

---

# What is a Large Language Model?

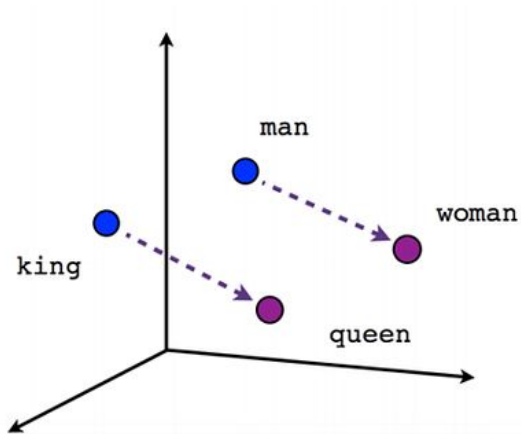
---

A Recent breakthrough was the addition of the “Transformer” (2017)<sup>1</sup>

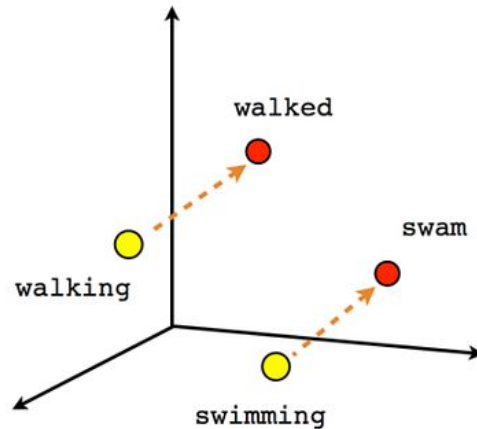
- A type of neural network architecture
- Word embeddings are used to represent text using high dimensional vectors that capture semantic and syntactic properties
- Positional encoding - Keeps track of position of words in text
- Attention - Allows the model to consider every word in the input (prompt) when generating the output
- Self-attention - Allows a word to be understood in the context of the words around it. The model can assign weighted values to each of the words.

1: Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems 30 (2017).

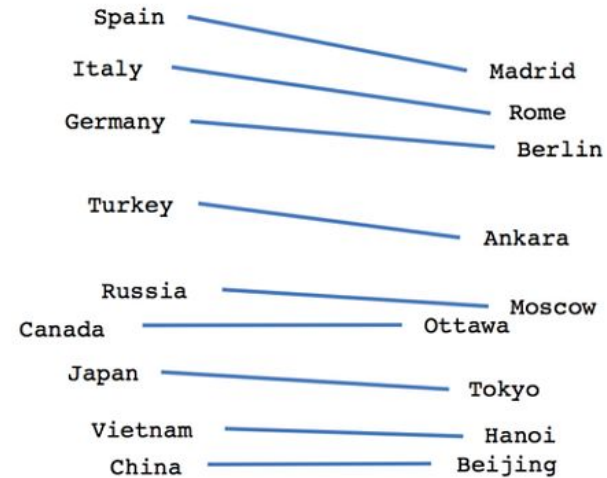
# Embeddings



Male-Female



Verb tense



Country-Capital

---

# Risks

---

- Models are biased!!!
- They hallucinate (although this can be controlled somewhat with prompt engineering)
- Reward hacking - If you train a model to achieve the highest satisfaction rating in a customer survey it might focus only on that outcome and ignore finding the best solution! (many worry this risk is an existential threat!)<sup>1</sup>



---

# Resume / CV example use case

---

This is just a bit of fun to show the difference between blatantly bad use and good constructive use of the tool, and highlight the utility of prompt engineering and few-shot prompting!



Please create a resume for Jack Johnson who is a song writer and musical performer. He sings and plays guitar.



**Jack Johnson**



Address: 123 Beach Avenue, Surf City, USA

Phone: (555) 123-4567

Email: [jack.johnson@email.com](mailto:jack.johnson@email.com)

**Objective:**

Experienced songwriter and musical performer seeking opportunities to collaborate and create memorable experiences through music. Adept at guitar and vocals, with a strong background in various musical genres, including folk, rock, and acoustic.

**Experience:**

**Solo Musical Performer** – *Self-Employed*

Surf City, USA | June 2000 – Present

- Performed at over 200 live shows across the country, including festivals, intimate venues, and large arenas.
- Released 6 studio albums with 4 reaching platinum status.
- Collaborated with other artists for live performances and studio recordings.
- Managed promotions, booking, and logistics for self-hosted tours.

**Lead Singer & Guitarist** – *The Waves Band*

Surf City, USA | January 1998 – May 2000

- Performed in 100+ shows throughout the region.
- Co-wrote 12 songs, 3 of which became top hits on regional charts.
- Worked collaboratively in a team to produce cohesive performances and recordings.

