internal structure of the neural network being a "black box" where the behavior is a result of the training, and not directly specified through normal code. While they are trained on massive amounts of data, they do not store the data they were trained on. Instead they retain the patterns of the data, allowing them to generate new text that matches the same pattern.

[3Blue1Brown](#) just published a fantastic video on [GPTs specifically](#). If you aren't familiar with 3Blue1Brown, it is an amazing math education YouTube channel by Grant Sanderson. He is one of the better educators I've encountered on these topics, and has over 170 videos, spanning multiple playlists. The GPT video is part of a [course on Neural Networks](#).

But what is a GPT? Visual intro to Transformers | Deep learning, chapter 5





Someone I hear very few people talking about when it comes to Large Language Models is [Claude Shannon](#). I'm not sure where I first learned about Mr. Shannon, but I assumed he was one of those people like [Charles Babbage](#), [Ada Lovelace](#), [Blaise Pascal](#), [Margaret Hamilton](#), [George Boole](#), [Hedy Lamarr](#), [Alan Turing](#), & [Grace Hopper](#) that everyone knew about (I included links for a reason, go read about them). But surprisingly he seems to be less know. In 1950 he built Theseus, the first example of artificial intelligence, a device that learned by trial and error. He was also the father of [Information Theory](#), laid the ground work for [Boolean algebra](#) to be used in the digital computing, and so much more. He is described as the 20th century engineer who contributed the most to 21st century technologies. His achievements are said to be on par with those of [Albert Einstein](#) and [Sir Isaac Newton](#) in their fields.

## Who is Claude Shannon?

There is a fantastic documentary on Claude Shannon called [Bit Player](#) that I highly recommend. (I found a version of it with [Brazilian Portuguese subtitles](#) too.) Here is an [older documentary](#), but I prefer [Bit Player](#).

Claude Shannon - The Bit Player Movie Trailer



Beyond all of that, the thing I find the most interesting and relevant to Large Language Models is his research into “word pairs.” The idea being that if you look at which word typically follows another word, you can simulate “intelligent” writing. Since he came up with this before the invention of digital computers, he would test his theory with a novel by looking for the next occurrence of a word, copying the word after it, and repeating the process until you end up with a crude sentence.

will see that it just contains a list of words and numbers, but doesn't contain the original analyzed text.

## Using ChatGPT 4 to Write Delphi Code

As much as possible I used ChatGPT 4 to write the code. There were a few small edits I made on my own, and I built the sample app myself, but ChatGPT did most of the library code and all the unit tests. I've shared the [history of my conversation with ChatGPT](#) so you can see the process. It did really well this time.

- [ChatGPT 4 conversation](#)
  - [Archive copy of conversation](#)
- [Full Source Code on GitHub](#)

## What Did we Learn?

I didn't time it, but I believe this was faster than if I wrote all the code myself. Where a Large Language Model really shines is when you are working on something you are less familiar with, saving you from needing to consult the documentation all the time. This could be an algorithm you are unfamiliar with, an API you are new to, etc.

If you try using ChatGPT or a similar large language model to write code, here are a few things to watch out for:

- **Hallucinations:** I used to have a friend who would confidently answer any question you asked him, even if he had no idea what he was talking about. That is exactly the same as an LLM. Unless you ask it to do something it was specifically trained not to do, it will give you an answer, and act very confident about it.
- You need to **know how to code:** Since it might give you garbage, you need apply your knowledge to see if the output makes sense and is accomplishing what you asked of it.
- **Iterate:** Have the LLM do small amounts of work, and give it feedback on as you go along, making changes.
- **Zero-Shot-Learning:** If it is completely lost, give it some sample code. That is sometimes enough to get it going in the right direction.
- **Older Training Data:** Most of the LLMs are trained on old data, that means it isn't always up to date on the latest changes in Delphi. Luckily ChatGPT 4, Bing CoPilot, and others are able to go online and update themselves. This produces mixed results in my experience, but if for example you ask it to take advantage of the inline variable declarations, it usually will figure it out.
- **Limited Scope/Token Window:** Basically if you have a really long chat, it will sometimes "forget" things that you discussed or asked at the beginning. This is because each pas of the chat only includes a limited amount of the previous conversation. The length of this look back is getting longer and longer, but still worth keeping in mind.
- LLMs are **not Search Engines:** If you look at the analysis output of the Delphi program you will see it doesn't contain the input text that it analyzed. This is the same as an LLM. This means it didn't "copy" the original work, and also means it most likely won't reproduce it exactly. It is entirely within the realm of possibilities that it will reproduce it, but it is also possible for a million monkeys at a million typewriters to write Shakespeare.
- **Limited Delphi Training data:** While I don't know for sure what these public LLMs are trained on, I'm pretty sure it was trained on the [DocWiki](#) and much of the Delphi code on GitHub and elsewhere on the web. It probably wasn't trained on the RTL itself, while it probably was trained on the [Free Pascal RTL](#) (since it is open source and freely available online.) The result is I encounter a lot of hallucinations where it confidently claims a method or class exists, when it doesn't. Again, this is because it isn't a search engine. It doesn't look up the method, but instead it extrapolates what it expects the method to be called and how it works. Sometimes it uses a FPC method, other times it uses a .NET Method. Usually you can provide feedback and it will correct itself, but I've had a couple of times it was insistent that I was wrong.
- **Prompt Engineering:** This is a whole new art form in how to ask LLMs questions that produce the answers you are looking for. You can read about what others are doing, or you can just experiment. If you aren't getting a good