**Songwriter** – *Beach Tunes Music Publishing*

Surf City, USA | June 1995 – December 1997

- Penned over 40 songs for various artists, 10 of which made it to the top 40 charts.
- Collaborated with multiple artists to refine lyrics and melodies to fit their unique styles.
- Conducted research on market trends to produce relevant and popular content.

**Education:**

Bachelor of Music – [Surf City University](#)

Regenerate



You are an expert resume writer. Please craft a resume that highlights the applicants creativity, dependability and deep knowledge and experience within the floral industry. Her experience and education with a masters in organizational communications make her the perfect candidate for the role of store manager. I will follow this system message with details about the applicant



Sure, please provide me with the necessary details about the applicant's work history, achievements, skills, and any other pertinent information you'd like to include in the resume. This could include the applicant's name, contact information, prior positions, companies they've worked for, their specific responsibilities and accomplishments in those roles, any relevant certifications or skills, and references, if applicable.



Please create a single page resume for Yolanta Karwowski-Gormanly, she lives at 1 Hollyberry Dr. Hopewell Junction, NY 12533. She is applying for a store manager role at a flower shop. Sections should include objective, work experience and education. She has worked at Twilight Florist in Hopewell Junction, NY since December of 1995 as an Assistant Floral Designer. In this role, she helped customers, cleaned and prepped flowers, prepared deliveries, worked the cash register and took orders in person and over the phone. Between May 1997 to August 1997 she worked at Flowers by Rene in Fishkill, NY in the same role as an assistant floral designer. She then returned to Twilight Florist in August 1997 in the same role and has been there ever since. She was promoted to floral designer in 2005 and has created arrangements for weddings, funerals, dances and all kinds of special events. She also manages the store in the absence of the owner. She has managed scheduling of store employees and drivers, routing of deliveries, ordering supplies, managing orders from wire services, dealing with disputes and keeping customers satisfied. Yolanta has a Masters Degree of the Arts in Organizational Communications from the University at Albany which she earned in 2004, and a Bachelors of Arts in Psychology that she earned in December 2001 also from the University of Albany.



**OBJECTIVE**

Detail-oriented and dependable floral professional with a deep understanding of the industry and over 27 years of experience. Seeking to leverage my creativity, strong communication skills, and comprehensive knowledge of store operations to secure a store manager position.

**WORK EXPERIENCE**

**Twilight Florist, Hopewell Junction, NY**

*Floral Designer / Acting Store Manager (2005 - Present)*

- Designed and crafted floral arrangements for a variety of occasions including weddings, funerals, and special events, demonstrating creativity and attention to customer preferences.
- Managed store operations in the owner's absence, including employee scheduling, delivery routing, supply ordering, and dispute resolution.
- Maintained strong customer relationships through effective communication and problem-solving, resulting in high customer satisfaction.

*Assistant Floral Designer (1997 - 2005, 1995 - 1997)*

- Provided exceptional customer service, handling both in-person and phone orders with accuracy and efficiency.
- Performed regular duties such as flower cleaning and preparation, handling cash register, and preparing deliveries.
- Demonstrated a strong commitment to the team and store success, leading to a promotion to Floral Designer.

**Flowers by Rene, Fishkill, NY**

*Assistant Floral Designer (May 1997 - August 1997)*

- Assisted in designing floral arrangements, gaining experience in different styles and techniques.
- Maintained store cleanliness and readiness, ensuring a welcoming environment for customers.
- Demonstrated adaptability and eagerness to learn, contributing to the overall success of the store.

**EDUCATION**

**Master of Arts in Organizational Communications (2004)**

**Bachelor of Arts in Psychology (2001)**

*University at Albany, Albany, NY*

**SKILLS**

- Expert in floral design and arrangement
- Strong customer service and communication skills
- Detail-oriented and organized
- Proficient in store operations and management
- Dependable team player and leader



---

# AI and Software Development

---

Let's specifically talk about this class and the rules of use going forward.

- OpenAI GPT-3 / 4
- GitHub Copilot / Copilot x / Tabnine
- Claude
- Your custom fine tuned LLM build using LangChain
- Any others that fit into this category

**Can we use a tool like this to create our software???**

**Let's find out.**



# The Program to build ->

This is a really good high level explanation of what needs to be accomplished to make this happen.

Notice it is not an actual program.

Once you stop asking for a solution to a specific thing, like “solve a Fibonacci sequence in javascript” the “cheat” factor ends.