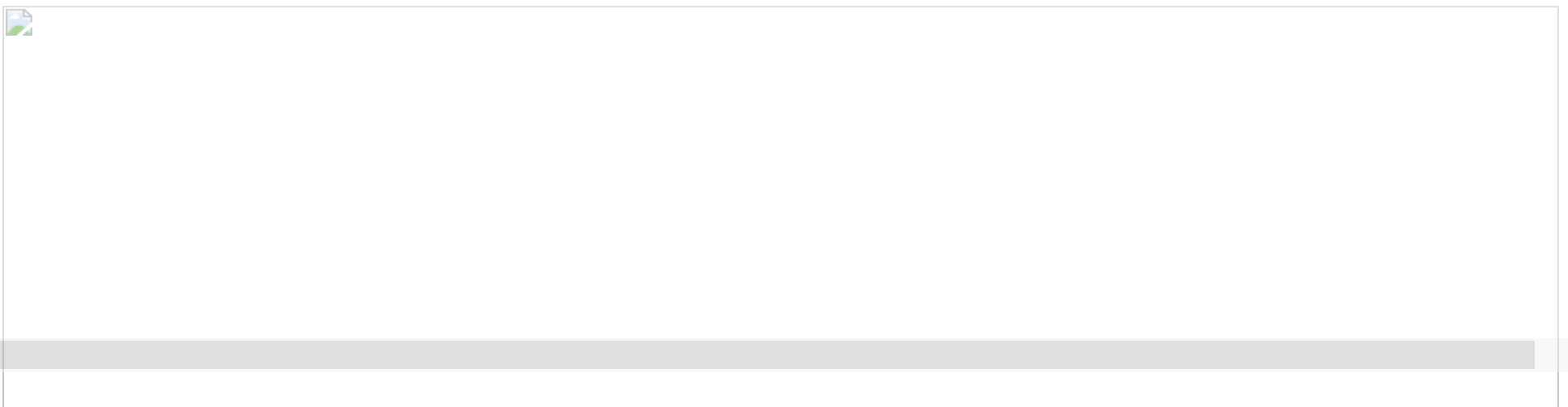


- **Synchronizing Changes:** One thing that tripped me up a few times is when I made changes to the code after I got it from ChatGPT without telling ChatGPT about the change. Then when I later asked it for some more code it wouldn't know about my changes so its output wouldn't be in sync with my code. For example I renamed a method, so every time it generated new code that called that method I would need to rename it again. The solution would be to input my changed code back into the conversation.

## What's Next?

There are free options to use [Bing CoPilot](#), [ChatGPT 3.5](#), [Google Gemini](#), and others. If you haven't already tried working with them, you really need to set aside some time to familiarize yourself. If you tried a while ago, try it again. Things are changing very fast right now, and while the AI may not directly take your job, if your co-workers learn to use these LLMs to be more productive, they may make you redundant.

Is all of this too much for you? Well you are in luck since GDK Software is completely focused on the rapidly evolving world of Delphi software development. Everything from updating your old Delphi 7 project, to implementing the latest machine learning features and functionality, we have you covered.



## CONTACT

Let us help you to realise your ambitions

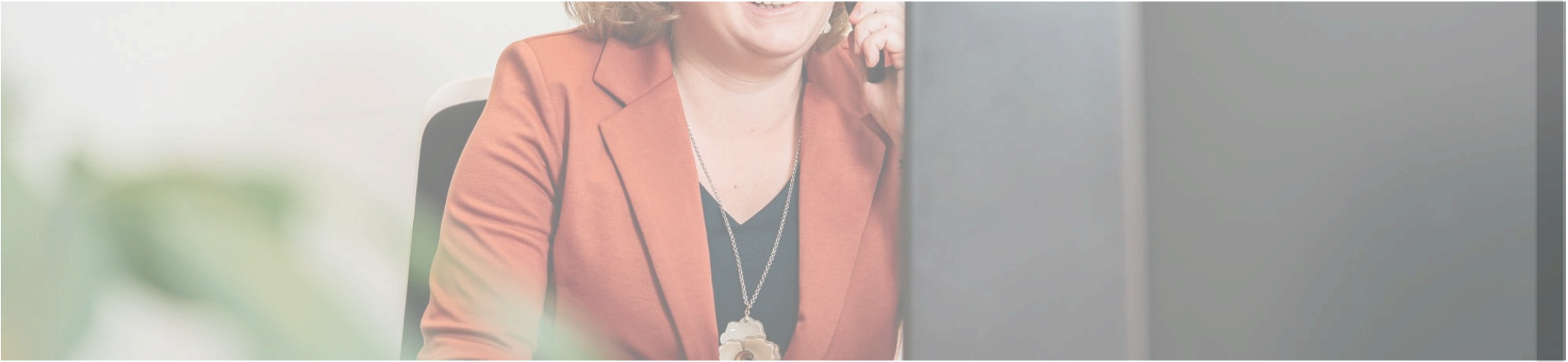
GDK SOFTWARE UK

**(+44) 20 3355 4470**

GDK SOFTWARE USA

**+1 (575) 733-5744**

**info@gdksoftware.com**



Contact details

GDK Software NL

(+31) 78 750 1661

Brederodelaan 5b

3353 GG Papendrecht

The Netherlands

GDK Software UK

(+44) 20 3355 4470

Press House, Suite 32, Crest View Drive, Orpington

BR5 1FE

United Kingdom

GDK Software BR

(+55) 19 99377 1300 Whatsapp

1450, 3 avenue, Jardim claret, Rioclaro /SP - Rooms 7 and 8

CEP - 13503-251

Brazil

GDK Software USA

+1 (575) 733-5744

6299 N Eagle Rd #1023 Boise

ID 83713

United States

Shortcuts

Borland Delphi

Delphi 5

Delphi 6

Delphi 7

Delphi programming

Delphi update

Delphi language

Delphi compiler

Delphi development

Delphi developer



Delphi application

Delphi coding