Note the last paragraphs:



Create a Node.js web application called myFileShare with a graphical UI that allows users to upload files and share them with each other. It needs the ability to allow users to create and edit a public profile, make files they upload public or private, show public files in a feed, allow users to like files that they see in the feed and to comment on the files.



Creating a Node.js web application with a graphical UI that allows users to upload and share files with each other would involve several steps. Here is an overview of the process:

1. Use a framework like Express.js to handle routing and requests.
2. Create a front-end interface using a JavaScript library like React or Angular to handle user interactions and display the file feed.
3. Implement a database (like MongoDB) to store user information, files, and other relevant data.
4. Use a package like Multer to handle file uploads and storage on the server.
5. Implement authentication and authorization to allow users to create and edit their public profiles, make files they upload public or private, and control access to the application's features.
6. Allow users to like and comment on files by adding functionality to the front-end and storing the data in the database.

Please note that the above steps are just an overview and there are many more details and considerations to take into account such as security, scalability, performance and error handling.

It's also important to take note that this is a complex task and it would require knowledge of web development and experience with Node.js, JavaScript and databases.

can you create a server.js file to handle the server-side logic using Express?

Sure, here is an example of a basic server.js file that uses Express to handle the server-side logic for a Node.js application called myFileShare:

```
const express = require('express');
const multer = require('multer');
const app = express();

const storage = multer.diskStorage({
  destination: (req, file, cb) => {
    cb(null, 'uploads/');
  },
  filename: (req, file, cb) => {
    cb(null, file.originalname);
  }
});
const upload = multer({ storage });

app.use(express.json());
app.use(express.urlencoded({ extended: true }));

app.post('/upload', upload.single('file'), (req, res) => {
  // handle file upload here
});

app.post('/createProfile', (req, res) => {
  // handle profile creation here
});

app.post('/shareFile', (req, res) => {
  // handle file sharing here
});

app.get('/feed', (req, res) => {
  // handle displaying a feed of public files here
});

app.post('/likeFile', (req, res) => {
  // handle file likes here
});

app.post('/commentOnFile', (req, res) => {
  // handle commenting on files here
});

app.listen(3000, () => {
  console.log('Server listening on port 3000');
});
```

This is just an example, and you would need to add the logic to handle the requests made to each endpoint. It's also important to note that you'll need to create the relevant database tables or use a database service to store the user's profile, files, likes, and comments data.

## Let's ask for a component of the program:

Again, an interesting start, it saves a bit of cookie cutter code, but the honest truth is I am going to modify it quite a bit. There really isn't any substance here, and for any of this to be useful you must understand it!!!

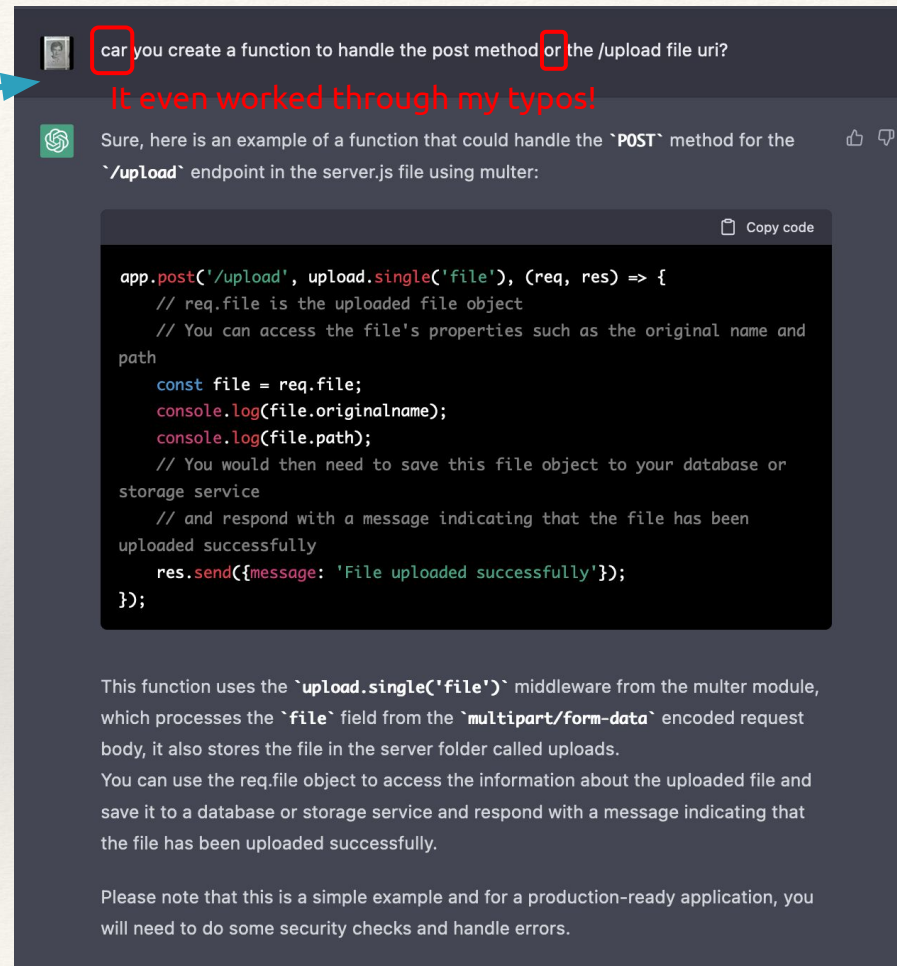
## Questions.

1. What is express / app?
2. What is post()?
3. What is this syntax? (req, res) => {}
  - a. Is it in a function parameter?? How?
4. What parameter(s) are being passed into multer.diskStorage()?.
  - a. What type are they?
  - b. Why is there a colon in the middle?
5. How do I call any of these app. post() functions??

# So what happens when we get specific with our questions ?

1. Look at what the notes are below the code! Even making a specific request still has many missing components
2. What is req? And why can I use it to access information about the uploaded file?
3. Where do I get started saving the data to a database??
4. What security checks will I need to do? What errors should I handle?? HOW!?!?

See the danger here?? YOU need to know this and YOU will never learn it if you lean on this technology too early.



The screenshot shows a GitHub discussion on a dark theme. At the top, a user profile picture is on the left, and a comment says "can you create a function to handle the post method or the /upload file uri?". Below this, a red text overlay reads "It even worked through my types!". The main content is a response from the OpenAI GPT-4 logo, which provides a code example for a POST endpoint using Multer. The code is in a dark-themed editor with syntax highlighting. Below the code, there is explanatory text about the Multer middleware and the req.file object, followed by a note about security and production readiness.

can you create a function to handle the post method or the /upload file uri?

It even worked through my types!

Sure, here is an example of a function that could handle the `POST` method for the `/upload` endpoint in the `server.js` file using `multer`:

```
app.post('/upload', upload.single('file'), (req, res) => {
  // req.file is the uploaded file object
  // You can access the file's properties such as the original name and
  path
  const file = req.file;
  console.log(file.originalname);
  console.log(file.path);
  // You would then need to save this file object to your database or
  storage service
  // and respond with a message indicating that the file has been
  uploaded successfully
  res.send({message: 'File uploaded successfully'});
});
```

This function uses the `upload.single('file')` middleware from the `multer` module, which processes the `file` field from the `multipart/form-data` encoded request body, it also stores the file in the server folder called uploads.

You can use the `req.file` object to access the information about the uploaded file and save it to a database or storage service and respond with a message indicating that the file has been uploaded successfully.

Please note that this is a simple example and for a production-ready application, you will need to do some security checks and handle errors.



# Final note on Debugging

In addition to helping with boiler plate code (once you understand the code it is generating, it is an extremely useful debugger.

This is IN ADDITION to, and not a replacement for other good practices!!

- Good Style
- Learning to use your debugger
- Good comments
- Resolving all error at compile and run time
- And more!



Why is the following javascript code only generating a '5'? let myArray = [1,2,3,4,5];  

```
for(let i=0; i=myArray.length; i++) {  
  console.log(i);  
}
```



The code is only generating a '5' because the for loop's condition is using the assignment operator (=) instead of the comparison operator (== or ===). This means that the loop's condition will always evaluate to true, and the loop will only run once, with the value of `i` being set to the length of `myArray`, which is 5.

It should be

```
for(let i=0; i < myArray.length; i++) {  
  console.log(i);  
}
```

or

```
for(let i=0; i <= myArray.length-1; i++) {  
  console.log(i);  
}
```

to print the indexes.

---

# Learn through practice

---

You do not learn the basics by osmosis. You learn them by DOING them. Seeing code generated by an AI is not the same as you 'learning' it.

(this is especially dangerous, because the things it can do well are the things you learn early on as we have seen) You cannot make up for building a solid foundation later.

The name Copilot was chosen with care, you are still the pilot and 100% necessary! This is a productivity tool in CS!

---

# AI assist class rules

---

1. Any use of code that is not 100% written by you, if allowed for by the assignment you are working on **must be cited**.
2. **If you use any AI assist**, include how you used it (use good commenting practices to explain in your code!) I want to hear about your experiences the good and the bad



---

# AI Assistance

---

What are your thoughts?

How Large an impact do you think these tools will have?

What industries / types of work will be most affected?